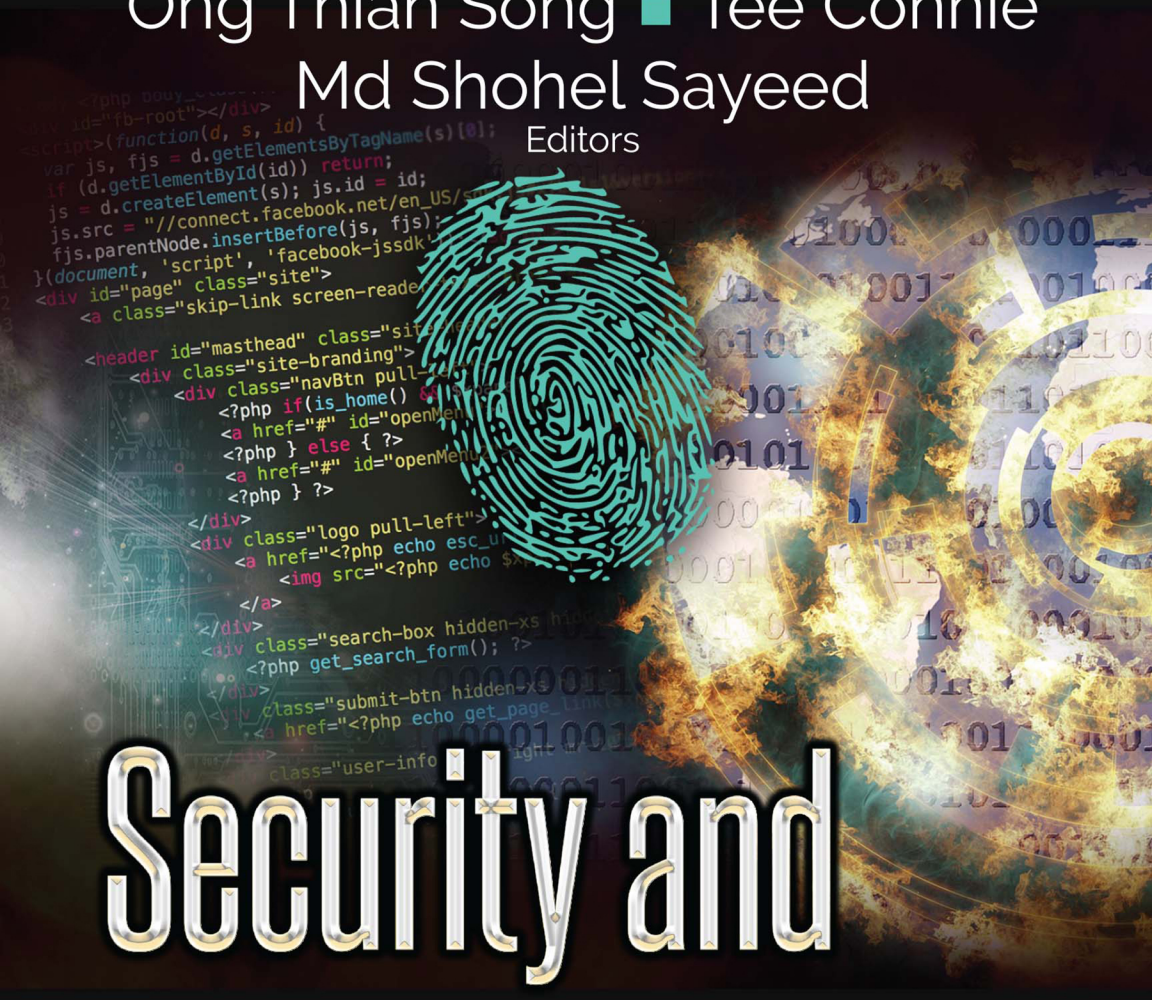
Ong Thian Song ■ Tee Connie
Md Shohel Sayeed
Editors



Security and Authentication

Perspectives, Management and Challenges

Cybercrime and Cybersecurity Research

NOVA

CYBERCRIME AND CYBERSECURITY RESEARCH

SECURITY AND AUTHENTICATION

PERSPECTIVES, MANAGEMENT AND CHALLENGES

CYBERCRIME AND CYBERSECURITY RESEARCH

Additional books in this series can be found on Nova's website under the Series tab.

Additional e-books in this series can be found on Nova's website under the e-Books tab.

CYBERCRIME AND CYBERSECURITY RESEARCH

SECURITY AND AUTHENTICATION

PERSPECTIVES, MANAGEMENT AND CHALLENGES

ONG THIAN SONG
TEE CONNIE
AND
MD SHOHEL SAYEED
EDITORS

nova
science publishers
New York

NOTICE TO THE READER

Library of Congress Cataloging-in-Publication Data

ISBN: 978-1-53612-943-4 (eBook)

Published by Nova Science Publishers, Inc. † New York

CONTENTS

Preface		**vii**
Chapter 1	State-of-the-Art Authentication Techniques:Threats and Vulnerabilities *Chin Poo Lee and Kian Ming Lim*	**1**
Chapter 2	Privacy Protection in Machine Learning: The State-of-the-Art for a Private Decision Tree *Yee Jian Chew, Kok-Seng Wong and Shih Yin Ooi*	**13**
Chapter 3	Key Distribution and Management in Cryptography *Bachir Bendrissou and Yean Li Ho*	**41**
Chapter 4	Targeted Image Forensics *Rimba Whidiana Ciptasari*	**65**
Chapter 5	Deep Learning for Abnormal Behavior Detection *Nian Chi Tay, Pin Shen Teh and Siok Wah Tay*	**87**
Chapter 6	Security Issues in Wireless Sensor Networks and IoT *Jayakumar Vaithiyashankar*	**117**

Chapter 7 Finger Vein Biometrics: The Future for a Mobile Authentication System **131**
Ahmad Syarif Munalih and William Ardianto

Chapter 8 Android Device Misplacement Remedy via Bluetooth-Enabled Technology **149**
Siew-Chin Chong and Kaven Raj S/O Manoharan

Chapter 9 A Labeled Network-Based Anomaly Intrusion Detection System (IDS) Dataset **181**
Nicholas Ming Ze Lee, Shih Yin Ooi, Yong Kian Lee and Ying Han Pang

Chapter 10 Seamless Biometrics for a Smart Office **217**
Soh Ting Yong and Michael Goh Kah Ong

Chapter 11 Hiding Information Within a QR code Based on a Color Subcell (SQRC) **241**
Ari Moesriami Barmawi and Yudha Viki Alvionata

Index **263**

PREFACE

Issues around security and authentication have received greater attention as the world becomes more digitized and interconnected. There are a myriad of technological advances like smart mobile devices, wearable devices, Internet of Things (IoTs), cloud computing and social networks that benefit people all over the world, transforming how they work and communicate with each other. However, these new technologies also bring new security and privacy challenges. For example, there are massive attacks by malicious malware like WannaCry that cost great financial loss to individuals and institutions. Besides, there are ample amounts of software and programs that quietly collect, share and sometimes disclose huge amounts of personal information.

This book presents the current popular issues in information security covering human users, hardware and software, the Internet and also communication protocols. The book provides a comprehensive combination of studies that offer integrated solutions to security and authentication problems. The topics covered in the book include mobile authentication systems, security in wireless sensor networks and IoTs, network-based intrusion detection systems, privacy protection in machine learning, deep learning for surveillance, and also targeted image forensics. An understanding of these areas ensures the ability to adapt to and address

new challenges in the technological dependent world as these fields evolve.

The primary target audiences of this book are students and researchers from security technology and information technology management. The editors have been blessed by the assistance of many people concerning all aspects for the preparation of this book. The editors would like to express their sincere gratitude to the anonymous reviewers for their professional support and dedication to reviewing the chapters of this book. They are deeply grateful for the excellent contributions of the authors. Last but not least, special thanks also go out to Nova Science Publishers for presenting the opportunity to prepare and publish this book.

January 2018
Ong Thian Song
Tee Connie
Md Shohel Sayeed

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 1

STATE-OF-THE-ART AUTHENTICATION TECHNIQUES: THREATS AND VULNERABILITIES

Chin Poo Lee[1,*] and Kian Ming Lim[1]
[1]Faculty of Information Science and Technology,
Multimedia University, Melaka, Malaysia

ABSTRACT

Today, most of us are struggling with many accounts, ranging from emails, social media networks to bank accounts. Therefore, we are inevitably obsessed with a copious number of passwords. Simple passwords are easy to memorize but susceptible to password attack. Complicated passwords, on the other hand, offer better protection, yet they are difficult to memorize. In view of this, several alternatives for security techniques are introduced. Some well-known security techniques are face recognition, fingerprint recognition, iris authentication, and recently pattern lock for mobile devices. Recent studies however reveal that these security techniques could be easily spoofed or fooled. Researchers from Carnegie Mellon University have shown that specially

[*] Corresponding Author, E-mail: cplee@mmu.edu.my.

designed spectacle frames can fool even state-of-the-art facial recognition software. Fingerprints printed using an inkjet printer loaded with capacitive ink and special paper can fool smartphone fingerprint sensors. Researchers from Lancaster University, Northwest University in China, and the University of Bath were able to crack more than 95 per cent of patterns within five attempts.

Keywords: Fingerprint, face, iris, passwords, pattern lock

INTRODUCTION

This chapter reviews some of the attacks that occur on the most commonly used authentication techniques, namely fingerprint, face, iris, passwords and pattern lock on mobile phones.

Fingerprints

Fingerprint recognition technology was previously used in commercialized systems like attendance system, immigration system, and bank system to verify identities. In recent years, the application of fingerprint recognition is extended to personalized devices such as smartphones, tablets and laptop computers. Despite the fact that fingerprint recognition could be easily embedded in many platforms, the technology however suffers from some vulnerabilities.

Some researchers from the Center for Identification Technology Research (CITeR) had created a 3-dimensional fingerprint mold simply based on a fingerprint image. Similarly, a security researcher, Jan Krissler, reconstructed a fingerprint copy of the Germany's Defense Minister using a publicly available software called VeriFinger with high resolution pictures of the minister's hand obtained from different angles. This work suggested that anyone with the set of necessary skills would be able to easily recreate fake fingerprints. There is also a recent concern aroused by the Japanese researchers about fingerprint replication from a peace-sign in

photographs. The peace sign or victory sign is an iconic gesture in the East-Asians community. The research team suggested that fingerprints can be forged from photographs snapped up to three meters, which is pretty normal for selfie photos.

Other researchers, Kai Cao and Anil Jain from Michigan State University showed that fingerprint sensors can be fooled with a standard inkjet printer (Cao et al., 2016). Professor Anil Jain suggested that differences in the valleys and ridges in the fingerprints conduct different electrical currents, hence producing unique images on the fingerprint sensor. In their experiments, they printed fingerprints using a standard inkjet printer. To realize the conductive property of the electrical currents, the printer was loaded with capacitive ink and the fingerprint was printed on an insulating paper. Two of the four smartphones tested were successfully unlocked using the printed fingerprint. It is worth noting that the same technique had successfully assisted the police in Michigan to unlock the phone of a victim of a murder case. The police wanted to access to the data contained in the phone of the victim for possible clues. However, the victim's phone was locked by the fingerprint scanner lock.

Figure 1. Fingerprints printed using a standard inkjet printer loaded with capacitive ink and insulating paper can trick smartphone fingerprint sensors. (Photograph: Michigan State University)

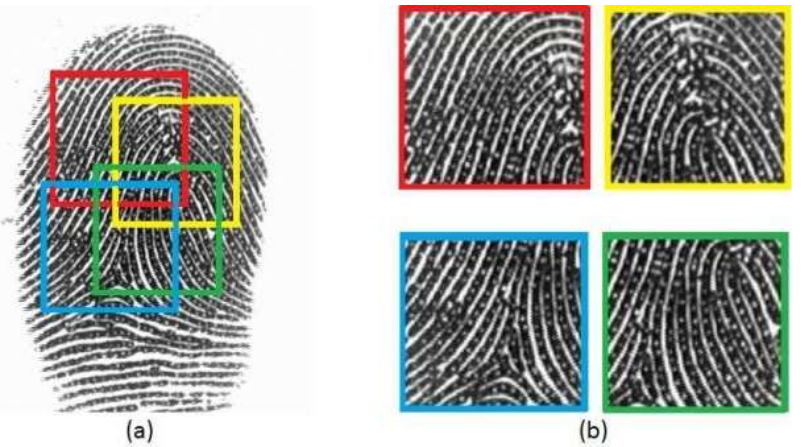


Figure 2. (a) The fingerprint, and (b) the extracted partial fingerprints. (Source: Aditi Roy, Nasir Memon, and Arun Ross)

Therefore, Professor Anil Jain and his team reproduced the capacitive characteristics of the fingerprint and successfully unlatched the fingerprint scanner lock. Recently, some researchers from New York University and Michigan State University developed a digital "master fingerprint" that is capable to bypass many fingerprint scanners (Roy et al., 2017). Their findings were built upon some characteristics of the fingerprint scanner. Firstly, the fingerprint scanner is usually small in size and captures only partial fingerprints. Secondly, the users can enroll fingerprint of multiple fingers. Thirdly, a smartphone always allows the user to make a few attempts of fingerprint unlock. The researchers had extracted thousands of partial fingerprints from a database of 800 fingerprints. From the partial fingerprints, they discovered some partial prints that are able to match the others at a high probability. In their studies, the "master fingerprint" is able to emulate up to 65% of the fingerprints depending on the size of database.

Face Authentication

The usage of face authentication software has grown rapidly in consumer products like laptops and smartphones.

Social media network like Facebook uses face recognition to tag individuals in photos. Google even embedded image processing chip into its smartphones for facial authentication purpose. Alibaba similarly plans to introduce a pay-with-your-face technology by 2017.

Faces are one of the easiest biometrics to steal. Almost everyone has dozens of photos available on the web. In a security conference, some computer vision specialists from the University of North Carolina demonstrated 3-dimensional facial models built from online photos (Xu et al., 2016). Firstly, they identified the landmarks from each photo of the person. These landmarks were used in reconstruction of the facial models. Subsequently, the facial models were enriched with the cues that the security systems normally check for, such as texture, gaze correction and facial expression. The facial models were finally shown using a virtual reality software on smartphones. The virtual reality based facial models also render motion and depth cues that are vital to deceive the facial authentication system that the shown facial model is a real-live person.

Other than that, researchers from Carnegie Mellon University had shown some interesting tricks to fool state-of-the-art facial recognition software (Sharif et al., 2016). They used some spectacles that were printed with specially designed patterns. The patterns were learnt from machine learning techniques exploited for facial recognition.

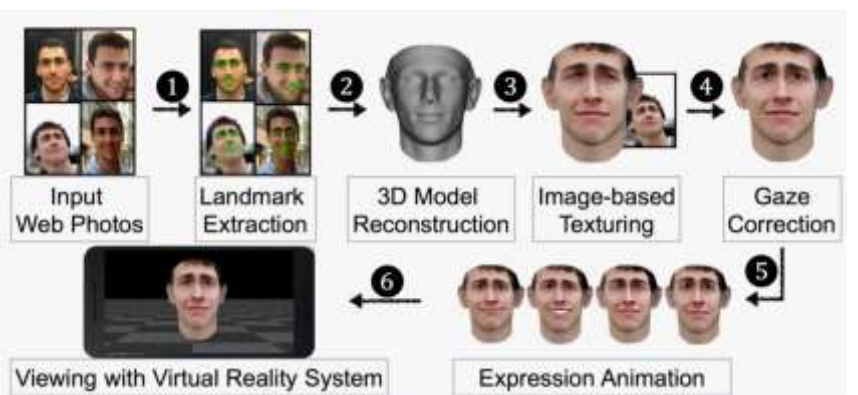
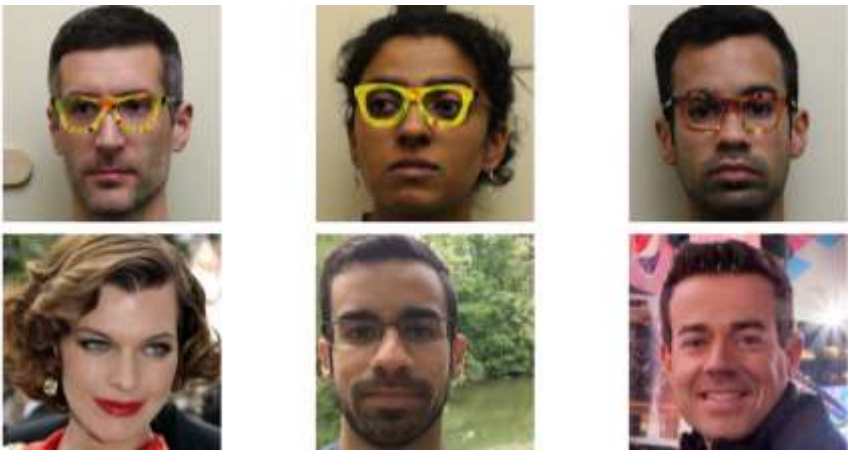


Figure 3. The process of constructing virtual reality based facial models. (Photography: University of North Carolina)



Photography: Carnegie Mellon University.

Figure 4. Top row: researchers wearing specially printed glasses, bottom row: the identity that face recognition software mistaken for. (Photography: Carnegie Mellon University)

The method discovered significant recurring patterns of different complexity levels from the input images. Therefore, by wearing glasses printed with these patterns, it can trick such software into thinking the wearer is someone else.

The researchers elaborated that it is not impossible to make the software into recognizing other objects, people or animals by further tweaking the patterns.

Apart from security and computer vision researchers, facial authentication software also received interests from artists. A Berlin artist, Adam Harvey in his CV Dazzle project, tried to hinder facial authentication system from detecting a face with specially designed makeup and hairstyling (Adam, 2012).

Adam Harvey presented another attempt to confuse facial authentication system in Hyperface project (Harvey, 2017).

In the project, Adam invented anti-surveillance clothing by printing abundant of patterns related to faces on clothing or textiles. The anti-surveillance clothing overwhelms the facial authentication system with thousands of false positives that prostrate it from recognizing the real face of the wearer.

Photograph: Adam Harvey.

Figure 5. Two models in CV Dazzle styling. (Photograph: Adam Harvey)

Figure 6. HyperFace Prototype by “Adam Harvey / ahprojects.com”.

Iris Authentication

In year 2016, Samsung incorporated an iris scanner in Galaxy Note 7 smartphone (Keeping an Eye on Security: The Iris Scanner of the Galaxy Note7, 2016). An infrared LED light and an iris camera placed on the front panel of the phone captures the iris image.

Figure 7. How the iris scanner works. (Graphic: Samsung)

The camera is designed with an image filter that enhances the infrared reflected image of the iris. Once the image is captured, the iris area is extracted and cleaned by eyelid removal. The iris data is then encrypted and stored in the phone.

The same security researcher, Jan Krissler who recreated the fingerprint of Germany's Defense Minister, claimed that the same technique could be used to bypass the iris scanner (Thomas, 2015). By using high resolution pictures gathered from the Internet, a print out of the iris could fool the iris scanner. Jan Krissler however addressed a few factors that determine the success of the iris scanner hack, namely the eyes must be bright, the image should be large, the diameter of the iris must be at least 75 pixels, and the print out of the iris should have a resolution of 1200 dpi.

Password

Passwords are usually adopted as a user authentication method to computers and systems. A password comprises string of alphabets,

numbers, punctuation or special characters. In year 2013, a malware attack on Target's security and payment system in United States had shocked the community (Riley et al., 2014). The malware had stolen 40 million credit card data from transactions in about 1800 stores of the company. Another cyber-attack on Adobe compromised 38 million usernames and encrypted passwords of its active users. Adobe had also lost part of the source codes of Photoshop, Acrobat PDF and ColdFusion software.

There are a few hacking methods used to break password-protected systems, namely brute force attack, dictionary attack and keylogger attack. A brute force attack is a script that adopts trial-and-error method to hack into password-protected systems. The attack systematically attempts to log in with password combinations or passphrases. This method works well with short passwords but the searching time increases exponentially with long passwords since there are too many possible combinations. Brute force attack is also used in encryption and decryption of confidential data.

A dictionary attack is a script that attempts to break password-protected system in a more intellectual manner. The attack tries to log in using predefined words, for instance, headwords in dictionary.

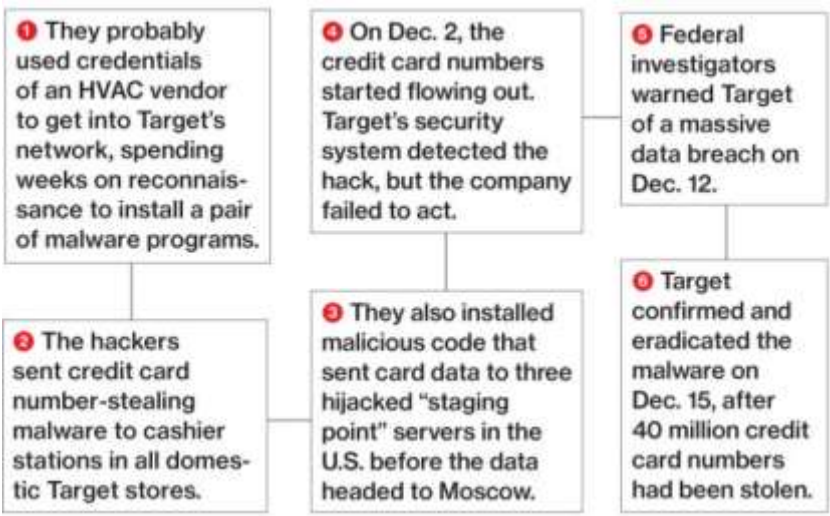


Figure 8. How the hackers broke into Target's security and payment system. (Graphic by Bloomberg Businessweek)

Dictionary attack has a higher success rate because people tend to use common words or their variants generated by appending numbers or special character to the word as passwords. These common words or their variants are easier to remember.

A keylogger attack is a script that records the user's keystroke. The keylogger script normally sneaks into a victim's computer when the user accidentally clicks on a link. It captures everything the user has typed, including usernames and passwords. In this case, even strong passwords do not provide protection against keylogger attack.

Pattern Lock

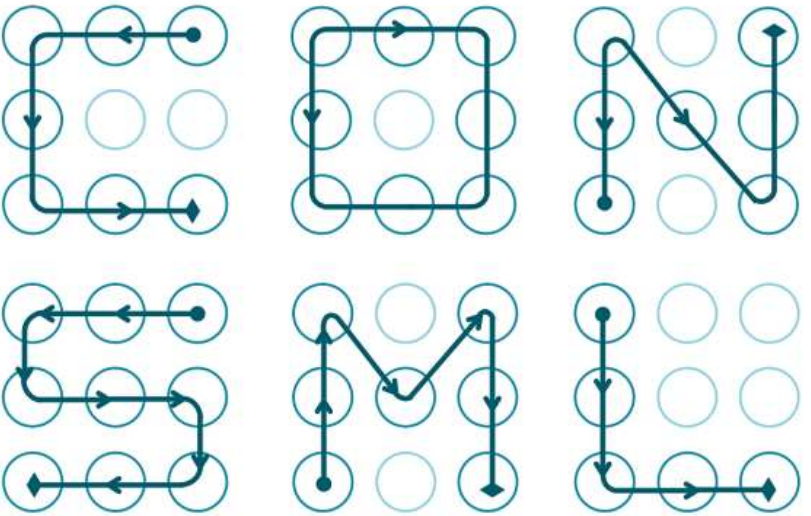
The usage of pattern lock has been proliferating in mobile devices. A pattern lock requires the user to perform a touch gesture to connect the dots shown on the touch screen.

The pattern lock was first introduced in Google's Android devices responding to the question raised by Steve Jobs that, how to prevent accidentally unlocking on touch screen devices? The pattern lock technique was patented by Google under the US patent office.

In an analysis done by Marte Løge, a Master student of the Norwegian University of Science and Technology, revealed that there is only a total of 389112 possible lock patterns (Løge and Dybevik, 2015). In the study, Løge analyzed 4000 Android lock patterns and discovered that 44 percent of the locks begin from the dot at the top left-most corner and 77 percent of the pattern locks start from one of the four corners. Additionally, 10 percent of the patterns resemble an alphabetic letter. Løge also suggested that people tend to avoid patterns that change in direction.

Another research from Lancaster University, Northwest University and the University of Bath adopted computer vision algorithm to guess the lock pattern (Ye et al., 2017).

By covertly observing the fingertip movements when the owner is drawing a lock pattern, the algorithm will suggest a few candidate patterns within seconds.



Photography: Marte Løge.

Figure 9. The simple lock patterns that resemble an alphabetic letter. (Photography: Marte Løge)

The study found that although complex patterns are more secure for a lock pattern guess, these complex patterns are easier to crack using computer vision algorithm. More number of lines used in the lock patterns actually helps the algorithm to narrow down the candidate patterns. In the experiments, the algorithm was able to crack all but one of the complex patterns with the first attempt, 87.5 percent of moderate complex patterns and 60 percent of simple patterns within the first attempt.

REFERENCES

"[In-Depth Look] Keeping an Eye on Security: The Iris Scanner of the Galaxy Note7". *Samsung Global Newsroom*. August 04, 2016. Accessed May 11, 2017. https://news.samsung.com/global/in-depth-look-keeping-an-eye-on-security-the-iris-scanner-of-the-galaxy-note7.

Adam Harvey. *"HyperFace Camouflage"*. Accessed May 11, 2017. https://ahprojects.com/projects/hyperface/.

Cao, Kai, and Anil K. Jain. Hacking mobile phones using 2D Printed Fingerprints. *MSU Technical report*, MSU-CSE-16-2, 2016.

Fox-Brewster, Thomas. "Hacking Putin's Eyes: How To Bypass Biometrics The Cheap And Dirty Way With Google Images". *Forbes.* March 06, 2015. Accessed May 11, 2017. https://www.forbes.com/sites/thomasbrewster/2015/03/05/clone-putins-eyes-using-google-images/#79be702214ae.

Harvey, Adam. "CV Dazzle: Camouflage from computer vision". *Technical report* (2012).

Løge, Marte Dybevik. *"Tell Me Who You Are and I Will Tell You Your Unlock Pattern"*. Master's thesis, NTNU, 2015.

Riley, Michael, Ben Elgin, Dune Lawrence, and Carol Matlack. "Missed alarms and 40 million stolen credit card numbers: How Target blew it". *Bloomberg Businessweek*, 13 (2014).

Roy, Aditi, Nasir Memon, and Arun Ross. "MasterPrint: Exploring the Vulnerability of Partial Fingerprint-based Authentication Systems". *IEEE Transactions on Information Forensics and Security* (2017).

Sharif, Mahmood, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition". In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1528-1540. ACM, 2016.

Xu, Yi, True Price, Jan-Michael Frahm, and Fabian Monrose. "Virtual U: Defeating Face Liveness Detection by Building Virtual Models from Your Public Photos". In *25th USENIX Security Symposium (USENIX Security 16)*, pp. 497-512. USENIX Association, 2016.

Ye, Guixin, Zhanyong Tang, Dingyi Fang, Xiaojiang Chen, Kwang In Kim, Ben Taylor, and Zheng Wang. *"Cracking Android pattern lock in five attempts"*. (2017).

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 2

PRIVACY PROTECTION IN MACHINE LEARNING: THE STATE-OF-THE-ART FOR A PRIVATE DECISION TREE

Yee Jian Chew[1], Kok-Seng Wong[2] and Shih Yin Ooi[1,*]
[1]Faculty of Information Science and Technology,
Multimedia University, Melaka, Malaysia
[2]School of Software, Soongsil University, Seoul, South Korea

ABSTRACT

The explosive growth and widespread accessibility of digital data have led to a surge of research activity in the machine learning field. Typically a massive data collection is required to increase the quality of machine learning result. Often, these data contained highly sensitive information such as medical history, or financial records. Hence, privacy concerns have overshadowed by other factors in today's machine learning systems. A fundamental problem in privacy-preserving machine learning (PPML) is how to make the right tradeoff between privacy and utility. On the one hand, the PPML solution must not allow the original data records

* Corresponding author, Email: syooi@mmu.edu.my.

(e.g., training data) to be adequately recovered (i.e., privacy loss). On the other, it must allow the system to learn the model that is closely approximates to the model that is trained using the original data (i.e., utility gain). In this chapter, we will discuss several emerging technologies that can be used to protect privacy in machine learning systems. In addition, we also provide a state-of-the-art of the adoption of privacy preserving schemes in decision tree algorithms.

Keywords: privacy-preserving, machine learning, classification, ID3, C4.5

INTRODUCTION

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed (Alpaydın, 2010). Almost all machine learning algorithms are focus on the task of data classification or prediction. In order for a machine learner to do such decision, a vast example data or experience needs to be provided to train the machine, especially when human expertise does not exist. Consider a case of converting the acoustic speech signal to a written text in a spoken speech recognition. This task can be accomplished by human almost without any difficulty; however, we may not be able to explain how we actually do it. This is in fact the most challenging part in machine learning: *how to program the learner when we cannot explain how we do it?* Thus, one vital approach in machine learning is through training by referring to a large collection of sample utterances from different people (from different sources) and let the machine learns to map them into words. In general, machine learning is dominated by two subfields, which are data classification (supervised learning) and data clustering (unsupervised learning) (Kohavi & Provost, 1998). In data classification, a set of labeled data set is needed for a classifier to build a reference model through the process of data training, where this model will be adopted to make classification or prediction later. On the other hand, the clustering

algorithms will take the efforts to group the data by observing its similarity to each and other, and this can be done without pre-labeled the data set.

Machine learning has played an increasing important role in many applications such as stock price index detection (Kim & Han, 2000) and hospital readmission risk prediction (Kansagara et al., 2011). Typically a massive data collection is required to increase the quality of machine learning result. Often, these data contained highly sensitive information such as medical history, or financial records. Hence, privacy concerns have overshadowed by other factors in machine learning systems. In particular, if machine learning output includes training data points, it may reveal sensitive information of some (or all) individuals in the training data. Hence, if a data owner does not trust the machine learning system, he or she may provide false data or no data at all. This can cause the machine learning model susceptible to accuracy problem (e.g., producing an overfitted model that will be inaccurate).

Privacy leakage is now a pervasive challenge in applied machine learning systems. In 2006, a famous on-line movie renting service provider (Netflix) starts a $1 million contest for the best technique to improve its movie recommendation system. Netflix publicly released 100 million records, showing the ratings given by 500,000 users to the movies they rent. The released records were anonymized by replacing the usernames with unique identification numbers. According to the study in (Narayanan & Shmatikov, 2008), more than 90% of the subscribers could be uniquely identified from the released records. In 2010, Netflix canceled its second $1 million Netflix Prize due to the raise of privacy concerns (Lohr, 2010).

Issue and Problem Statement

The environment in which a machine learning algorithm operates has driven recent progress in machine learning. In a classical machine learning system, it involved a single program running on a single machine. With the rapid development of network technologies, it is now common to deploy the machine learning system with distributed architecture design. However,

there is growing recognition that such deployment causes privacy concerns to the data owners and model owners. For instance, the goal of knowledge extraction from a large amount of distributed data collides with the privacy of individuals. On one hand, data owners and model owners can extract useful patterns from collected data using these technologies. On the other hand, these technologies can become a threat for individual privacy (i.e., privacy of data) as well as sensitive private patterns (i.e., privacy of model) (Natwichai, Li, & Orlowska, 2005).

Consider a scenario where multiple hospitals want to perform collaborative machine learning to build a predictive model for disease risks on their proprietary inputs. Due to data protection laws and regulations (e.g., EU data protection directive (Boillat & Kjaerum, 2014) and HIPAA (Gould, 2006)), they are not allowed to share individual patient data with others. As stated in EU General Data Protection Regulation (File, 2012): "*Personal data which are, by their nature, particularly sensitive in relation to fundamental rights and freedoms, deserve specific protection as the context of their processing may create important risks for the fundamental rights and freedoms.*" At the same time, the regulation acknowledges that an absolute prohibition on sharing sensitive medical data is not realistic. The sharing permission should be granted for serious threats to health, prevention of communicable diseases, and scientific research purposes. Therefore, the hospitals can jointly build an anonymized database to protect privacy of their patient records. Given the symptoms exhibited in a patient, the practitioner can predict whether the patient is likely to have a disease. However, finding the optimal balance between privacy and utility for privacy-preserving machine learning systems still remains a challenge issue.

Organization

The rest of this chapter is organized as follows: The background and preliminaries for this research are presented in Section 2. We describe the emerging technologies for privacy protection in machine learning in

Section 3. We provide a state-of-the-art based on decision tree in Section 4, followed by the conclusion in Section 5.

BACKGROUND AND PRELIMINARIES

Privacy-Preserving Machine Learning Model

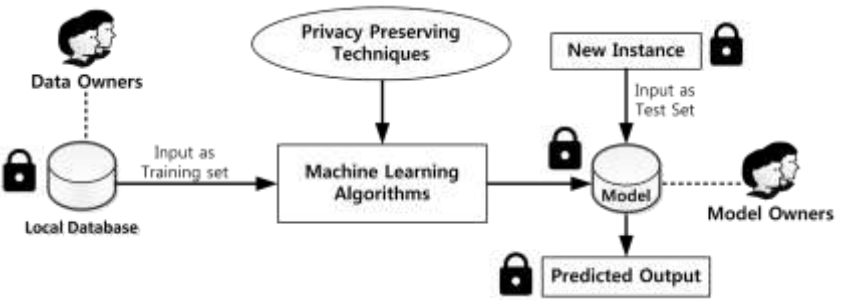


Figure 1. Privacy-preserving machine learning model.

In general, there are four attack points in machine learning system that required privacy protection (as illustrated in Figure 1):

- Some machine learning systems involve training data that is sensitive, such as the medical histories of patients in a clinical trial. From the legal aspects, the collection and processing of sensitive data using machine learning algorithms has been neglected (Malle, Kieseberg, Weippl, & Holzinger, 2016).
- Many machine learning models in recent years have been released to the public (i.e., as an open source software) due to models deployment on the cloud infrastructure. The machine learning models may be deemed confidential (e.g., fraud detection algorithms) because it can leak information about the individual data records on which they were trained. In specific, the attacker can correctly determine whether a given record was part of the

model's training data set. Furthermore, a model may inadvertently and implicitly store some of its training data.

- The new instance can be sensitive for an individual who wants to make a prediction using the trained model.
- The predicted output from the machine learning model could lead to a breach of privacy. For example, there is a potential risk for identity leakage when data-mining algorithms are used to find patterns in the predicted outputs.

The existing literature on privacy protection in machine learning mostly focuses on how to address the following objectives (Reza & Shmatikov, 2015):

1. To protect privacy of the data used for learning a model (or as input to an existing model).
2. To protect privacy of the machine learning model (including the intermediate results).
3. To protect privacy of the model's output.

Threat Models

To further motivate the need for privacy protection in machine learning, we identify several practical attacks in the literature that target machine learning systems. These attacks normally violate privacy by either inferring sensitive information in the trained models, or reduce security by polluting the prediction results of anomaly detection system (Cao & Yang, 2015).

1. **Inference attacks**. In many cases, the attacker may have some background knowledge about the population from which the target model's training data set was drawn. By exploiting the data's lineage, an attacker can illegitimately gain knowledge about a target individual. For instance, the attacker can perform membership

inference queries (Shokri, Stronati, Song, & Shmatikov, 2016) to know whether a particular training point was used to train the model by observing the prediction results.

2. **Training data pollution attacks**. To mislead the machine learning algorithms, an attacker may inject carefully polluted data samples into a learning system. This will cause the machine learning algorithms to output incorrect feature set and model. Subsequently, it may generate too many false positives in the system that is difficult for the analyst to correctly identify them.
3. **Test data poisoning attacks**. Attackers may modify the test data to avoid the detection of malicious samples on a well-trained model. For instance, an adversary may attempt to evade a deployed system at test time by carefully manipulating attack samples (Biggio et al., 2013).
4. **Model extraction attacks.** Machine learning models can be stolen and are vulnerable to reverse engineering (to extract the parameters of a model) (Tramer, Zhang, Juels, Reiter, & Ristenpart, 2016). When an adversary obtains black-boss access to some target model, it can learn a model that closely approximates to the original model. A quantitative approach on how machine learning models leak information about their training data sets can be found in (Shokri et al., 2016).

Security Properties

The security properties of implementing privacy-preserving machine learning are as follows:

- ***Strong anonymity***: training set should be anonymized properly such that nobody can identify an individual based on a targeted record.
- ***Confidentiality***: no data owner can learn the other owners' local database during model training. In addition, the model owners should not be able to identify the owner for any training set input.

- ***Privacy***: no party should learn anything more than its prescribed output. In particular, the only information that should be learned about other parties' inputs is what can be derived from the output itself. For example, the model owners only obtain the learned model without knowing the original record of any individual.
- ***Correctness***: Each party is guaranteed that the trained machine learning model is correct. To continue with the example of disease prediction learning, this implies that the hospitals (data owners) can guarantee that their local database is used in the learning, and no party including the model owner can alter this. To do so, the data owner can use any record to predict expected changes to the output for particular changes to the input (e.g., metamorphic testing (Xie et al., 2011)).
- ***Guaranteed output delivery***: Corrupted parties should not be able to prevent honest parties from receiving their output (e.g., predicted output). In other words, the adversary should not be able to disrupt the computation by carrying out a "denial of service" attack.
- ***Non-repudiation***: no data owner can change his or her mind by modifying the input for training set.

EMERGING TECHNOLOGIES FOR PRIVACY PROTECTION IN MACHINE LEARNING

A straightforward solution for privacy protection in machine learning is to use a trusted third party (TTP) to collect data from all data owners. Then the TTP functions as the central repository to perform the machine learning tasks without revealing any sensitive information to other parties. However, the level of trust in this solution is not acceptable because in the real world, a totally trusted party does usually not exist. In this section, we will discuss some emerging technologies that can help in privacy protection for machine learning tasks.

Anonymization and Perturbation

In the healthcare scenario, removing personally identifiable information (PII) such as name and home address from each patient record in the database is not an effective method of protecting user privacy in a fine-grained manner. Many anonymization schemes offer ways that help with the protection of privacy regarding personal data and sensitive information used for machine learning. For instance, data anonymization is a possible solution to protect the privacy of users (Byun, Sohn, Bertino, & Li, 2006). In 1998, Sweeney and Samarati proposed the k-anonymity model to ensure that each released data is indistinct from at least $(k-1)$ other data (Samarati & Sweeney, 1998). Although k-anonymity can be used to address the linking attack (Sweeney, 2002), but it is vulnerable against background-knowledge attacks (Machanavajjhala, Kifer, Gehrke, & Venkitasubramaniam, 2007). To complement the k-anonymity model, a privacy model known as "l-diversity model" was proposed in (Machanavajjhala et al., 2007). This model requires the representation of the sensitive attributes in the released data set with at least l "well-represented" values. In (Wong & Kim, 2015), a new notion known as k_i–anonymity has been proposed to allow the user to choose their preferred anonymity level during the data collection. A survey of the recent attacks and privacy models in data publishing can be found in (Fung, Wang, Chen, & Yu, 2010).

Data perturbation (Kargupta, Datta, Wang, & Sivakumar, 2005) is a common approach in the areas of statistical disclosure control that can be used for protecting the privacy of individual data instances. Through data perturbation, the original (private) data set is perturbed and the result is released for data analysis (Chaudhuri, Monteleoni, & Sarwate, 2011). Data perturbation has been adopted in varies machine learning systems such as learning of Logistic Regression, Support Vector Machine (SVM) and Linear Regression Models. Other than perturbing the training data, some works choose to perturb the learning algorithm (Chaudhuri & Monteleoni, 2008), or the objective function in the model (Zhang, Zhang, Xiao, Yang, & Winslett, 2012).

Cryptography

Cryptography is a promising technique that can be used to protect the intermediate results obtained in the machine learning. In 1982, Andrew Yao introduced the first two-party computation protocol (also known as the "millionaires' problem") in (Yao, 1982). He sought a way that can allow two individuals to compare their wealth without either having to reveal the extent of their wealth to each other. Since then, many secure multi-party computation (SMC) protocols have been proposed in the literature. SMC has been used for learning decision trees (Lindell and Pinkas, 2000), linear regression functions (Du, Han, & Chen, 2004), association rules (Vaidya & Clifton, 2002), Naive Bayes classifiers (Vaidya, Kantarcıoğlu, & Clifton, 2008), and k-means clustering (Jagannathan & Wright, 2005). As proved by Goldreich et al. in (Goldreich, Micali, & Wigderson, 1987), a secure solution exists for any functionality that can be represented as a combinatorial circuit; however, the generic construction of this circuit evaluation is somehow inefficient when a large number of parties are considered because the computational cost for a large input can be very high. Often, SMC is used as the basic component in privacy preserving data mining (PPDM).

In (Pathak, 2012), the author adopted homomorphic cryptosystem and hash functions to obfuscated data for speech processing algorithms. Another work in (Yonetani & Kitani, 2017) utilizes homomorphic cryptosystem to aggregate and encrypt private data in learning visual classifiers.

Differential Privacy

In recent years, differential privacy (ε-differential privacy) has emerged as an important paradigm for statistical analyses and machine learning systems. Differential privacy is a strong notion of privacy that guarantees the privacy protection in the presence of arbitrary auxiliary information. Intuitively, it aims to limit the information leakage from the

output while a small change on the inputs (Dwork, 2006). In particular, the adversary is not able to infer any information of the targeted individual with high confidence.

To achieve ε-differential privacy, one can add Laplace noise into the results of queries and analyses. For example, the construction of privacy-preserving classifier in (Agrawal and Srikant 2000) is based on the statistical noise that used to hide the individual data entries. Recently, (Fan & Jin, 2015) proposed a practical framework for data analytics which can provide differential privacy guarantees to the data owners. Their idea is to sample a fixed number of records from each user to be used for data mining tasks. This approach can alleviate the high perturbation errors caused by differential mechanism. In (Rubinstein, Bartlett, Huang, & Taft, 2012), the authors introduced weight regularization in objective loss function of SVM. The adoption of differential privacy in machine learning is practically important as it maintains the anonymity of individual training data. Since differential privacy is a promising formal approach to data privacy, it has been used extensively in a timely and promising area of machine learning, i.e., deep learning (Liu, Jiang, Chen, & Badokhon, 2016; Phan, Wang, Wu, & Dou, 2016; Shokri & Shmatikov, 2016).

Privacy-by-Design

Privacy-by-Design is an approach to systems engineering which takes privacy into account throughout the whole engineering process (Cavoukian, 2009). This concept takes human values into account in a well-defined manner throughout the design process. The adoption of principles in privacy-by-design approach can ensure privacy protections are built by default into a machine learning system from the beginning. An investigation about the effects of embodiment and transparency on privacy and user experience was studied in (Vitale et al., 2017).

Machine Unlearning

Recently, an interesting approach known as machine unlearning (or simply unlearning) was proposed in by Cao & Yang (2015). The idea of machine unlearning is to transform learning algorithms used by a system into a summation form. To forget a training set, the proposed approach simply updates the summations and then computes the updated model. Machine unlearning allows the learning algorithms depend only on the summations instead of individual data. This approach can eliminate the dependency between the learning algorithm and the training set. The motivation of machine unlearning can be seen from three aspects. From a privacy perspective, users often want the system to forget their data and lineage. Data's lineage is refer to the information derived by the system from that data together form a complex propagation network. From a security perspective, the anomaly detector must forget the injected data by the attacker who manually injects crafted data into the training data set. From a usability perspective, the user can remove noise and incorrect entries from the analytics results.

Summary

There are still other possible technologies that can be used to achieve the same goal. For instance, data minimization is a practice to limit the collection of sensitive data (only relevant and necessary data can be collected) in order to minimize the privacy leakage to unauthorized parties (Pfitzmann & Hansen, 2010). It may be enforced within an organization through some security policies.

We summarize the emerging technologies that can help in privacy protection for machine learning tasks in Figure 2.

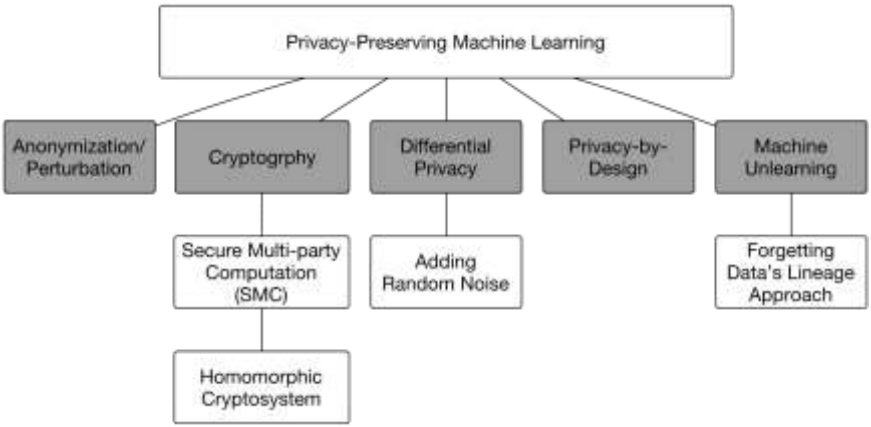


Figure 2. Summary of the emerging technologies for privacy-preserving machine learning.

SATE OF THE ART: DECISION TREE ALGORITHMS FOR PRIVACY-PRESERVING DATA CLASSIFICATION

There are many machine learning algorithms have been extended to consider the protection of data privacy. However, in this chapter, we are keener to perform a thorough analysis into a need of privacy protection especially on a white box learner – decision tree.

Decision Tree Model

Decision tree (Quinlan, 1986) is a predictive model which is commonly used to examine an item based on its attribute values and to decide its final value. It is an efficient nonparametric method, and well representing by a hierarchical data structure based on the divide-and-conquer strategy, where it splits the source set into subsets based on attribute value test, usually either information gain ratio, or Gini index. This process will be repeated until a defined situation met, and is widely known as recursive partitioning, or recursive binary splitting. There are a

number of notable extensions of decision tree, such as Iterative Dichotomiser 3 (ID3) (Quinlan, 1986), and C4.5 (Salzberg, 1994) (Quinlan, 1996). C4.5 is an extension from ID3, and improved the ID3 in terms of it is able to handle both continuous and discrete attributes as well as missing attribute values.

In a tree model, the observations of attributes (features) are represented by branches, and the decision is represented by using a leaf (usually indicate the class label). A model is depicted in Figure 3. In this white box architecture, classification information is shown clearly (Wang, Liu, Pedrycz, & Zhang, 2015). All the information carried in branches explaining how they contribute to the last decision (class). From the perspective of machine learning, this is definitely helpful for the user to understand how a class has been made. However, this "transparent" machine model might be also exposing too much of information when a data set is carrying some sensitive data.

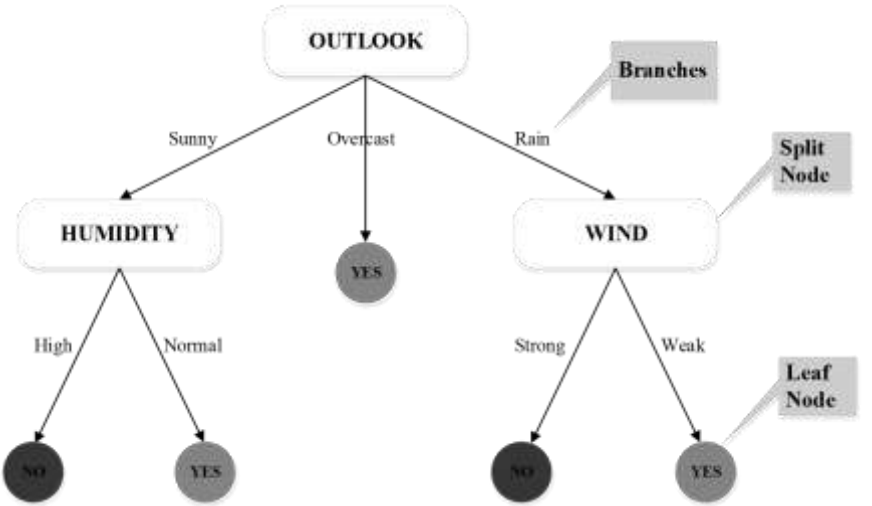


Figure 3. A model of decision tree learning (redrawn from (Mitchell, 1997)).

State-of-the-Art: Decision Tree with Privacy Protection

Randomization

The first adoption of decision tree classifier in privacy preserving context was contributed by Agrawal & Srikant (Agrawal and Srikant, 2000). They introduced value distortion methods through two random distributions, (i) uniform, and (ii) Gaussian. These methods were used to distort the original data distribution, thus creating a new set of randomized data. The randomized data will be used throughout the decision tree learning, thus the privacy of data is ensured through the introduced distortion methods. Definitely, the reconstruction of the original data distribution from the randomized data is needed at the end of classification. The tree splitting is based on Gini index, and tested on a set of their independent synthetic data set. The evaluation results substantiated that the accuracy of the decision tree performed on randomized data is comparable to its performances when conducting on original data.

Another randomization scheme was proposed by Du and Zhen, where it covered two components (Du & Zhan, 2003). The first component is a multivariate data disguising technique, and mainly used to randomize the data collection; whereas the second component is the modified ID3 algorithm which was designed to handle the randomized data resulted from first component. The experimental tests was conducted on a UCI repository data set of "Adult" with some sensitive information such as salary.

Fong et al. introduced a different randomization approach to protect the centralized data sets (Fong & Weber-Jahnke, 2012). This centralized samples were sanitized before releasing to third parties without degrading the classification accuracy. By sanitizing the samples, a set of unrealized data can be created. With this design, the ID3 tree can be directly built from the sanitized sample without the need of reconstructing the original data set. This proposed randomization scheme can be applied even during the data collection process. By doing so, the model can automatically omitted the sensitive attributes which are not really meaningful for decision making. The same idea was adopted by Liu et al. (2009) and

Baghel (2013) respectively in a C4.5 algorithm. The modified C4.5 algorithm which learned from the perturbed and unrealized data sets eventually surpassed the original C4.5 algorithm in classification performances.

Secure Two-Party Computation

Lindell and Pinkas considered a scenario when two parties desired to run a machine learning algorithm on the union of their data sets (Lindell and Pinkas, 2000). Assuming that each of them are owning their own confidential data sets, and they do not wish the learner to reveal any unnecessary information at the end of classification. To ensure the privacy of their independent data sets, Lindell and Pinkas adopted the approach of secure two-party computation setting (cryptography) based on Yao's protocol (Yao, 1982), where the majority of the computation should be done by the individual parties independently. This composition of private protocols were used to compute distributed ID3, with the index of information gain. Lindell and Pinkas performed a thorough complexity analysis to compare the performance of private distributed ID3 to the original distributed ID3 in terms of the usage of bandwidth and computation resources, and concluded that they are somehow quite comparable.

An extensive discussion on combination usage of cryptographic techniques and ID3 is later enclosed by Pinkas (Pinkas, 2002). On the other hand, Du and Zhan (Du & Zhan, 2002) proposed a scalar product protocol to work on an untrusted third-party server – commodity server. This third party server acts as an intermediary server for both parties to upload their vertical partitioned private data, and run the ID3 algorithm. The proposed scalar product protocol was used to compute Gini index and entropy, by taking independent vectors submitted by both parties into calculation.

Extending the work by Lindell and Pinkas (Lindell and Pinkas, 2000), Brickell and Shmatikov proposed a new cryptographically secure protocol in constructing a secure classification tree based on Gini index (Brickell & Shmatikov, 2009). The proposed protocol effectively works between a party and a server. A party can submit his desired input to the server, and

these inputs can be the data attributes (columns), the class attribute, and, even the instances (rows) to be used during a decision tree model construction. To ensure the privacy, the server is ensured to learn nothing from the submitted attributes or instances from each party, including the number of submitted vectors. The proposed method was tested on UCI repository – car data set. To enable the handling of both discrete and continuous attribute values, some researchers extending the scheme into C4.5 algorithm instead of ID3 (Xiao et al. 2006; Shen, Shao, and Yang 2009; Shen, Shao, and Huang, 2009).

Secure Multi-Party Computation

Secure multi-party computation involving more than two parties is first introduced by Xiao et al., where they proposed five sub-protocols for the secure multi-party *xLogx* protocol (Xiao, 2005). These five sub-protocols were invented based on the data disguise techniques, namely (i) a secure two-party multiplication protocol, (ii) the secure two-party reverse multiplication protocols, (iii) multi-party reverse multiplication protocols, (iv) a secure multi-party x*Logx* protocol, and lastly (v) a secure multiparty FindMax protocol. All of these proposed protocols were embedded into ID3 algorithm in computing the information gain as well as entropy, and tested on a horizontally partitioned data set.

Considering more than two parties in a secure multi-party computation with vertically partitioned data set, Vaidya and Clifton invented a secure ID3 algorithm by incorporating a generalized privacy preserving variant (Vaidya & Clifton, 2005). The proposed method extends the ID3 in such a way that they distributed the nodes (or attributes of the data set), where only the party who own the attributes can view them, and only the party who requesting classification (e.g., a master site) can view the root node of the ID3 tree. Based on the security analysis, authors projected that this proposed secure ID3 is comparable to the original ID3 algorithm once the tree has been built.

Looking into the same perspective, Emekci et al., proposed a secure ID3 distributed algorithm by using Shamir's secret sharing scheme to compute the summation of the secret values over n parties (where $n > 2$) (Emekci, Sahin, Agrawal, & El Abbadi, 2007). The correctness of the final results was validated through a series of validation methods. The proposed algorithm was tested on two data sets, Mfeat-feature Data set and Nursery Data set. The classification accuracies are greatly improved when more training data are involved, and authors postulated that this can be achieved through the union of data which belong to n parties.

Utilizing the benefits of the Shamir's secret share, Sheela proposed an efficient secure method to find the cardinality of scalar product by using less communication and computation cost (Sheela, 2013). This scalar product was used to select the attribute with the highest information gain from ID3 algorithm.

Homomorphic Encryption

Zhan et al. adopted homomorphic encryption and digital envelope technique to collaborate ID3 classification without sharing the private data which belong to other parties (Zhan, 2007). In general, homomorphic encryption works in such a way where it assumes each party owns a private data set, which may contains a number of attributes. During vertically collaboration, each party can submit their own attribute vectors, and a single vector will be computed based on the submitted vectors. This single vector will be shared by all parties, and thus, can be further protected through a homomorphic encryption.

The adoption of homomorphic public key encryption is also discussed by Vaidya et al., in which they applied this encryption method to a Weka-packaged ID3 algorithm (Vaidya, Clifton, Kantarcioglu, & Patterson, 2008). The proposed algorithm was tested on two public data sets from UCI repository: weather data set and car data set. The experimental results shown that the classification accuracy is quite promising, and the only drawback of this proposed algorithm is that its computation time is slightly slower compared to the original ID3 algorithm.

Limitations

Obviously, the cost and complexity of computation as well as communication are greatly hampered especially when the size and dimension of data are huge. An interesting fact will be the trade-off between the classification accuracy and the privacy of data. On top of this, when considering the environment of secure multi-party computation, scalability is also an arising issue when the number of collaborative parties increases.

In general, the goal of most of the existing works is to provide a privacy solution that is practical for real world applications. Unfortunately, most of them are not practical due to high complexity, high computational cost or limited data access. For example, data minimization can limit the machine learning systems from learning good models since a large part of sensitive data will be filtered from the training data set. In differential privacy, although it can provide strong privacy guarantees, but the utility of the privatized data sets diminishes due to too much noise (Muralidhar & Sarathy, 2010; Sarathy & Muralidhar, 2010).

CONCLUSION

In conventional machine learning system, there always exist a tradeoff between a model's ability to minimize bias and variance in learning. Ideally, the machine learning system aims to learn a model that both accurately captures the regularities in its training data, and at the same time generalizes well to unseen data. Unfortunately, it is not possible to achieve both requirements at the same time. On the other hand, there is also a growing interest in investigating privacy-preserving machine learning systems that provide a balance between privacy and utility (Mivule, Turner, & Ji, 2012; Xu, Yue, Guo, Guo, & Fang, 2015). However, finding the optimal balance between privacy and utility for privacy-preserving machine learning systems still remains a challenge issue. The combination of different techniques (existing techniques or new findings) should

provide strong privacy guarantees in order to enable the collection of enormous amounts of new data for machine learning purposes.

REFERENCES

Agrawal, R., & Srikant, R. (2000). Privacy-preserving data mining. *SIGMOD Rec.*, *29*(2), 439–450. Journal Article. https://doi.org/10.1145/335191.335438.

Alpaydın, E. (2010). *Introduction to machine learning*. *Methods in Molecular Biology* (2nd Edition, Vol. 1107). United States of America: Massachusetts Institute of Technology (MIT). https://doi.org/10.1007/978-1-62703-748-8-7.

Baghel, R. (2013). *Privacy Preserving Classification By Using Modified C4.5*, 124–129.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Roli, F. (2013). Evasion Attacks against Machine Learning at Test Time. *Machine Learning and Knowledge Discovery in Databases*, 8190, 387–402. https://doi.org/10.1007/978-3-642-40994-3_25.

Boillat, P., & Kjaerum, M. (2014). *Handbook on European data protection law*. Luxembourg: Publications Office of the European Union.

Brickell, J., & Shmatikov, V. (2009). *Privacy-Preserving Classifier Learning*, 128–147.

Byun, J. W., Sohn, Y., Bertino, E., & Li, N. (2006). Secure anonymization for incremental datasets. *Proceedings of the Third VLDB International Conference on Secure Data Management*. Seoul, Korea: Springer-Verlag. https://doi.org/10.1007/11844662_4.

Cao, Y., & Yang, J. (2015). Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy (pp. 463–480)*. IEEE. https://doi.org/10.1109/SP.2015.35.

Chaudhuri, K., & Monteleoni, C. (2008). Privacy-preserving logistic regression. *Advances in Neural Information Processing Systems*, 289–296. https://doi.org/10.12720/jait.6.3.88-95.

Chaudhuri, K., Monteleoni, C., & Sarwate, A. D. (2011). Differentially Private Empirical Risk Minimization. *Journal of Machine Learning Research*, *12*, 1069–1109. Retrieved from http://arxiv.org/abs/0912.0071.

Du, W., Han, Y. S., & Chen, S. (2004). Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 2004 SIAM International Conference on Data Mining* (pp. 222–233). Conference Proceedings, SIAM.

Du, W., & Zhan, Z. (2002). Building decision tree classifier on private data. *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining*-Volume 14, 1–8.

Du, W., & Zhan, Z. (2003). Using randomized response techniques for privacy-preserving data mining. *Conference on Knowledge Discovery and Data Mining*, 505–510. https://doi.org/10.1145/956804.956810.

Dwork, C. (2006). Differential privacy. *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, 1–12. https://doi.org/10.1007/11787006_1.

Emekci, F., Sahin, O. D., Agrawal, D., & El Abbadi, A. (2007). Privacy preserving decision tree learning over multiple parties. *Data and Knowledge Engineering*, *63*(2), 348–361. https://doi.org/10.1016/j.datak.2007.02.004.

Fan, L., & Jin, H. (2015). *A Practical Framework for Privacy-Preserving Data Analytics. Www'15*, 311–321. https://doi.org/10.1145/2736277.2741122.

File, I. Proposal for a Regulation of the European Parliament and of the Council on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free movement of Such Data (General Data Protection Regulation) (2012). *European Commission*.

Fong, P. K., & Weber-Jahnke, J. H. (2012). Privacy preserving decision tree learning using unrealized data sets. *IEEE Transactions on Knowledge and Data Engineering*, *24*(2), 353–364. https://doi.org/10.1109/TKDE.2010.226.

Fung, B. C. M., Wang, K., Chen, R., & Yu, P. S. (2010). Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, *42*(4), 1–53. https://doi.org/10.1145/1749603.1749605.

Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play ANY mental game. *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*. New York, New York, United States: ACM. https://doi.org/10.1145/28395.28420.

Gould, K. (2006). *The State of HIPAA Privacy and Security Compliance*, (April).

Jagannathan, G., & Wright, R. N. (2005). Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (pp. 593–599). Conference Proceedings, ACM.

Kansagara, D., Englander, H., Salanitro, A., Kagen, D., Theobald, C., Freeman, M., & Kripalani, S. (2011). Risk Prediction Models for Hospital Readmission. *JAMA*, *306*(15), 1688. https://doi.org/10.1001/jama.2011.1515.

Kargupta, H., Datta, S., Wang, Q., & Sivakumar, K. (2005). Random-data perturbation techniques and privacy-preserving data mining. *Knowledge and Information Systems*, *7*(4), 387–414. Journal Article.

Kim, K.-J., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications*, *19*(2), 125–132. https://doi.org/10.1016/S0957-4174(00)00027-0.

Kohavi, R., & Provost, F. (1998). *Glossary of Terms*, *30*(23), 271–274. https://doi.org/http://dx.doi.org/10.1016/B978-0-08-096682-3.10017-4.

Lindell, Y., & Pinkas, B. (2000). Privacy Preserving Data Mining. *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*. Conference Paper, Springer-Verlag.

Liu, L., Kantarcioglu, M., & Thuraisingham, B. (2009). Privacy preserving decision tree mining from perturbed data. *Proceedings of the 42nd Annual Hawaii International Conference on System Sciences, HICSS*, *5*, 1–10. https://doi.org/10.1109/HICSS.2009.353.

Liu, M., Jiang, H., Chen, J., & Badokhon, A. (2016). *A Collaborative Privacy-Preserving Deep Learning System in Distributed Mobile Environment*. https://doi.org/10.1109/CSCI.2016.42.

Lohr, S. (2010). Netflix Cancels Contest After Concerns Are Raised About Privacy. *The New York Times*.

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). l-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, *1*(1), 3. https://doi.org/10.1145/1217299.1217302.

Malle, B., Kieseberg, P., Weippl, E., & Holzinger, A. (2016). The right to be forgotten: Towards machine learning on perturbed knowledge bases. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9817, pp. 251–266). Springer, Cham. https://doi.org/10.1007/978-3-319-45507-5_17.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill. https://doi.org/10.1145/242224.242229.

Mivule, K., Turner, C., & Ji, S. Y. (2012). Towards a differential privacy and utility preserving machine learning classifier. *Procedia Computer Science*, *12*, 176–181. https://doi.org/10.1016/j.procs.2012.09.050.

Muralidhar, K., & Sarathy, R. (2010). Does Differential Privacy Protect Terry Gross' Privacy? *LNCS*, *6344*, 200–209.

Narayanan, A., & Shmatikov, V. (2008). Robust De-anonymization of Large Sparse Datasets. Proceedings of the 2008 IEEE Symposium on Security and Privacy. *IEEE Computer Society*. https://doi.org/10.1109/sp.2008.33.

Natwichai, J., Li, X., & Orlowska, M. (2005). Hiding Classification Rules for Data Sharing with Privacy Preservation. In *International Conference on Data Warehousing and Knowledge Discovery* (pp. 468–477). Springer Berlin Heidelberg.

Pathak, M. A. (2012). *Privacy-Preserving Machine Learning for Speech Processing*. *PhD*. https://doi.org/10.1007/s13398-014-0173-7.2.

Pfitzmann, A., & Hansen, M. (2010). *A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity*

Management. Technical University Dresden, 1–98. https://doi.org/10.1.1.154.635.

Phan, N., Wang, Y., Wu, X., & Dou, D. (2016). Differential Privacy Preservation for Deep Auto-Encoders : An Application of Human Behavior Prediction. *AAAI*, 1309–1316.

Pinkas, B. (2002). Cryptographic techniques for privacy-preserving data mining. *ACM SIGKDD Explorations Newsletter*, *4*(2), 12–19. https://doi.org/10.1145/772862.772865.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*(1), 81–106.

Quinlan, J. R. (1996). Improved use of continuous attributes in C4. 5. *Journal of Artificial Intelligence Research*, *4*(1996), 77–90. https://doi.org/10.1613/jair.279.

Reza, S., & Shmatikov, V. (2015). Privacy-Preserving Deep Learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security* (pp. 1310–1321). ACM.

Rubinstein, B. I. P., Bartlett, P. L., Huang, L., & Taft, N. (2012). Learning in a Large Function Space: Privacy-Preserving Mechanisms for SVM Learning. *Journal of Privacy and Confidentiality*, *4*(1), 65–100. Retrieved from http://arxiv.org/abs/0911.5708.

Salzberg, S. (1994). Book Review: C4.5: Programs for Machine Learning. *Machine Learning*, *1*(16), 235–240.

Samarati, P., & Sweeney, L. (1998). Generalizing data to provide anonymity when disclosing information (abstract). *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Seattle, Washington, United States: ACM. https://doi.org/10.1145/275487.275508.

Sarathy, R., & Muralidhar, K. (2010). Some additional insights on applying differential privacy for numeric data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 6344 LNCS, pp. 210–219). https://doi.org/10.1007/978-3-642-15838-4_19.

Sheela, M. A. (2013). *A novel privacy preserving decision tree induction*, (Ict), 1075–1079.

Shen, Y., Shao, H., & Huang, J. (2009). Research on Privacy Preserving Distributed C4. 5 Algorithm. *Intelligent Information Technology Application Workshops, 2009. IITAW '09. Third International Symposium on*, 216–218. https://doi.org/10.1109/IITAW.2009.81.

Shen, Y., Shao, H., & Yang, L. (2009). Privacy preserving C4.5 algorithm over vertically distributed datasets. Proceedings - International Conference on Networks Security, Wireless Communications and Trusted Computing, *NSWCTC* 2009, *2*, 446–448. https://doi.org/10.1109/NSWCTC.2009.253.

Shokri, R., & Shmatikov, V. (2016). *Privacy-preserving deep learning. 2015 53rd Annual Allerton Conference on Communication, Control, and Computing*, Allerton 2015, 909–910. https://doi.org/10.1109/Allerton.2015.7447103.

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2016). *Membership Inference Attacks against Machine Learning Models.* Retrieved from https://arxiv.org/pdf/1610.05820.pdf.

Sweeney, L. (2002). k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, *10*(5), 557–570. https://doi.org/10.1142/s0218488502001648.

Tramer, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing Machine Learning Models via Prediction APIs. In *USENIX Security* (Vol. 94, p. 34301). https://doi.org/10.1103/PhysRevC.94.034301.

Vaidya, J., & Clifton, C. (2002). Privacy preserving association rule mining in vertically partitioned data. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Conference Paper, Edmonton, Alberta, Canada: ACM. https://doi.org/10.1145/775047.775142.

Vaidya, J., & Clifton, C. (2005). *Privacy-Preserving Decision Trees*, (312357), 139–152.

Vaidya, J., Clifton, C., Kantarcioglu, M., & Patterson, A. S. (2008). Privacy-preserving decision trees over vertically partitioned data. *ACM Transactions on Knowledge Discovery from Data*, *2*(3), 1–27. https://doi.org/10.1145/1409620.1409624.

Vaidya, J., Kantarcıoğlu, M., & Clifton, C. (2008). Privacy-preserving naive bayes classification. *The VLDB Journal—The International Journal on Very Large Data Bases*, *17*(4), 879–898. Journal Article.

Vitale, J., Tonkin, M., Ojha, S., Williams, M.-A., Wang, X., & Judge, W. (2017). *Privacy by Design in Machine Learning Data Collection: A User Experience Experimentation*. Retrieved from https://aaai.org/ocs/index.php/SSS/SSS17/paper/viewFile/15305/14583.

Wang, X., Liu, X., Pedrycz, W., & Zhang, L. (2015). Fuzzy rule based decision trees. *Pattern Recognition*, *48*(1), 50–59. https://doi.org/10.1016/j.patcog.2014.08.001.

Wong, K.-S., & Kim, M. H. (2015). Towards a Respondent-Preferred k_i Anonymity Model. *Frontiers of Information Technology & Electronic Engineering*, *16*(9), 720–731. https://doi.org/10.1631/FITEE.1400395.

Xiao, M. (2005). *Privacy Preserving ID3 Algorithm over Horizontally Partitioned Data*, 0–4.

Xiao, M. J., Han, K., Huang, L. S., & Li, J. Y. (2006). Privacy preserving C4.5 algorithm over horizontally partitioned data. *Proceedings - Fifth International Conference on Grid and Cooperative Computing, GCC* 2006, 78–85. https://doi.org/10.1109/GCC.2006.73.

Xie, X., Ho, J. W. K., Murphy, C., Kaiser, G., Xu, B., & Chen, T. Y. (2011). Testing and validating machine learning classifiers by metamorphic testing. In *Journal of Systems and Software* (Vol. 84, pp. 544–558). https://doi.org/10.1016/j.jss.2010.11.920.

Xu, K., Yue, H., Guo, L., Guo, Y., & Fang, Y. (2015). Privacy-Preserving Machine Learning Algorithms for Big Data Systems. 2015 *IEEE 35th International Conference on Distributed Computing Systems*, 318–327. https://doi.org/10.1109/ICDCS.2015.40.

Yao, A. C. (1982). Protocols for secure computations. Proceedings of the 23rd Annual Symposium on Foundations of Computer Science. *IEEE Computer Society*. https://doi.org/10.1109/sfcs.1982.88.

Yonetani, R., & Kitani, K. M. (2017). *Privacy-Preserving Visual Learning Using Doubly Permuted Homomorphic Encryption. arXiv:1704.02203 [cs.CV]*.

Zhan, J. (2007). Using homomorphic encryption for privacy-preserving collaborative decision tree classification. *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining, CIDM* 2007, (CIDM), 637–645. https://doi.org/10.1109/CIDM.2007.368936.

Zhang, J., Zhang, Z., Xiao, X., Yang, Y., & Winslett, M. (2012). Functional mechanism: regression analysis under differential privacy. *Proceedings of the 38th International Conference on Very Large Data Bases*, 5(11), 1364–1375. Retrieved from http://dl.acm.org/citation.cfm?id=2350229.2350253.

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 3

KEY DISTRIBUTION AND MANAGEMENT IN CRYPTOGRAPHY

Bachir Bendrissou and Yean Li Ho*
Faculty of Information Science and Technology,
Multimedia University, Melaka, Malaysia

ABSTRACT

Cryptographic key distribution and management is a very crucial part of any cryptographic system when it comes to securing data. Problems related to cryptographic key distribution and management are difficult to solve and vulnerable to exploit at the same time. Therefore, they are targeted by attackers. It is the aim of this book chapter to provide a practical survey of the techniques and challenges of key distribution and management in cryptography. The whole chapter is divided into four main parts. The first part contains an introduction to key distribution and management in cryptography. The second part introduces the fundamental algorithms used in key distribution, most notably Diffie–Hellman and RSA key exchange. We will also examine two widely used infrastructures, Public Key Infrastructure (PKI) and Identity-based public key cryptography (ID-PKC). In the third section, we discuss some current

* Corresponding Author, Email: ylho@mmu.edu.my.

key distribution protocols, namely TLS and PAKE protocols. We will dedicate the fourth part to study the general challenges of cryptographic key management focusing on challenges that exist under cloud technology. Finally, we will offer some analysis and solutions with respect to the challenges reported in the previous section.

Keywords: key distribution and management, public key infrastructure, identity-based cryptography, transport layer security, password-based authenticated key exchange, cloud key exchange

INTRODUCTION

At its core, cryptography in today's Information Technology (IT) security industry is the subject of continuous inquiry surrounding the process of encryption and decryption of information. Hence, cryptographic algorithms are developed, tried and then applied to secure information so that it is only accessible by authorized users. Information security standards are most useful in promoting reliable and interoperable mechanisms that can help prevent the inadvertent introduction of security weaknesses.

Since the advent of computers, the sheer processing power and the effectiveness of cryptanalysis algorithms (systematic code breaking) have all been steadily increasing. Naturally, most cryptographic methods used today have become vulnerable and inadequate. This situation called for a better cryptographic model, one that can provide adequate security. However, as advancements are achieved in the field of cryptography, old algorithms become vulnerable and insecure. In principle, computer security standards promote the use of protocols (specific algorithms and their implementations) that are designed to be sufficiently secure for some given task.

Key management is basically the direct application of cryptography in information security field. Key management essentially sits between cryptography and cryptographic entity authentication. Due to the fact that the security level of encrypted information depends primarily on the

secrecy of the key that decrypts it, computer security standards thus, take into great consideration the various environments within which cryptographic keys are established and distributed. By doing so, information security protocols are able to ensure that any subsequent information encrypted with keys established and distributed securely will also be secure, thus enhancing the security of the whole system.

FUNDAMENTALS

Asymmetric Key Encryption (Public-Key)

The increasing adoption of computers led to the invention of an encryption system that would rely on a predetermined key. Recent cryptographers understood that if they wanted to send a message securely without having previously met the recipient, they would need a system that uses two different keys, one for encryption and another for decryption. In comparison with systems which adopt symmetric key encryption, this system can be thought of as a lock that has two corresponding keys, one key for engaging the lock and another key for disengaging the lock.

Diffie-Hellman Key Exchange

The Diffie-Hellman (DH) Key Exchange is a widely used cryptographic protocol which permits two parties with no prior knowledge of each other to establish and share a secret key. This protocol is normally used in symmetric key cipher systems. The Diffie-Hellman Key Exchange was initially published in 1976 by Whitfield Diffie and Martin Hellman (Diffie & Hellman, 1976).

The computation in the Diffie-Hellman Key Exchange relies on exponential functions, which operate much faster than discrete logarithms under normal conditions. When deployed properly, the Diffie-Hellman key exchange protocol gives two parties the same key without having actually

transmitted it. The robustness of this algorithm however, depends on the amount of time required to compute a discrete logarithm of the public keys transmitted (Diffie & Hellman, 1976).

Figure 1 illustrates the steps for establishing a key using the DH Key Exchange. Alice, who wants to establish a key with Bob, first has to choose a secret integer "a," a base "g," and a prime number "p." Bob decides his secret number "b." Now after sending the public keys/numbers, each party is able to compute the key "K" on their own. It can be observed that "K" is never sent over the network. Moreover "K" was merely a result of both Alice and Bob's computations and was not in any way predetermined. This process has a crucial advantage as it allows both parties to arrive at the same key without ever having to see each other. One disadvantage of the DH key exchange is that it does not include the encryption function. The DH algorithm does not allow any predetermined message to get inserted to it. The number transmitted over the channel is simply the result of a sequence of computations which is hard to decompose. An attacker wishing to unravel the value of "K" must compute a logarithm of "A" or "B". However, if numbers "a," "b," and "p" are chosen to be extremely large, then it could theoretically take billions of years to compute the logarithm of "A" or "B."

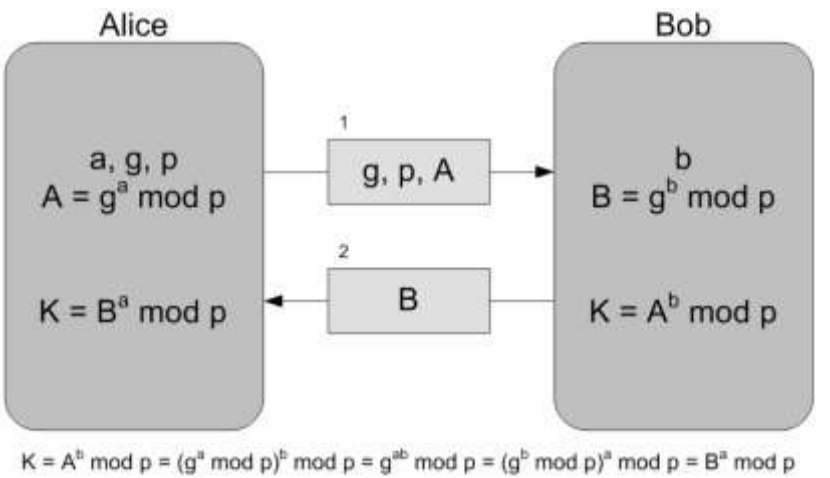


Figure 1. Diffie-Hellman key exchange protocol (Diffie & Hellman, 1976).

RSA Encryption

Despite the useful characteristics of the DH Key Exchange protocol discussed in the previous section. The protocol remains unable to transmit a secret message. Ron Rivest, Adi Shamir, and Leonard Adleman managed to develop a system (Rivest, Shamir, and Adleman, 1978) similar to the DH protocol (Diffie & Hellman, 1976) but differ in that a message could be both embedded and transmitted.

The RSA encryption, named after the inventors' surnames, relies mainly on the fact that multiplication and exponentiation are much faster than prime factorization. The whole algorithm is built from two large prime numbers. These prime numbers go through a sequence of computations to give a public key and a private key. These keys can be used many times after they have been generated. Ordinarily, the private key is kept secret while the public key is made public. The public key can then be used by any participating entity to encrypt a message and sent to the public key's owner. Now the receiver can decrypt the encrypted message using his private key that is known only to him. Note that only the person having knowledge of the private key can decrypt the message. The modulus in RSA encryption is a very special number. It is generated by multiplying two large prime numbers together. For an attacker to break the ciphertext, he is obliged to reproduce the original two primes given the modulus, by prime factoring the modulus. The robustness of RSA encryption relies largely on the difficulty of reversing the modulus generation operation, that is, to reproduce the prime factors that were initially used to compute the modulus. In the field of e-commerce applications, RSA encryption is the most common type of asymmetric key encryption.

A simple example to illustrate how RSA algorithm works would be as follows. In order for Alice to generate her RSA keys. Alice would select two primes: $p = 11$ and $q = 13$ as RSA keys. She then can compute the modulus n: $n = p \times q = 143$ and the totient of n: $\phi(n) = (p-1) \times (q-1) = 120$. Alice then chooses 7 as her RSA public key e. Using the Extended Euclidean Algorithm, Alice can compute her RSA private key. The private

key in this case would be 103. Bob who wishes to send Alice an encrypted message M would have to obtain her RSA public key (n,e), In this case it's (143, 7). We assume the number 9 is his plaintext message. Subsequently, the message "9" is encrypted into ciphertext C as follows:

$C = M^e \bmod n = 9^7 \bmod 143 = 48$

After Bob's encrypted message is received by Alice, Alice now can decrypt it by using her RSA private key (d, n) as follows:

$M = C^d \bmod n = 48^{103} \bmod 143 = 9$

An RSA key can also be used to generate a digital signature of a given message. The scenario goes like this: Alice would generate a message digest (also known as a hash value) of her message that is intended to Bob. Then using her RSA private key, Alice would encrypt the hash value and append it to the message. After receiving the message, Bob can use Alice's public key to confirm that the message was truly sent by Alice and that the message was not altered during transmission. To do that, Bob would need to compare the retrieved hash value with the hash of the original message. A match indicates that Alice is the real owner of the message (authentication and non-repudiation) and that the message received was not modified during transmission. In other words, the integrity of message is intact. Additionally, Alice can ensure the confidentiality of her message by encrypting it with Bob's public key. A digital certificate typically contains information that can help identify the certificate's owner as well as the owner's public key. A certificate is issued and signed by a certificate authority. The primary role of certificate authorities is to distribute public keys and verify identities.

Public Key Infrastructure (PKI) and Identity-Based Public Key Cryptography (ID-PKC)

We attempt to provide a technical overview of the traditional PKI model as well as the ID-PKC model in this section.

Essentially, both models are variants of asymmetric cryptography. Diffie and Hellman in 1976 were the first to introduce asymmetric cryptography. Before that, symmetric cryptography was the only model being used in cryptographic systems, where both parties had to share a predetermined key. Under the asymmetric cryptography model, two related keys (public and private) are generated for every user. The private key, as the name suggests is known only to its owner, while the public key is made known to all. The private key has two main functions, to decrypt data that is encrypted with the related public key and to digitally sign documents, which can be verified with the corresponding public key.

The distinction between the two versions of the asymmetric model is drawn by the way the keys are generated. In the traditional asymmetric model which is the base of most PKI systems, none of the information identifying the key pair is used to generate the key pair. For that reason, a certificate is required to bind the public key with its relevant usage.

On the contrary, an ID-PKC system typically generates key pairs from data which identifies how the key is to be used. For instance, a user's identity in respect to the system being used can be used to derive the key pair. We intend to extend our understanding of PKI and ID-PKC in the upcoming paragraphs.

Public Key Infrastructure (PKI)

Currently, Public Key Infrastructure (PKI) is considered to be the primary means by which asymmetric cryptography systems are deployed. In this book chapter, when we say PKI, we normally refer to infrastructures that support the use of traditional asymmetric cryptographic algorithms, like RSA (Rivest, Shamir and Adleman, 1978). Since public keys are public by nature, their integrity becomes of great concern. For that reason, a certificate is always required to prove such integrity. We can define PKI as the infrastructure that is responsible for managing and storing keys and certificates.

The core components of a PKI, as discussed by Paterson and Price (2003) are:

Certificate Authority (CA): It is the entity that is responsible for generating certificates. It is particularly responsible for ensuring that the correct key is attached to its corresponding certificate. Another one of its tasks is to verify the certificate content.

Registration Authority (RA): The main role of the RA is to make sure that users who receive certificates are registered in its registry and are authentic. Sometimes, the CA and RA are embedded within a single entity.

Certificate Storage: Certificates and other related updates such as Certificate Revocation Lists are usually stored in a CA managed database.

Software: The software responsible for managing the certificates must be able to access the information contained in a certificate with respect to the security policy specifications.

Policies and Procedures: There is a strong necessity to ensure the proper flow and work of PKI functions. Because of such necessity, Certification Practice Statements (CPS) and the Certificate Policy (CP) are commonly used to provide definitions and guidelines on how certificates ought to be created, stored and managed. The function of certificates with respect to the overall security architecture is also defined by both CPS and CP. One distinctive property of traditional PKIs is that the system owner can choose where key pairs should be generated. There are mainly two choices, either the CA generates the key for the user, or the user generates the key on its own and share its public key with a CA to validate and certify. This choice is primarily determined by the security policy. It is also determined by the purpose of usage. For instance, in the case that a key is intended to be used for signing documents for non-repudiation purposes, then the client himself should generate the key. Whereas, if the purpose is to maintain confidentiality of the company's data, then it is best that the key is generated by the CA and not by the client, so that the encrypted information can always be restored in the event of the client losing the key. This is because CAs are usually more reliable than clients.

Identity-Based Public Key Cryptography (ID-PKC)

Certificate and key management continue to be a difficult problem under the PKI model. To solve this problem, a new model was constructed, namely identity or identifier based cryptography. This scheme was first proposed by Shamir in 1984. The general idea of this scheme was to provide a means of generating a user's key based on some user's public identifiable information, such as: name, email or address. However this scheme was only applicable to digital signature and not data encryption. A more efficient identity based encryption scheme was finally proposed by Boneh and Franklin in 2001.

The major difference between the two variants, namely PKI and ID-PKC lies upon the way a key is generated. As opposed to traditional PKI, a key pair in an ID-PKC system is explicitly derived from data which is of relevance to the key usage. For instance, a user's identity within the system can be used to derive the corresponding key pair. This has an attractive property in that it allows Alice to generate the public keys of Bob without having to go through the usual process of asking Bob to send his key or having to perform a directory search.

However, because of the underlying mechanics of the ID-PKC algorithms, the master secret is necessary in generating the private key. A trusted authority (TA) (which is equivalent to the CA in a PKI setting) is responsible for holding such a secret.

Only recently, it has been discovered that a client's public key can be derived from information other than the client's identity. Meaning to say, data used to generate the key pair can include information like: key validity period, user position in the relevant organisation, etc. This new feature confers a broader understanding to the identifier-based public key cryptography model.

Provided that the TA is the entity that has a direct responsibility over generating private keys in any given ID-PKC setting, there exists a built-in escrow facility within in the system. Depending on various factors, escrow facility may or may not be advantageous. This calls for a change in the architecture of the trusted third party, particularly on its role with respect to

the whole system. The role of the CA in PKI is different from the role of the TA in ID-PKC, such that in a PKI, the job of the CA is merely to validate authenticate the certificate and its content, while in an ID-PKC, the TA has a larger job scope since it is concerned with the generation and distribution of all keys within a system.

One important requirement is the establishment of a secure channel between the TA and the client prior to the distribution of the private key. The channel needs to be secure enough as to protect the confidentiality and authenticity of the private key being transmitted.

PROTOCOLS

TLS v1.3 Handshake Protocol

The chief purpose of the TLS protocol is to establish and maintain a secure channel between an authenticated web site and an unauthenticated web browser. Despite the fact that the protocol is currently being used in other less relevant applications, many libraries implementing TLS are easy to use and highly accessible. Essentially, the TLS protocol consists of two parts, namely the handshake protocol and the record protocol. The first is concerned with client authentication and key establishment, while the latter is concerned with protecting bulk data using the established keys. In the following paragraphs, we attempt to provide a good and up to date analysis of the Handshake protocol and its use in the Record protocol.

The arrows in Figure 2 show the general flow of the message while the boxes represent the cryptographic operations associated with the key schedule of the full handshake. In the following, we list and explain briefly the handshake messages as discussed by (Dowling et al., 2017):

- ClientHello(CH)/ServerHello(SH) comprise of necessary cipher suites and supported versions for the negotiation phase, including random nonces. A session identifier (session_id) is included in SH

for the purpose of tracking interrupted sessions. Several extension fields can also be added to CH and SH.

- ClientKeyShare(CKS)/ServerKeyShare(SKS) contain, for one group or more than one group selected by a CH/SH extension field, ephemeral Diffie-Hellman shares $X = g^x$ and $Y = g^y$ respectively.

Both client and server can at this point calculate the Diffie–Hellman shared secret (XY) as the pre-master secret (PMS) and then use a pseudo-random function PRF to deduce what is known as the handshake master secret (HMS) and the handshake traffic key (tkhs). At this stage both client and server still remain unauthenticated.

Note that the handshake traffic key (tkhs) is used to encrypt all subsequent messages:

- EncryptedExtensions (EE) contains many more extensions.
- ServerCertificate(SCRT)/ClientCertificate(CCRT) each includes the relevant public-key certificate.
- CertificateRequest (CR) is a request issued by the server asking the client to authenticate himself using a certificate.
- ServerCertificateVerify(SCV)/ClientCertificateVerify(CCV) is a record that has a digital signature of the session hash or in other words: the hash of all handshakes messages that have been transmitted in the protocol execution history.
- ClientFinished(CF)/ServerFinished(SF) contain the pseudo-random function (PRF) evaluation on the session hash keyed with the handshake master secret (HMS). Both Client and Server now are able to compute the application traffic key (tkapp), the master secret (MS) and the resumption master secret (RMS) which allows interrupted future sessions to resume operations.

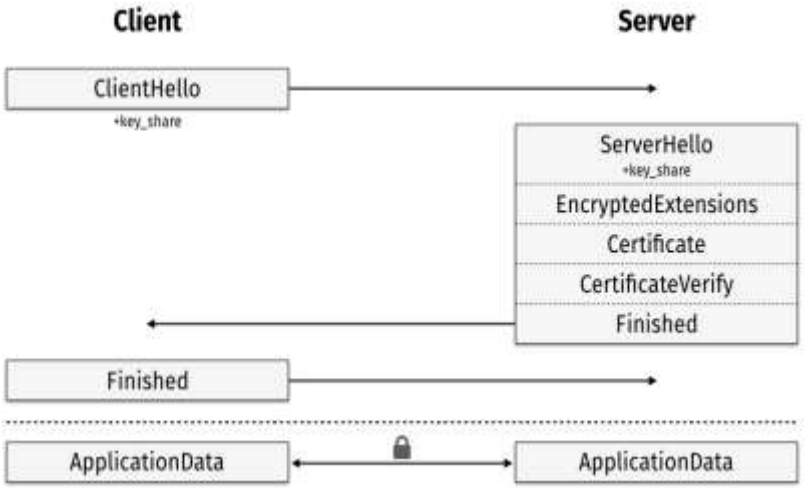


Figure 2. TLS 1.3 handshake protocol (European Union Agency for Network and Information Security, 2014).

TLS v1.3 no longer supports static RSA key exchange, which can be noted as one good improvement over the old versions. This translates to the fact that we can perform full handshakes using the Diffie-Hellman forward-secure scheme directly.

It should also be noted that the ChangeCipherSpec subprotocol has been removed. Additionally, for the purpose of protecting data from passive attacks very early in the course of communication, the ephemeral secret is used in TLS v1.3 to encrypt all messages that are sent after ServerHello. Hello extension data that is included in EncryptedExtensions must be encrypted, because the data is not required to establish the secure channel.

The removal of the ClientKeyExchange and ServerKeyExchange messages is perhaps the most important change when speaking of 1-RTT (Round-trip time). New special KeyShare extensions now contain the public keys and Diffie-Hellman parameters. This type of extension is contained in the ClientHello and ServerHello messages. Putting this data in the Hello extensions helps in keeping the TLS v1.3 handshake compatible with TLS v1.2 handshake, essentially because the message sequence is kept intact.

Password-Authenticated Key Exchange (PAKE)

Password-based authenticated key exchange (PAKE) protocol was first introduced by Bellovin and Merritt in 1992. The protocol allows two parties to exchange messages and set up a cryptographic key, based solely on a predetermined password. The two common threats against PAKE protocol are offline and online dictionary attacks. An offline dictionary attack is a situation in which an attacker attempts to discover the user's password, by trying all possible dictionary words and searching for a match based on a message exchange record. Conversely, in an online dictionary attack, the attacker tries to authenticate himself by persistently trying to guess the password using a dictionary.

After the introduction of the PAKE concept by Bellovin and Merritt (1992), several PAKE versions have been suggested. Generally speaking, there are two types of PAKE systems as noted by Yi et al. (2016), the two types are distinguished according to how the password is distributed. In the first type, the password resides in one single server, whereas the password is distributed across multiple servers in the second type. As for the single-server type, there are three main subcategories:

Password-only PAKE: A protocol in which two parties exchange packets encrypted using a pre-shared password in order to set up a shared secret key.

PKI-based PAKE: In this setting, the client not only shares a password with a server, but also keeps the public key of the corresponding server.

ID-based PAKE: ID-based PAKE is a middle ground between password-only PAKE and PKI-based PAKE, such that the client needs to know the server's identity and the password, while the server must have the private key relevant to its identity and the shared password.

As noted above, in a single-server environment, all passwords required for client authentication reside in a single server. All passwords stored in the server face the risk of getting disclosed in the event of server compromise like an unauthorized access by hackers. The multi-server

model was introduced by (Ford and Kaliski, 2000) and (Jablon, 2001) to answer the problems presented by the single-server model. Unlike single-server systems, multi-server systems allow the password to be stored across n servers. PAKE multi-server protocols can be grouped in two sub-categories (Yi et al., 2016):

Threshold PAKE: This protocol allows the client's password to be distributed among n servers. In return, these servers coordinate to authenticate the client and set up session keys. When under attack, the protocol is expected to remain secure if the number of compromised servers is less than n.

Two-server PAKE: This protocol is a special type of threshold PAKE where n is equal to two. The protocol becomes insecure when both servers become compromised.

CHALLENGES

A Comparison of PKI and ID-PKC

In this section, we identify and explain the differences between PKI and ID-PKC with regards to key management:

- The Trusted Authority (TA) in an ID-PKC system is required to record, through a database system, all IDs to which keys have been issued. Unless this requirement is fulfilled, the system remains vulnerable to key collision events. A key can go through many transformations after the end of its life-cycle. For instance, the system may revoke, remove or regenerate the key. Now, suppose a new client, having the same identity within the system as the first client, requests a key. Naturally, the two clients will have the same key pair or as we call it: key collision. Keys are assumed to be unique; any identical keys can lead to confusion and privacy issues. Some may argue that this issue is also present in the PKI setting, in the sense that a CA can certify two users sharing the

same identity. However the issue is different with ID-PKC. Due to the inherent relationship between the public key and ID in ID-PKC, same IDs results in same keys. Whereas in PKI systems, keys are generated using a client random generator and the client ID is not involved in the key making process. Inevitably, this results in different keys regardless of IDs variance.

- A similar problem may arise in case we propose the standard ID-PKC model as a solution to the above mentioned problem. Meaning to say, we include many more identity attributes in order to decrease the chances of having a duplicate identity. This may sound a plausible solution at first, however, the more details we include the more similar it becomes to the certificate system in the PKI. Having more attributes would create more complications. A client would have many issues to deal with when generating a key, such as the format of fields, the correct definition of each field, etc. This situation will require a new set of standards and specifications, thus coming to a set of issues that is somehow similar to the ones we have in PKI. The actual meaning of each identifier may not be clear to the client without a predefined agreement between the TA and other parties. Thus, problems related to certificate management still persist.
- The above two points originate from the fact that identities and public keys are linked in ID-PKC, whereas the link does not exist in PKI. We cannot determine whether this is an advantage or a drawback without considering the application we are dealing with.
- There is another problem that can affect the availability and normal operation of key generation in ID-PKC. Due to the fact that the identity is determined by the client, there is a chance that the client may submit an identity that cannot be processed by a TA or used to generate a private key. This has a negative consequence on the client as it makes him unable to decode important messages. This can still be solved by manual intervention, but it is highly inconvenient.

One common issue when using PKI is revocation. There is very little concern about managing identities which are used to generate keys in ID-PKC. This is mainly due to the fact that ID-PKC does not work by certificates. This issue is similar to the issue of certificate management in a PKI system. We explain in this section why revocation presents a real challenge in ID-PKC, and not only in PKI. The following points point out the two main challenges of revocation in ID-PKC as seen by Paterson and Price (2016):

- As discussed previously, keeping track of a database of identities or identifiers that were used to produce keys is a potential problem when it comes to key management in ID-PKC. Due to the relationship between identities and keys in ID-PKC, identity revocation must precede the relevant public key revocation. This issue can become a serious challenge when the identifier in question is considered to be hard to change, such as a phone number. But these are exactly the kind of attributes that are relatively easy to predict by a party trying to issue a key to a client in an independent manner. Knowing that, we should use less predictable identifiers in ID-PKC implementations. For instance, attaching issue numbers to identity attributes or identifiers is theoretically simple, but makes identifiers less predictable. Thus in real implementation, to generate a key, we should attach more complex identifiers to the client's identity. Unfortunately, this results in more complexity to the entity responsible of managing such identifiers.
- There is another challenge when we try to address the re-certification issue, again due to the inherent linkage between keys and identifiers in ID-PKC. Paterson and Price (2016) demonstrated in a previous PKI workshop that an organisation could modify a certificate's content, or in other words, perform recertification, without having to change the client's private key in the process. This proves to be very cost-effective. The key step of the experiment was to store a certificate and a corresponding key in

separate storage mediums. The same cannot be done in an ID-PKC implementation.

Key Management Challenges in the Cloud

The cloud is made up of shared network devices and computing resources (e.g., applications, services, servers, storage and networks) and provides easy, on demand access to those resources. The cloud can be released, managed and rapidly provisioned with very little effort required from the cloud manager (Mell and Grance, 2011). A clear and simple taxonomy of three service types available to cloud users were pinpointed by the National Institute of Standards and Technology (NIST), namely, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

Due to the complex nature of the cloud architecture, things like data ownership, resource distribution and key management system (KMS) can be complex as well. KMS is responsible for managing and executing all cryptographic operations in the cloud. One common challenge arises from the fact that in most cases the data is owned by the cloud consumer, but the same data is physically stored in multiple resources that are under the control of the cloud provider. In most cases, the KMS which is responsible for the cryptography key management tasks, needs to execute cryptographic operations in resources or hardware that does not belong to the data owner. Under these conditions, the cloud consumer may not be able to get full assurance of whether his privacy is protected.

One important security challenge that exists in the cloud is the issue of authenticating virtual machine (VM) templates. The bootstrapping problem arises when cryptographic methods (i.e., cryptographic hash function, digital signature or message authentication code) are used to authenticate VM templates. To mitigate this problem, a full security analysis should be performed and we should not focus only on investigating the key management problem.

It can be observed that the IaaS cloud consumer controls the Database Encryption Key (DEK). This is because it has control over the registered Database Management System (DBMS) instance. Because encryption is carried out at the input/output (I/O) level, the DEK and the database data need to reside closely in terms of storage location. This condition leaves the cloud consumer with one option, which is to store the DEK in the same location as the DBMS instance. Despite that, there exists other implementations of Transparent Data Encryption (TDE) (known as encrypting data at rest (Microsoft, 2017)) which provide table-level and column granularity for encryption. It is commonly used for storage encryption, and therefore, this implementation cannot be customized to offer different keys to different clients according to permission level.

The SaaS cloud provider retains the right to control the cryptographic keys of its clients, hence there is normally no assurance given from the cloud provider to its clients against insider attacks. Consequently, there can be cases where data owned by different users share a single storage unit and is encrypted using a single key. In such a situation, there are clearly no boundaries among different data belonging to different owners. Moreover, the sheer data volumes stored and distributed in several SaaS resources would necessitate huge numbers of symmetric keys. This will require several key management servers to manage those keys. Besides that, a lot of Hardware Security Module (HSM) partitions are required to be installed and maintained in the case where key management operations are executed using HSM.

Challenges in TLS

TLS 1.3 was designed to bring new improvements to the previous TLS version. One of which is to improve the handshake protocol by trying to realize the zero round-trip time (0-RTT). This new feature has the potential to save latency by enabling the client to send application data to the server during the first message flow. This appealing feature has its inherent

drawbacks (Dowling et al., 2017). In particular, it makes the system vulnerable to replay attacks and lacks forward secrecy.

Challenges of PAKE

There are a lot of research efforts invested in designing and modeling a secure PAKE protocol. As noted by Nam et al., (2014), when discussing PAKE protocols, the most crucial challenge that we run into is dictionary attacks and how to prevent them. A dictionary attack is an attack in which an adversary attempts to find the correct password by trying out all dictionary words. In the 3-party setting, there is a difficult challenge in designing PAKE protocols that are immune against dictionary attacks. That is due to the fact that the two clients aiming to set up a session key share their password with an authentication server but not with any other client. An inside attacker in 3-party PAKE settings might be disguised as a registered client and thus have access to the server. The attacker then is able to establish protocol sessions and communication channels with other legitimate users.

PROPOSED SOLUTIONS

Cloud Security

It is strongly advised (Chandramouli, Iorga and Chokhani, 2013) that the cloud user should employ Secure Sockets Layer (SSL) or Secure Shell (SSH) to set up a secure session with the engine that is responsible of virtual machine (VM) template integrity verification.

The configuration of the application entity that supports the VM integrity verification engine operations must be adjusted to allow the application to function as a secure appliance on a specially enhanced VM. Based on the authentication technique selected by the cloud provider to authenticate the VM template (Chandramouli, Iorga and Chokhani, 2013),

the corresponding verification engine needs to have the necessary hash values, secret keys and public keys. Then, there will be no need for an out-of-band, secure channel between cloud provider and consumer. This proposed improvement has another advantage in that it enables the cloud provider to make changes to the authentication method, VM template, algorithms and keys without requiring a secure, out of band channel with the cloud user. In this case, a cloud manager can employ different cryptographic methods (digital signatures, MAC, or cryptographic hash) to secure different VM templates.

To ensure adequate security, the cloud user, before launching the VM, must verify and check the VM template and also has to run the authentication procedures provided by the cloud manager.

Certificateless Public Key Cryptography

In this part, we discuss a new public key cryptography paradigm (Al-Riyami and Paterson, 2003.), known as certificateless public key cryptography, in short CL-PKC. Designers of this concept were motivated by the need for a new scheme that doesn't include the key escrow feature, which is in ID-PKC. Besides that, they wanted a certificateless model, which is a scheme that does not require certificates unlike PKI. In other words, it is an intermediate form that enjoys both properties of PKI and ID-PKC. In the following paragraphs we define CL-PKC features and characteristics:

Although a CL-PKC system is certificate free, it is still in need of a trusted third party (TTP), namely the key generating centre (KGC). Unlike the private key generator (PKG) of ID-PKC, a KGC has no access to private keys of its registered entities. Essentially, the KGC uses a master key and an identifier of entity A (IDA) to generate a partial private key (DA) and then submits it to the designated entity. For the purpose of this subject we assume that A and its corresponding identifier IDA are equivalent and exchangeable. We can think of identifiers as random strings. The security of this system depends largely on ensuring an

authentic and confidential key delivery to the corresponding legitimate entity.

In order for entity A to generate its full private key (SA), A must incorporate two elements, namely the partial private key (DA) and some other secret information. This method can provide the assurance that the KGC does not know about A's actual private key. For A to compute its public key (PA), it needs to combine some of the KGC's public parameters with its secret information. It can be observed that the entity's secret information is the only element required to generate both SA and PA. More importantly, entity A can issue a PA without the prior knowledge of an SA. Since the client's identity is not the only input parameter in generating the public key, we can safely say that this is not an identity based model.

There are mainly two methods for an entity A to share its public key. One way is to send the public key along with a message, which can be implemented in a login application. Another way, which is more suitable for encryption applications, is to place the public key in a public directory. The latter method provides no security assurance in protecting the public key. Particularly, there is no certificate that protects and binds A's public key. Now, if entity B wants to send an encrypted message to A or check a digital signature coming from A, then B needs to use IDA and PA.

PAKE Safeguards against Insider Dictionary Attacks

We present the first 3-party PAKE protocol, whose indistinguishability-based security as well as password security against all classes of dictionary attacks are formally proved in a well-defined communication model by Nam et al. (2014). We recommend using a simple and intuitive approach of capturing dictionary attacks. The indistinguishability-based security property in the Bellare-Pointcheval-Rogaway model implies security against (both insider and outsider) offline dictionary attacks. It is proven that a protocol cannot achieve the

indistinguishability-based security if it is not secure against an offline dictionary attack.

Insider and outsider attacks can be prevented if one of the two protocols, 2PAKE (2-party PAKE protocol) or 3KD (3-party Key Distribution protocol), is instantiated with a protocol that provides client-to-server authentication (Nam et al., 2014). It is observed that a typical 3-party key distribution protocol is not expected to provide client-to-server authentication, and hence, we suggest that the countermeasure targets the instantiation of 2PAKE. While some might also suggest that a round-optimal protocol (i.e., a protocol that runs in a single round) should be used in the instantiation of 2PAKE to achieve better efficiency, we caution against this as no round-optimal 2-party PAKE protocol is known to provide client-to-server authentication and achieve security against offline dictionary attacks.

CONCLUSION

In summary, we explained the fundamental algorithms underlying most of the current key management protocols, namely Diffie-Hellman and RSA key exchange. We also managed to give a fair description of two of the common protocols implemented in the internet today: TLS and PAKE. Although TLS v1.3 offers new improvements over the previous versions, it still remains vulnerable due to the lack of forward secrecy and the possibility of replay attacks, mainly due to its design specifications. We discussed PAKE and its different sub-classes. Although PAKE is a subject of dictionary attacks, it remains an attractive protocol when it comes to web based applications. Ample attention was given to key management in cloud services. Cloud technology introduces some serious challenges to information security and privacy in general and to cryptography key management in particular. We also attempted to propose some solutions to the aforementioned problems. We offered the certificateless public key cryptography model to answer the challenges posed by Public Key Infrastructure and Identity-Based Public Key Cryptography.

REFERENCES

Al-Riyami, S. S. and Paterson, K. G. 2003. "Certificateless public key cryptography." *Advances in Cryptology - ASIACRYPT 2003*, pp. 452–473.

Bellovin, S. M. and Merritt, M. 1992. "Encrypted key exchange: Password based protocol secure against dictionary attack*."* In *Proc. 1992 IEEE Symposium on Research in Security and Privacy*, 72-84.

Boneh, D. and Franklin, M. 2001. "Identity-based encryption from the Weil pairing." *Advances in Cryptology – CRYPTO 2001*, volume 2139 of LNCS. 213-229.

Chandramouli, R., Iorga, M. and Chokhani, S. 2013. "Cryptographic Key Management Issues & Challenges in Cloud Services." *National Institute of Standards and Technology Interagency or Internal Report 7956*.

Diffie W, Hellman M. 1976. *New Directions in Cryptography*. Stanford University; 40 p.

Dowling, B., Fischlin, M., Gunther, F. and Stebila, D. 2017. "A cryptographic analysis of the TLS 1.3 handshake protocol candidates." *ACM Conference on Computer and Communications Security (CCS)*.

European Union Agency for Network and Information Security. 2014. *Study on cryptographic protocols*.

Ford, W. and Kaliski, B. S. 2000. "Server-assisted generation of a strong secret from a password." In *Proc. 5th IEEE Intl. Workshop on Enterprise Security*.

Jablon, D. 2001. "Password authentication using multiple servers." *In Proc. CT-RSA'01*, 344-360.

Mell, P. and Grance, T. 2011. *The NIST Definition of Cloud Computing.* NIST Special Publication SP 800-145.

Microsoft. 2017. Transparent Data Encryption (TDE). *Microsoft documentation*. Retrieved from: https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption-tde.

Nam, J. Choo, K. K. R., Kim, M., Paik, J. and Won, D. 2014. "Password-Only Authenticated Three-Party Key Exchange Proven Secure against Insider Dictionary Attacks." *The Scientific World Journal Volume 2014*, Article ID 802359, 15. Retrieved from: http://dx.doi.org/10.1155/2014/802359.

Paterson, K. G. and Price, G. 2003. "A comparison between traditional public key infrastructures and identity-based cryptography." *Information Security Technical Report*, 8:57–72.

Rivest, R., Shamir, A. and Adleman, L. 1978. "A Method for Obtaining Digital Signatures and Public Key Cryptosystems." *Commun. ACM*, 21(2):120–126.

Shamir, A. 1984. "Identity-based cryptosystems and signature schemes." *In Advances in Cryptology – CRYPTO '84*, volume 196 of LNCS. 47-53.

Yi, X., Rao, F.-Y., Tari, Z., Hao, F., Bertino, E., Khalil, I. and Zomaya, A. Y. 2016. "*ID2S Password-Authenticated Key Exchange Protocols.". IEEE Transactions on Computers*, Volume: 65, Issue: 12.

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 4

TARGETED IMAGE FORENSICS

Rimba Whidiana Ciptasari*
School of Computing, Telkom University, Bandung, Indonesia

ABSTRACT

Constructing a forensic method with various types of manipulation still remains a challenging task. The most common form of manipulation is combining part of the image fragment into a different image either to remove or blend the object. This chapter will discuss how the forensic method could be constructed with two types of mechanisms. The proposed system could detect the traces of tampering derived from either the image under investigation or other similar images, so-called single authentication and reference images utilization, respectively. The discussion begins with specific problem formulations, particularly in image splining, eye specular highlight, and image splicing. Then, several technical methods used in developing the forensic system are discussed. The performance of the methods are demonstrated in real-world examples.

* Corresponding Author, Email: rimbawh@telkomuniversity.ac.id

Keywords: eye-specular highlight, weighted-average splining, reference images utilization, single authentication, pixel intensities reconstruction, least-squares estimation method

INTRODUCTION

The advent of the digital era has spread the use of digital multimedia into virtually all areas. A multimedia signal can be easily reproduced and manipulated without any trace of alterations. Constructing a scientific and automatic way for confirming the legitimacy or truthfulness of the structure and/or content of multimedia is an important task, which is the aim of this research. There are two major approaches to multimedia authentication: *active and passive approaches*. The former may work in two areas: digital watermarking and signature authentication. In the case of digital watermarking, it works by inserting a digital watermark into the host data at the source side and verifying the watermark integrity at the detection side. In contrast, the latter accomplishes data authentication without any prior information. That is, data integrity is verified without the presence of either a digital watermark or signature.

The main purpose of digital image forensics, based on the passive approach, is the proper identification and collection of computer evidence. However, all of the image forensics' schemes developed so far have not provided verifiable information on the *source of tampering.* In this chapter the focus will be on discussing so-called *targeted image forensics*. Two methods will be introduced to verify the integrity of a questioned image, namely single authentication and reference image utilization. In the case of single authentication, the algorithms are constructed from knowledge of the image-processing functions and are thus targeted to a specific image manipulation operation. While in reference image utilization, the reference images, assumed to be the authentic ones, are targeted to the other images constructing the questioned image.

To begin with, the problem of specific image manipulations is formalized and the particular methods which are useful in deriving the

tampering artifacts are described. Apart from the outstanding image forensic research reported in (Kee and Farid, 2010; O'Brien and Farid, 2012; Johnson and Farid, 2005; O'Brien and Farid, 2012; Popescu and Farid, 2005), this chapter will describe the framework of parameter estimation to detect traces of the digital tampering. The methods are evaluated for both synthesized dataset and realistic forgery images which are reported on section *System Performance and Evaluation*, and concluded with a summary.

FORGERY PROBLEM FORMALIZATION

In terms of image manipulation, the digital image could have undergone various processing operations. In the case of a forensic's tool, it is impossible to construct such a system that precisely identifies all types of manipulation used. Therefore, relatively different types of image manipulation operations will be described. Afterwards, a model of a single authentication that will be useful for deriving parameter estimators and for constructing detectors will be determined.

Weighted-Average Splining Detection

A set of color images $\mathcal{I}$ whose sizes are $M \times N \times 3$ matrices, and their entries are integer values $\in [0,255]$ will be considered, given two vectors α_l and α_r denoted as left and right weighting function, whose entries are in interval [1,0] and [0,1], respectively.

Suppose there exists two individual different color images, $\{I_l, I_r\} \in \mathcal{I}$, to be splined at position $\mathrm{P} = \{p_1, p_2, \dots, p_n\}$, where p_i is referred to as transition zone T consisting of a pixel coordinate (x, y). Given a set of weight coefficients α, by applying splining function $S: \{I_l, I_r\} \times \alpha \rightarrow I_S$, the splined image I_S is constructed.

Given a questioned image $\tilde{I}$, the image prediction function is defined as $fE: \tilde{I} \times \Phi \rightarrow \{\hat{I}_l, \hat{I}_r\}$, where Φ is the set of predictive coding parameters

and $\{\hat{I}_l, \hat{I}_r\}$ is subjected as a pair of left and right images constructing image $\tilde{I}$. The function of weight coefficient estimation, i.e., the least-square estimation method, is formulated as $LS: \{\hat{I}_l, \hat{I}_r\} \times \alpha \rightarrow \vec{\alpha}$, where $\vec{\alpha}$ is the predicted coefficients. Finally, an image is formulated as $\tilde{I} \approx I_S$ iff $\vec{\alpha} \approx \alpha$.

Eye-Specular Detection

Suppose there exists a set of color images $\mathcal{I}$, where $I \in \mathcal{I}$ whose sizes are $M \times N \times 3$ matrices, and their entries are integer values $\in [0,255]$. Each I contains n human objects $I_0 = \{O_1, O_2, O_3\}$ where each human object has a pair of eyes $E = (E_r, E_l)$ with the reflection of light on the eye $p \rightarrow E$.

Based on observations, the image is considered to be the authentic one if it consists of at least two people having particular eye characteristics. The characteristics of the light source is captured in the form of the amount of light, shape and pattern $C = \{C_p, C_b, C_f\}$. The authenticity of the image is determined by comparing these characteristics in terms of shape of light on the eye area between objects.

Given an arbitrary digital image *I*, it is considered to be authentic if all three characteristics are satisfied, i.e., $C_{pO1} = C_{pO2} \cap C_{bO1} = C_{bO2} \cap C_{fO1} = C_{fO2}$, otherwise it is deemed as unauthentic.

Image Splicing Detection

Suppose there exists two set of grayscale images, $\mathcal{I}_A$ which refers to nontampered images and $\mathcal{I}_S$ which represents spliced images. Consider that the image $I \in (\mathcal{I}_A \cup \mathcal{I}_S)$ is an $M \times N$ matrix, whose entries are integer values $\in [0,255]$. The concept of pixel-based alignment is adopted to verify the validity of suspicious region(s). Given a template image *t*(*x*) sampled at discrete pixel locations $\{\mathrm{x}_i = (x_i, y_i)\}$, the location of reference images $\Re(\mathrm{x})$ is determined. It is considered that an image retrieval system

(IRS) Φ_f consisting of a set of properties φ_f characterizes the IRS. The IRS is described as $\Phi_f(\cdot): I \times \varphi_f \rightarrow \mathcal{I}_A$. The IRS is a system to retrieve images similar to a user-defined specification or pattern (e.g., shape sketch, an image example). The domain, the set of $\mathcal{D}$, is the set of input images on which the IRS can work; the codomain $\mathcal{C}$ is the set of output images defined as: $\mathcal{C}: \{\Re \in \mathcal{I}_A: \exists I * \in (\mathcal{I}_A \cup \mathcal{I}_S), \exists p * \in \varphi_f: \Re = \Phi_f(I *, p *)\}$. Note that testing of the validity of images $\Re$ used to verify the image authenticity is required. However performing such a test is fundamentally complicated. For the sake of simplicity, it is assumed that the resulting IRS in authentic images $\Re$ refers to those images which have not undergone some image processing functions.

Definition 1. (Suspected spliced-image). Suppose there are *n* suspicious regions $T = \{t_1, t_2, \ldots, t_n\}$ derived from a given image *I*, and *m* reference images $\Re = \{r_1, r_2, \ldots, r_m\}$. Assume that there exists at least one t_i such that $t_i \subseteq r_j$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. Then, image *I* is said to be a suspected spliced-image.

ENHANCED EDGE DETECTION

It is basically difficult to select the appropriate edge which falls into a spliced artifact among the detected edges. By applying a superior detector, highly detailed edges can be produced. To precisely attain the spliced artifacts, Robert cross operator, which utilizes a 2-D mask with a diagonal preference, is chosen for conducting edges detection on $I(x)$.

The drawback of this approach, however, is that images will introduce various directions in intensity which are likely undetectable. To overcome this limitation, the mask value of the initial operator is slightly modified by using a weight of 2 and -2 instead of 1 and -1 for implementing the diagonal differences.

$$g_{dh} = \frac{\delta_f}{\delta_x} = 2f(x+1, y+1) - 2f(x, y). \quad (1)$$

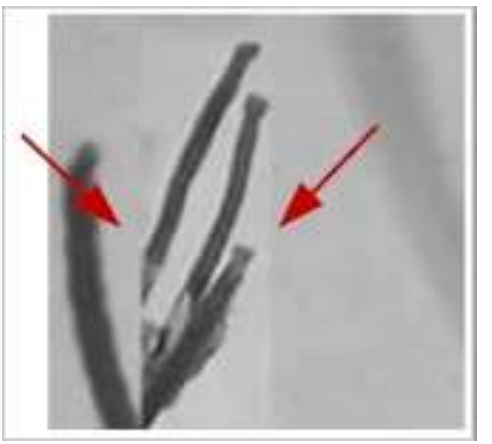
$$g_{dv} = \frac{\delta_f}{\delta_y} = 2f(x+1, y) - 2f(x, y+1). \quad (2)$$

In order to have the strongest responses, the 2×2 mask with other 2-D masks for detecting edges along both vertical and horizontal directions are combined:

$$g_h = \frac{\delta_f}{\delta_x} = 2f(x, y) - 2f(x+1, y).` \quad (3)$$

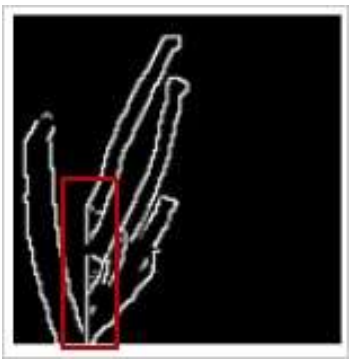
$$g_v = \frac{\delta_f}{\delta_y} = 2f(x, y) - 2f(x, y+1). \quad (4)$$

These masks are basically referred to as *pixel differences*. It can be seen in Figure 1(c) that using a 2 provides edge sharpening compared to that of Figure 1(b). The higher the weight factor used, the more accurate the edges obtained. Combining 2×2 masks for all directions produces relatively strong responses to genuine edges as depicted in Figure 1(d) which confirms that a modified mask improves the detection result while preserving the appropriate edges.

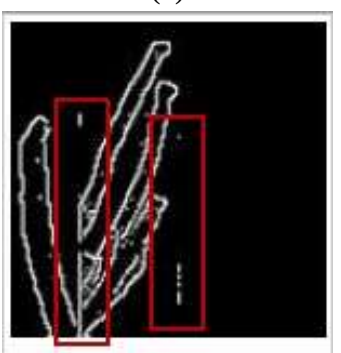


(a)

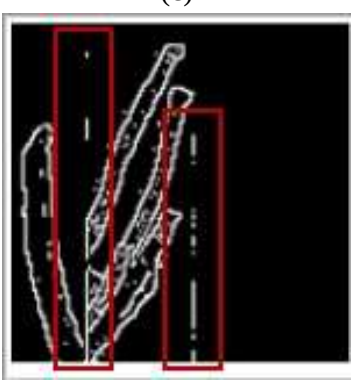
Figure 1. (Continued)



(b)



(c)



(d)

Figure 1. The sample results of exploiting modified Robert's detector (a) The sample of a forgery image of size 128×128. The edges of interest are shown by red arrows. (b) The edge pixels resulting from original Robert's detector. (c) The edge pixels resulting from a modified detector on diagonal differences. (d) The edge pixels resulting from a modified detector on all directions (diagonal, *x*-direction, and *y*-direction) (Rimba Whidiana Ciptasari, Rhee, and Sakurai 2013).

INTENSITY RECONSTRUCTION

In certain cases of forgery detection, it is required to examine the correlation of the pixel neighborhood. The correlation value could represent whether the specific region in a questioned image is originally derived from the image itself or another part of a different image. The detector is then formulated using these correlation values.

One way to analyze the pixel neighborhood correlations is to estimate their intensity values by exploiting prediction algorithms. In this section, there is a description of how the provided methods could be implemented to reconstruct the image.

Image Interpolation

Suppose that there exists an image containing a spliced region derived from another part of a different image. Upon observation, the spliced portion exhibits a higher value of prediction error of the standard deviation than that of non-spliced edges. To confirm the hypotheses, the pixel intensities at either side of the edge of interest (i.e., left/right or top/bottom) are interpolated. Once the estimated values are calculated, the prediction error is then computed by subtracting the predicted intensity values from its original values.

In order to simplify further discussion, only the edge segments that are detected in horizontal and vertical direction as illustrated in Figure 2 are considered. Suppose that $P = (x, y)$ is the intensity being observed. The intensity reconstruction of *P* is then formulated as:

$$f(R_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{21}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}). \quad (5)$$

$$f(R_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{12}). \quad (6)$$

$$f(P) = \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2). \quad (7)$$

The prediction error $\hat{e}$ is then computed as the difference between the predicted $f(P)$ and actual value of the pixel $f(x)$. To measure the average of the prediction error $\hat{e}_p$, mean square error (MSE) on E is computed.

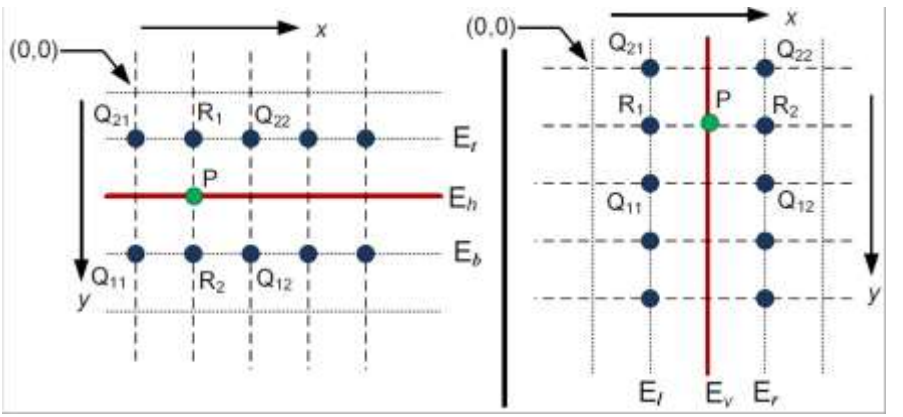


Figure 2. Pixel intensities reconstruction. The left column shows an edge detected E and its adjacent pixels in horizontal direction denoted as E_t and E_b, and the right column is edge E introduced in vertical direction and its adjacent pixels denoted as E_l and E_r (Rimba Whidiana Ciptasari, Rhee, and Sakurai 2013).

Image Prediction

Consider a composite image constructed by a weighted-average splining composed of two individual images. The specific curvature of weight coefficients could be obtained by constructing the image prediction of those two individual images, $I = (I_l, I_r)$. The problem of predicting these images can be approached using predictive coding. Since there are two images to be predicted, the process is conducted in two stages: *forward* and *backward* prediction. The former stage represents the left image I_l prediction, while the latter draws the reconstruction of the right image I_r. In general, the prediction is formulated as follows.

$$\hat{I}(x,y) = round[\beta . I(x,y-1)]. \quad (8)$$

where $\hat{I}(x,y)$ and $I(x,y)$ are subjected to predicted and splined images, respectively.

The multiplier β is formulated as follows:

$$\beta_F = \frac{I(x,y)}{I(x,y-1)}, \quad (9)$$

$$\beta_B = \frac{I(x,y)}{I(x,y+1)}, \quad (10)$$

$$\beta_T = \frac{\hat{I}(x,y)}{I(x,y-1)}, \quad (11)$$

where β_F, β_B, and β_T refer to backward and transition zone multipliers, respectively.

CASES IN TARGETED IMAGE FORENSIC

A Photo Composite Detection

Outstanding camera-based forensics work using specular highlight has been proposed by Johnson and Farid (Johnson and Farid, 2007). The image integrity is verified by exposing the direction of the light source from a 3D point of view of the light reflection. In this chapter, the characteristics of eye specular highlight are exposed using a pixel-based approach (Nugraha, Ciptasari, and Pudjoatmojo, 2015).

As described in the previous section on problem formalization, the characteristics of the light source are revealed in the form of the amount of light, shape and pattern. The eye region of interest is manually captured; the segmentation is then conducted to separate areas of light reflection and non-light reflection. Afterwards, the specular highlight is analyzed by calculating the euler number, outer line, and its formation denoted as $C =$

$\{C_p, C_b, C_f\}$. The illustration of these three processes is depicted in Figures 3, 4, and 5, respectively. To confirm the integrity of a questioned image, these parameters are evaluated using a concept described in the problem formalization.

(a) (b)

Figure 3. The samples of Euler number resulting from the segmentation process: (a) The Euler number equals 2, (b) The Euler number equals 3 (Nugraha, Ciptasari, and Pudjoatmojo 2015).

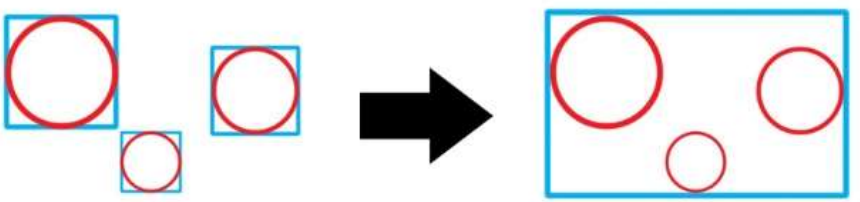


Figure 4. The example of capturing the outer line (Nugraha, Ciptasari, and Pudjoatmojo 2015).

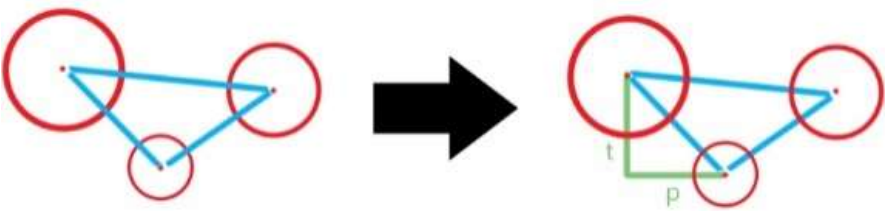


Figure 5. The example of capturing the eye-specular formation (Nugraha, Ciptasari, and Pudjoatmojo 2015).

Exposing the Splining Artifact

Consider that an image composite could be constructed by the splining operation (Burt and Adelson, 1983). Unlike the system developed for splicing verification, the resulting image exhibits smooth transition. The problem of revealing the specific form of a splined image is approached by estimating the set of weights, $\vec{\alpha}$, that satisfy:

$$Y = I_{\mathrm{L}}\vec{\alpha} + I_{\mathrm{R}}(1 - \vec{\alpha}) \tag{12}$$

Due to imprecise matching, Eq. 12 may not yield accurate results. To attain optimal weight coefficients, the estimate for the parameter $\vec{\alpha}$ is computed by minimizing the following least squares objective function (Rimba W. Ciptasari, 2016).

$$LS(\vec{\alpha}) = \sum_{i=1}^{N}\left(Y_i - (\alpha L_i + (1-\alpha)R_i)\right)^2 \tag{13}$$

The objective function provided in Eq. 13 is minimized by taking the first derivative with respect to $\vec{\alpha}$, setting the result equal to zero, and solving for $\vec{\alpha}$, yielding:

$$\vec{\alpha} = \sum_{i=1}^{N}\left\{\frac{Y_i - R_i}{L_i - R_i}\right\} \tag{14}$$

where L and R refer to I_L and I_R, respectively. To classify whether the image under investigation has undergone splining process, the prediction error between the paired samples is computed as

$$\varepsilon = \sqrt{\frac{\sum(\alpha - \vec{\alpha})^2}{N}} \tag{15}$$

The specific threshold is evaluated empirically over original images in the database such that they yield a false positive rate of less than 4%.

Image Splicing Verification

Another important consideration is that in order to verify the integrity of an image under investigation, it requires multiple image references which are assumed to be authentic ones. Another assumption for the image is an absence of post-processing operations. In practice, of course, the spliced artifacts introduced are hardly noticeable. The case may fall into an uncertainty problem. It can be considered that exploiting membership function is most adequate in representing the uncertainty in measurement. Another rationale is that both spliced and non-spliced artifacts exhibit particular characteristics.

Image interpolation, as described in section intensity reconstruction, is exploited to reconstruct pixel intensities around the edges of interest. In order to identify whether the artifacts are considered spliced or authentic, acceptable and excessive prediction error $\hat{e}_p$, prediction error-ratio $\hat{e}_r$, and weighted edge pixels bw are investigated for verification. Upon observations of either the prediction error or prediction error-ratio of spliced and non-spliced artifacts, the rules and degrees associated with these three parameters are then devised (Rimba Whidiana Ciptasari, Rhee, and Sakurai, 2013). Gaussian and Sigmoid are considered the proper membership functions describing the cases. The following are the formulation of the three rules:

Degree of rule 1 is defined as:

$$\mu_1^E = \begin{cases} 1 & , \hat{e}_p > \gamma_{mse} \\ \dfrac{1}{1+\exp\left(-\dfrac{(\hat{e}_p-\gamma_{mse})^2}{(\hat{e}_p-\alpha_{mse})\sigma_{mse}^2}\right)} & , \text{otherwise} \end{cases} \tag{16}$$

where γ_{mse} is the maximum acceptable value of $\hat{e}_p$, α_{mse} is the minimum acceptable value of $\hat{e}_p$, and σ_{mse}^2 refers to the standard deviation. The following is the degree of non-spliced artifact formulated by using the Gaussian function:

$$\mu_1^{av} = \exp\left(-\frac{\left(\hat{e}_p - \bar{E}\right)^2}{\hat{e}_p \cdot \sigma_{mse}^2}\right) \tag{17}$$

where $\bar{E}$ is an acceptable $\hat{e}_p$ of average value.

Degree of rule 2 is devised as:

$$\mu_2^E = \begin{cases} 1 & , \hat{e}_r \geq \gamma_r \\ \frac{1}{1+\exp\left(-\frac{(\hat{e}_r - \gamma_r)^2}{(\hat{e}_r - \alpha_r)\sigma_r^2}\right)}, & \text{otherwise} \end{cases} \tag{18}$$

where γ_r is the maximum acceptable value of $\hat{e}_r$, α_r is the minimum acceptable value of $\hat{e}_r$, and σ_r^2 refers to the standard deviation. And the degree of non-spliced artifact is defined as:

$$\mu_2^{av} = \exp\left(-\frac{(\hat{e}_r - \bar{R})^2}{\hat{e}_r \cdot \sigma_{ratio}^2}\right) \tag{19}$$

where $\bar{R}$ is an acceptable $\hat{e}_r$.

The degree of rule 3 is expressed as:

$$\mu_3^E = \begin{cases} 1 & , w_E \geq \gamma_{oc} \\ \frac{1}{1+\exp\left(-\frac{(w_E - \gamma_{oc})^2}{(w_E - \alpha_{oc})\sigma_{oc}^2}\right)}, & \text{otherwise} \end{cases} \tag{20}$$

where w_E is the number of edge pixels of EOI, γ_{oc} is the maximum acceptable value of $\hat{w}$, α_{oc} is the minimum acceptable value of $\hat{w}$, and σ_{oc}^2 refers to the standard deviation. The degree of non-spliced artifact is devised as follows:

$$\mu_3^{av} = \exp\left(-\frac{(oc - \overline{OC})^2}{oc \cdot \sigma_{oc}^2}\right) \tag{21}$$

where $\overline{OC}$ is an acceptable $\hat{w}$. Then, these rules are combined to calculate the degree of spliced and non-spliced:

$$\begin{cases} D_s = \left\{ max\{\mu_1^E, \mu_2^E, \mu_3^E\} \right\} \\ D_A = max\{min(1-\mu_1^E, 1-\mu_2^E, 1-\mu_3^E), min(\mu_1^{av}, \mu_2^{av}, \mu_3^{av})\} \end{cases} \quad (22)$$

where D_s is the degree of spliced, and D_A is the degree of non-spliced artifacts. If $D_s > D_A$, then the region is identified as a spliced one; otherwise it is considered as a non-spliced artifact.

CASES IN TARGETED IMAGE FORENSIC

In this section, the practical implementation of the proposed forensic methods will be demonstrated. To confirm how the methods could work on real imagery, some sample experimental results are reported as well. The reader is referred to (Rimba Whidiana Ciptasari, Rhee, and Sakurai, 2013; Rimba W. Ciptasari, 2016; Nugraha, Ciptasari, and Pudjoatmojo, 2015) for a more detailed experimental verification of a photo composite, image splicing, and image splining, respectively.

Single Authentication on Exposing Splining Artifacts

In the case of integrity verification based on the artifacts extracted from the image under investigation, *single authentication* is utilized. The experiment was carried out on synthesized splined images generated from 25 uncompressed BMP true color images (256×256 pixels) with a transition zone ranging from 16 to 64 pixel width. Figure 6 describes the system performance in estimating the set of splining weights that achieves high true positive rates at roughly 97.39%.

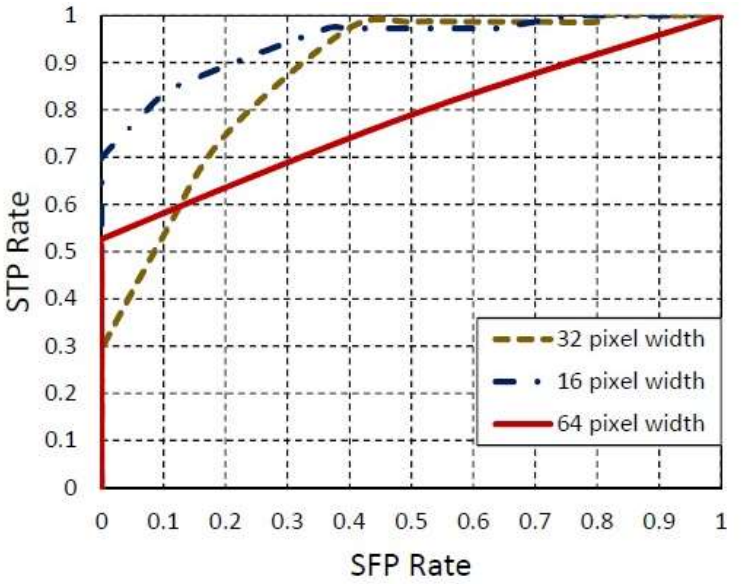


Figure 6. ROC curve of various width of transition zones.

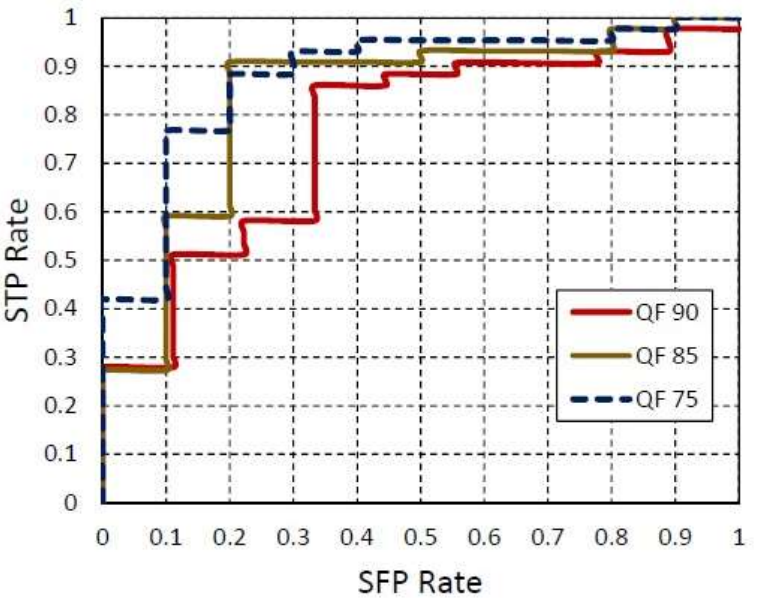


Figure 7. ROC curve of various quality factors of JPEG.

The system is tested on images that have undergone JPEG compression as well. Upon observation, the experiment was conducted on a splined image with a broad transition (T = 64 pixel width) leading to smooth artifacts. The system performance is reported in Figure 7 confirming that the overall detection performance is relatively robust to various JPEG quality factors (QF). The detection system achieves relatively high true positive rates (on average 86.67%) with 0.19 false positive rates. The reader is referred to an original publication for detailed report (Rimba W. Ciptasari, 2016).

Single Authentication on Eye-Specular Detection

Another objective of single authentication is to detect the presence of eye-specular highlight in a given image composite. To evaluate the system, the synthesized image composite is generated in several types of distortion and environments. The experiment is carried out on 100 images consisting of 20 authentic and 80 composite images in which the sample images are depicted in Figure 8. In the case of composite detection, the ROC curve confirmed that the system achieved good performance having an AUC (Area Under Curve) of 0.9304. In all cases, such as the number of persons in a photograph and the environment (indoor/outdoor), the system had good accuracy greater than 85%. The reader is referred to (Nugraha, Ciptasari and Pudjoatmojo, 2015) for a detailed report.

Reference Images Utilization

In the case of providing proof of tampering in the court of law, other authentic samples related to the investigated material are occasionally required. Considering this situation, the forgery detection system was tested on both synthesized and beyond the dataset involving multiple reference images which were assumed to be the authentic ones. The

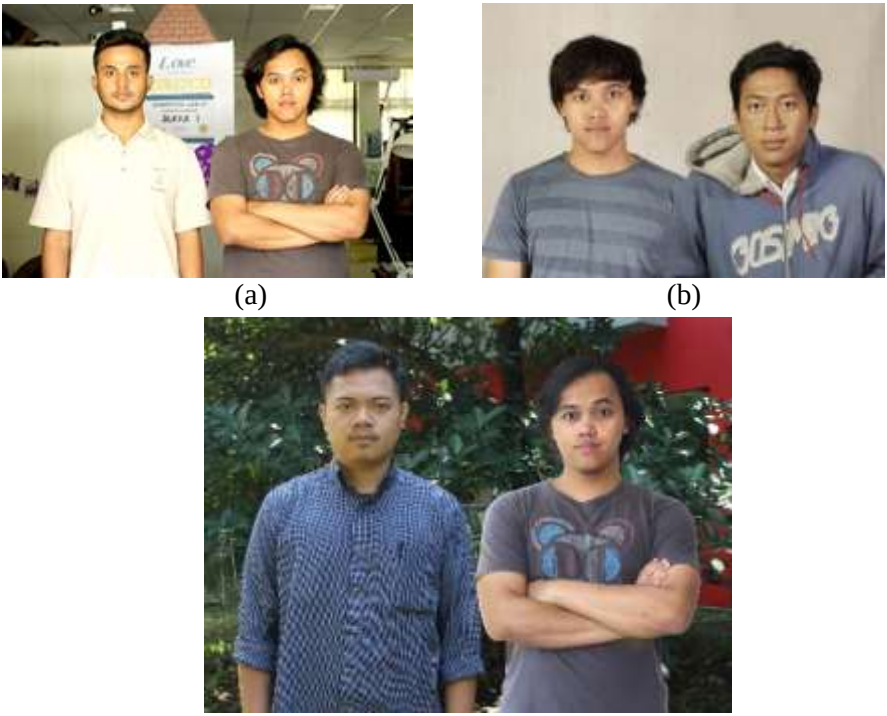


Figure 8. The samples of images in the dataset used in the experiment: (a) The original image. (b) The composite image taken in an indoor environment. (c) The composite image taken in an outdoor environment.

detailed synthesized image constructions are classified into three categories - smooth, texture, and smooth-texture - as reported in (Rimba Whidiana Ciptasari, Rhee, and Sakurai, 2013).

Prior to splicing detection, the configuration of particular parameters required in membership functions is carried out in such a way that the spliced artifacts exhibit has an outlier value over the others. To confirm the efficacy of the splicing detection, the experimental results are depicted in Figure 9. It shows that according to Figure 9(b) there are two reference images similar to the image under investigation 9(a). Figure 9(d)-9(h) shows five pairs of suspicious regions extracted from the target image. By conducting an alignment algorithm and calculating their correlations, it is concluded that the questioned image is deemed as a spliced one. The

detailed explanation and more examples of splicing detection on realistic forgery images can be found in the original publications of (Rimba Whidiana Ciptasari, Rhee and Sakurai, 2013).

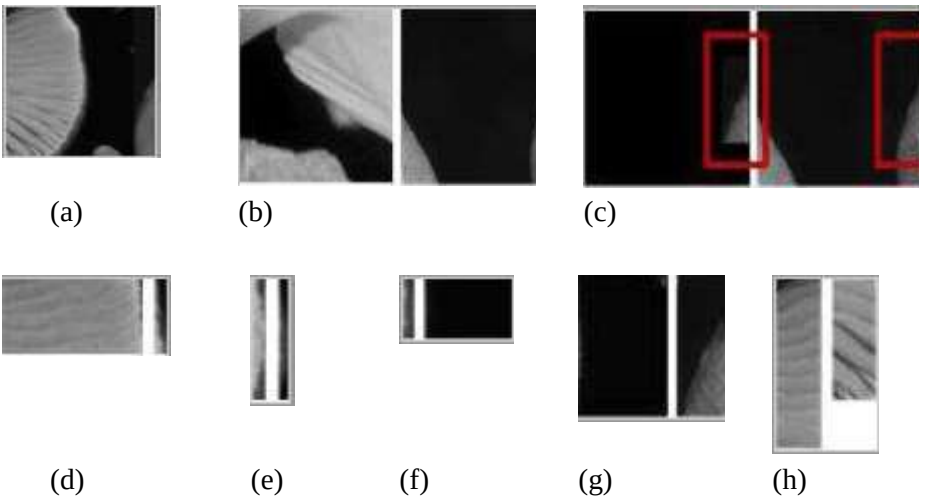


Figure 9. Verification mechanisms. (a) target, (b) reference images, (c) verification results, (d)-(h) extracted suspicious regions (Rimba Whidiana Ciptasari, Rhee, and Sakurai 2013).

SUMMARY

Two verification schemes on forgery detection, namely single authentication and reference images utilization have been introduced in this paper. Despite the fact that there have been several single authentication schemes published by other researchers, consideration in this paper was given to the idea that a forgery might be generated from a splining operation. The least-square estimator is exploited to estimate the set of weights that might emerge in a questioned image.

In addition, exploiting reference images which are similar to the image under investigation were discussed. It is proposed that this condition might occur in real situations, especially for providing tampering proof in the court of law.

Looking forward, both aforementioned schemes will still remain as ongoing and challenging research fields. The performance of all forensic methods discussed here were evaluated on real images. Throughout the text, references to previously published articles will lead any interested readers to more detailed information.

REFERENCES

Burt, Peter J., and Edward H. Adelson. 1983. "A Multiresolution Spline with Application to Image Mosaics." *ACM Trans. Graph.* 2 (4): 217–236. doi:10.1145/245.247.

Ciptasari, Rimba W. 2016. *"Single Authentication: Exposing Weighted Splining Artifacts."* In: 9869:98690I–98690I–9. doi:10.1117/12.2222513.

Ciptasari, Rimba Whidiana, Kyung Hyune Rhee, and Kouichi Sakurai. 2013. "Exploiting Reference Images for Image Splicing Verification." *Digital Investigation* 10 (3): 246–58. doi:10.1016/j.diin.2013.06.014.

Johnson, Micah K., and Hany Farid. 2005. "Exposing Digital Forgeries by Detecting Inconsistencies in Lighting." In *Proceedings of the 7th Workshop on Multimedia and Security*, 1–10. MM&Sec '05. New York, NY, USA: ACM. doi:10.1145/1073170.1073171.

Johnson, Micah K., and Hany Farid. 2007. "Detecting Photographic Composites of People." In *Digital Watermarking*, 19–33. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-92238-4_3.

Kee, E., and H. Farid. 2010. "Exposing Digital Forgeries from 3-D Lighting Environments." In *2010 IEEE International Workshop on Information Forensics and Security*, 1–6. doi:10.1109/WIFS.2010.5711437.

Nugraha, P. A., R. W. Ciptasari, and B. Pudjoatmojo. 2015. "A Photo Composite Detection Based on Eye Specular Highlights Using Pixel-Based Approach." In *2015 3rd International Conference on*

Information and Communication Technology (ICoICT), 195–200. doi:10.1109/ICoICT.2015.7231421.

O'Brien, James F., and Hany Farid. 2012. "Exposing Photo Manipulation with Inconsistent Reflections." *ACM Trans. Graph.* 31 (1): 4:1–4:11. doi:10.1145/2077341.2077345.

Popescu, A. C., and H. Farid. 2005. "Exposing Digital Forgeries by Detecting Traces of Resampling." *IEEE Transactions on Signal Processing* 53 (2): 758–67. doi:10.1109/TSP.2004.839932.

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 5

DEEP LEARNING FOR ABNORMAL BEHAVIOR DETECTION

Nian Chi Tay[1,*], Pin Shen Teh[2] and Siok Wah Tay[3]
[1]Faculty of Information Science and Technology,
Multimedia University, Malacca, Malaysia
[2]School of Computer Science, University of Manchester,
Manchester, UK
[3]Department of Computer Science, University of Bath, Bath, UK

ABSTRACT

Abnormal behavior detection has become more crucial nowadays in the field of public safety. Many methods have been conducted to detect abnormal behavior using the technique of computer vision. In this book chapter, it will cover the basic concepts of artificial intelligence, representation learning, machine learning, neural network, deep learning and convolutional neural network (CNN). Deep learning is a state-of-the-art method used in this work to detect abnormal behavior with Web Dataset and UMN Dataset for training and testing.

Keywords: deep learning, abnormal behavior detection, neural network

* Corresponding author, Malaysia, Email: nianchi.tay95@gmail.com.

INTRODUCTION

The rising trend of security issues has been a hot topic for society nowadays. Now and then, there are news on criminal crimes like robbery, fighting cases and especially terrorism has grown more rapid recently in crowded places (Fox and Gilbert, 2016). Hence, many monitoring devices have been used to ensure public safety in places for instance schools, subway stations and shopping malls.

However, humans are incapable of effectively identifying the criminals by monitoring security cameras all the time, it is considered as an impossible task for human. Thus, this is when artificial intelligence (AI) and computer vision come into play.

ARTIFICIAL INTELLIGENCE

Artificial intelligence can be defined in many ways. The following are the four definitions from the dictionary "The New International Webster's Comprehensive Dictionary of the English Language, Encyclopedic Edition" (Smith, 2003):

- Artificial intelligence is an area of study in the field of computer science. It is concerned with the development of computers able to engage in human-like thought processes such as learning, reasoning, and self-correction.
- Artificial intelligence is the concept that machines can be improved to assume some capabilities normally thought to be of human intelligence such as learning, adapting, self-correction, etc.
- Artificial intelligence is the extension of human intelligence through the use of computers, as in times past physical power was extended through the use of mechanical tools.

- Artificial intelligence is, in a restricted sense, the study of techniques to use computers more effectively by improved programming techniques.

COMPUTER VISION

Computer vision resides in the field of artificial intelligence. It is a technology that provides a computer system with a visual understanding of the world, in short, is to help a computer system to "see."

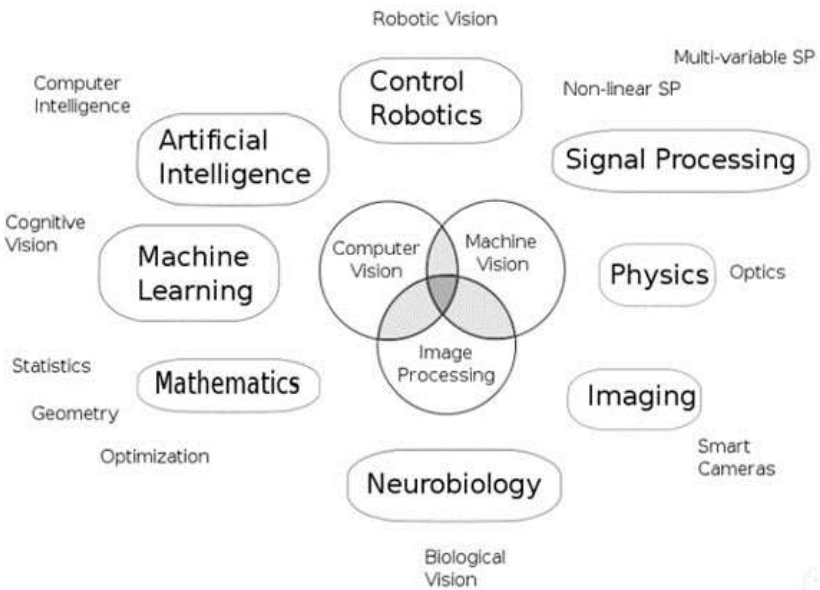


Figure 1. Main components of machine understanding (Kaiser, 2017).

Before discussing the computer vision approaches used for abnormal behavior detection, the understanding of computer vision and what are the processes involved are needed. The goal of computer vision is to make useful decisions about real physical objects and scenes based on sensed images by going through a series of steps.

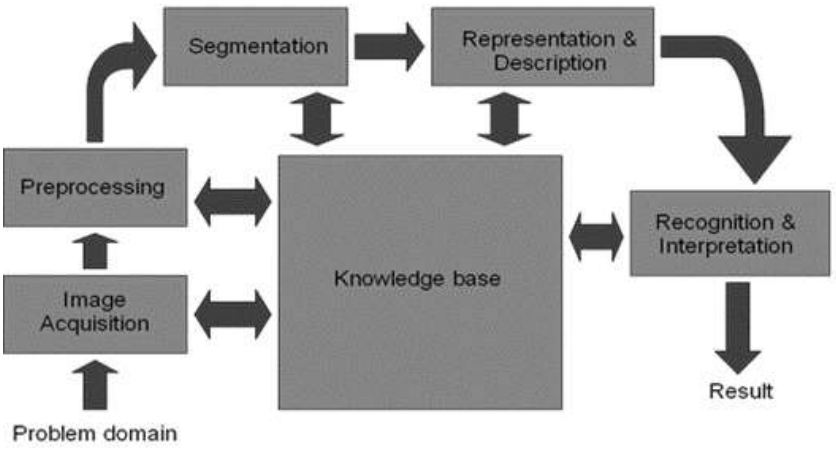


Figure 2. Fundamental steps in computer vision (Forsyth, 2014).

As shown above, there are a total number of five fundamental steps in computer vision as discussed earlier: (i) Image Acquisition, (ii) Preprocessing, (iii) Segmentation, (iv) Representation & Description and (v) Recognition & Interpretation.

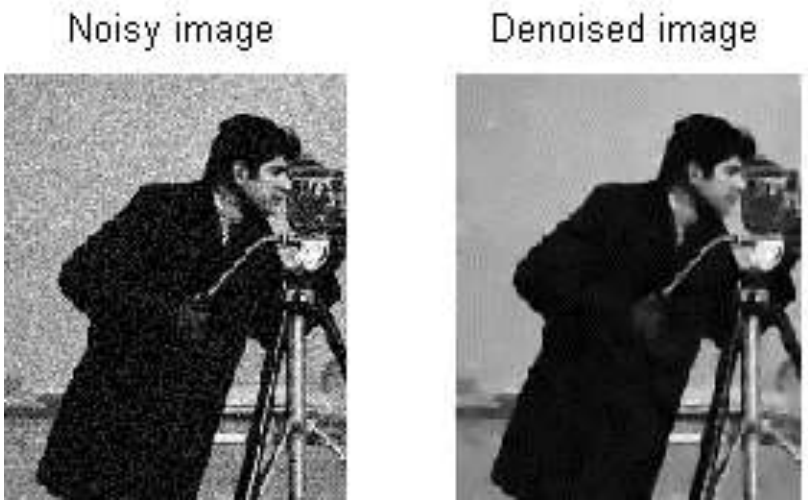


Figure 3. Left shows the noisy image before preprocessing and a right shows a clearer image after removing the noise (Rudin, 1992).

The first step in computer vision is image acquisition. The meaning of image acquisition is to acquire a digital image that is transformed from the analog signal by going through the process of digitization. The process requires sensors such as webcams, surveillance cameras or infrared cameras to collect the data.

The second step in computer vision is image preprocessing. The purpose of doing image preprocessing is to enhance the quality of images obtained. The techniques for image preprocessing normally include noise removing, contrast enhancing, region isolation, etc.

The next step after preprocessing is segmentation. This process involves cropping out the Region of Interest (ROI) from the background of the image.

The next step in computer vision is representation and description. This step involves representing the raw pixel data obtained from segmentation as a boundary or an entire region. The decision can be made based on some criteria. If the focus of the image is on external shape characteristics like corners or inflections, then boundary representation is more suitable to be imposed. If the focus of the image is on internal properties such as texture or skeletal shape, then regional representation would be more appropriate to be used.

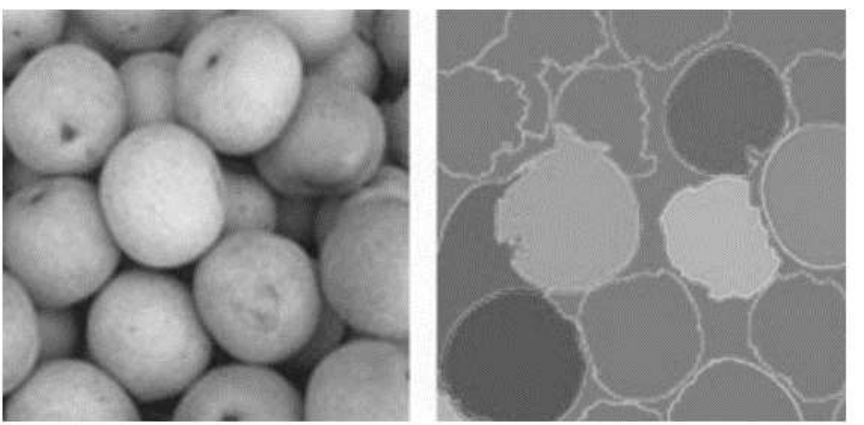


Figure 4. Right shows image after segmentation from the background (Mathworks, 2017).

Finally, the last step in computer vision is recognition and interpretation. This stage involves assigning outputs and meanings to the images, which is also known as classification of images into correct categories.

Today's modern surveillance system not only aims to install cameras in places to monitor and substitute the human eye but also to carry out surveillance automatically and autonomously through the use of computer vision. The cameras will capture every behavior and then analyzed by the system to identify if there is any abnormal behavior. The criteria for abnormal behavior vary depending on the situation. For instance, people walking in a shopping mall is considered normal behavior, whereas if there is anyone running inside or fighting, those will be identified as abnormal behaviors.

Abnormal behavior detection can be divided into two ways; one is using normal behavior database. Normal behavior database may contain actions like walking on a pedestrian crossing or running in a park. The actions which are not in the database are considered as abnormal behaviors. Other than that, another way is to use abnormal behavior database. Abnormal behavior database may contain actions such as running in a different way in a parade or fighting in a public place, and the actions which in the database are abnormal behavior. Next section will discuss the main approaches and state-of-the-art abnormal behavior detection methods.

Figure 5. Left shows the normal behavior of a crowd walking in a public area. Right shows the abnormal behavior of a crowd running in a public area (UMN Dataset).

EXISTING TECHNIQUES

Many approaches have been proposed to detect abnormal behavior in surveillance system. Various techniques have been used and combined in an attempt to obtain best results from the system. Some focus on the abnormal behavior in videos by recognizing specific actions. Other than that, optical flow approaches like social force model have been proposed to extract low-level motion features. Some have combined the low-level features to form a high-level histogram of optical flow orientation method. Besides, some use Histogram of Oriented Gradients (HOG) as feature extraction and combine with Support Vector Machine (SVM) as classifier. Of all the methods, deep learning is selected to be discussed in this book as it is considered as one of the most popular methods in recognition nowadays. Deep learning is a type of machine learning, which is a computer system that learns without explicitly programmed and improves based on past experiences on a task (Tom Mitchell, 1997).

Abnormal Behavior Analysis Using Latent Dirichlet Allocation

The authors propose Latent Dirichlet Allocation (LDA) to analyze abnormal behavior. (Haixian and Li, 2012) Gabor filter is used to extract space-time interest points from a series of spatial-temporal words obtained from a video input. The authors also remove the redundant points caused by the noise of the camera and the movement in neighboring frame. The feature extraction method used is 3D-sift features and algorithm used is k-means algorithm to cluster the points into categories of words. A video is processed as a text whereas the interest points are processed as the words. The LDA then assumes there are some latent points between the words and the text, and the video will be analyzed by the distribution of the topic. The authors can get a good accuracy even though the interest points used are few. The Weizmann human action datasets and KTH datasets are used to feed the algorithm.

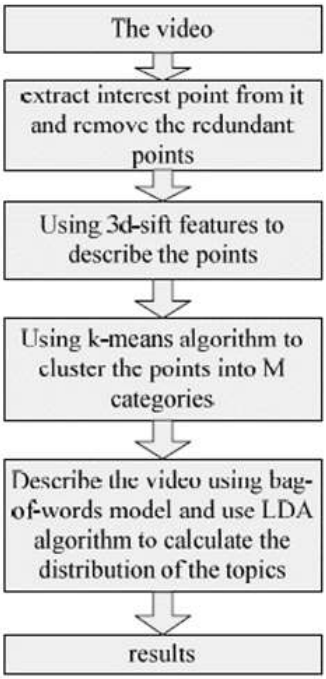


Figure 6. Flowchart of the approach (Haixian, 2012).

First, the authors use Gabor Filter and Gaussian Filter to extract the interest points from a video and remove its redundant points. Next, 3D-Sift features are used to describe the points. Then, k-means algorithm is used to cluster the points into M categories. Finally, the video is described using bag-of-words model and also LDA algorithm to calculate the distribution of topics.

Interest Points Extraction

Interest points are extracted using Gabor Filter and Gaussian Filter to apply to a video. However, the noise of the camera and the movement in neighboring frame can make a maximum sometimes, and those are called redundant points. The authors also extract the foreground by subtracting the background obtained from the video frame.

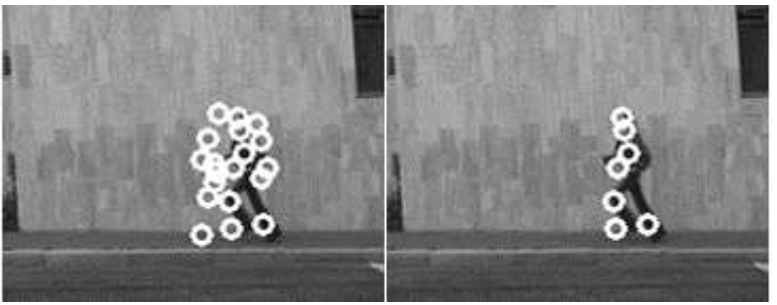


Figure 7. Left shows the interest points extracted with redundant points, right shows the interest points extracted excluded the redundant points (Haixian, 2012).

After obtaining the interest points, 3D-sift features are used to describe the points. The result obtained is a one-dimensional vector and 3D-sift features can be considered as a gradient histogram. K-means algorithm is used to cluster the points into M categories after getting the interest points and features.

Behavior Analysis Using LDA

LDA assumes the documents as random mixtures over latent topics, and each topic is grouped distribution of words. The process of how to produce a word in the document is shown in the steps below:

1. Choose a parameter $\theta \sim p(\theta)$
2. For each of the M words w_i:
   a. Choose a topic z_k, $p(z_k) \sim p(\theta)$
   b. Choose a word w_m from the topic, and $p(w_m) \sim p(w_m|z_k)$

RESULTS

The results obtained from the abnormal behavior analysis using space-time interest point and LDA have a high accuracy. If “side” and “skip” are

chosen as abnormal behavior in the dataset, the accuracy of the analysis is 80%. However, the accuracy of their method depends on which behavior is defined as abnormal behavior.

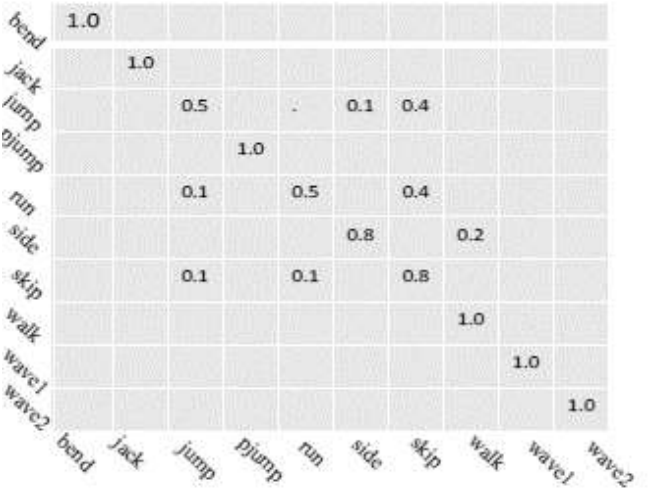


Figure 8. Results obtained (Haixian, 2012).

Abnormal Crowd Behavior Detection using Social Force Model

The authors introduce Social Force Model to detect and localize abnormal behavior in crowd videos. (Mehran, Oyama, and Shah, 2009) A grid of particles is placed over the image with the space-time average of optical flow. Social force model estimates the interaction forces between the moving particles. These interaction forces will be mapped into the image plane later to get Force Flow for each pixel in each frame. Spatio-temporal volumes of Force Flow are then randomly selected to model the abnormal behavior of the crowd. The authors classify the frames as normal and abnormal behavior by using bag of words approach. The regions of anomalies found in the abnormal frames are localized using interaction forces. The dataset from the University of Minnesota for escape panic scenarios are used.

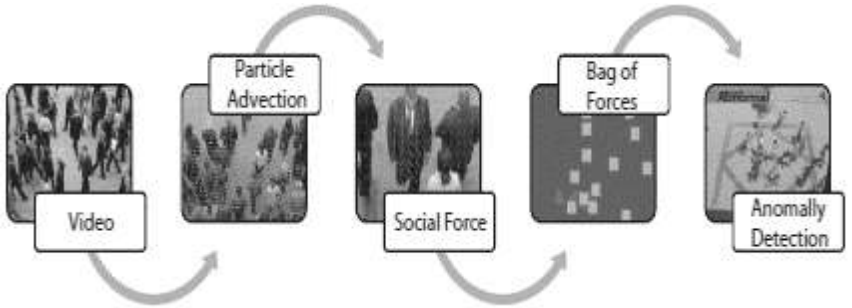


Figure 9. Summary of the approach for abnormal behavior detection in crowd videos (Mehran, 2009).

Social Force Model

The authors describe the social force model for pedestrian motion dynamics by considering personal motivations and environmental constraints.

The particles moving by optical flow is like the flow of leaves over the water. This analogy helps in understanding how the social force model takes part in particle grid. For example, the leaves themselves have a different velocity than the average flow when there is an obstacle, branching of the fluid or something unusual. Thus, the authors conclude that particles also react the same. They are capable of revealing any divergent flows from the average flows when some unusual scenarios happened.

Localization of Abnormalities

The authors use LDA model with force flow to categorize normal frames and abnormal frames. However, the method does not implicitly localize the unlikely visual words. The force flow is corresponding to the interaction forces in the frames. The active regions which have many social interactions are recognized as abnormal behavior. Thus, the abnormalities are localized by locating the high force flow regions.

RESULTS

Abnormal Behavior Detection in Crowd Scenes Using Hybrid Histogram of Oriented Optical Flow

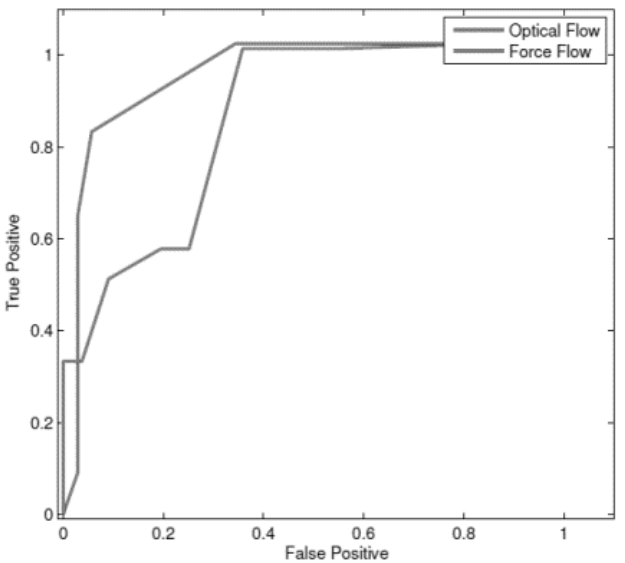


Figure 10. Social flow model method outperforms the pure optical flow method (Mehran 2009).

The authors present a new feature descriptor, which is called the hybrid optical flow histogram (HOFH). (Wang, Ma, Luo, Liu, and Zhang, 2016) The descriptor is mainly focused on the information on the movement by using the concept of acceleration. The method can indicate the change of speed in every direction of a movement. Spatial and temporal region saliency determination method is also used to extract the effective area only for samples to reduce the computational costs. Sparse representation is applied too to help in abnormal behaviors detection because it is very stable and has high accuracy. University of Minnesota (UMN) datasets are used in the experiments and the results obtained are proven to be more accurate compared to existing algorithms.

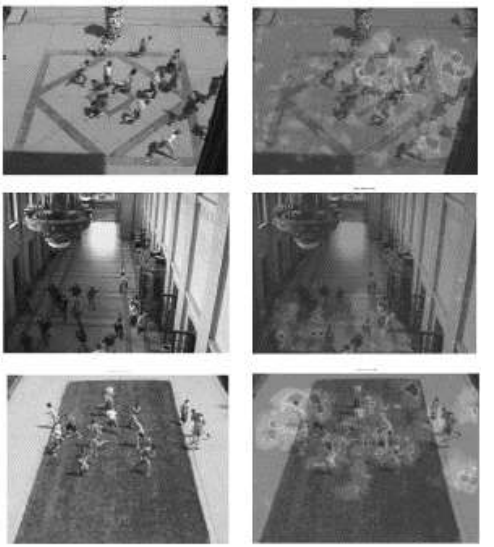


Figure 11. Localization of abnormal behaviors in different scenes using the interaction force. Left are the original frames, right are the localized abnormal behaviors (Mehran, 2009).

Feature Extraction Using HOFH

HOFH is used to depict any motion changes happened between video frames. The optical flow of an image needs to be computed first to obtain the statistical distribution of optical flow in every direction. However, HOFH can compute the changes in optical flow for directions and magnitude from the statistical distribution obtained. It helps to express the optical flow acceleration.

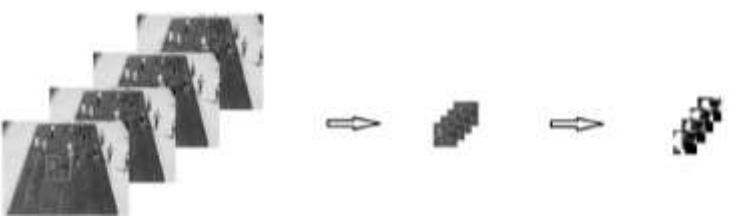


Figure 12. Feature extraction from full video frames to local spatial-temporal container after video segmentation to extracted feature in grayscale (Wang, 2016).

Sparse Reconstruction

Sparse representation is used to detect abnormal behavior in crowd scenes because it can achieve good application effects in noise reduction and face recognition. It also works well in high-dimensional training with few samples. The author computes the sparse reconstruction cost (SRC) to identify abnormal behavior from normal behavior. The formula is as shown below:

$$\text{SRC} = \frac{1}{2} \| y - \Phi \cdot x* \| \frac{2}{2} + \lambda \| x* \|_1. \quad (1)$$

The result will be classified into normal sample if its SRC is less than the threshold and vice versa.

RESULTS

The results obtained show high accuracy. The salient visual areas can be determined by selecting only the interesting areas in the frames. Sparse representation theory represents the samples while SRC determines which one is abnormal behavior.

Figure 13. Frames with normal behaviors (Wang, 2016).

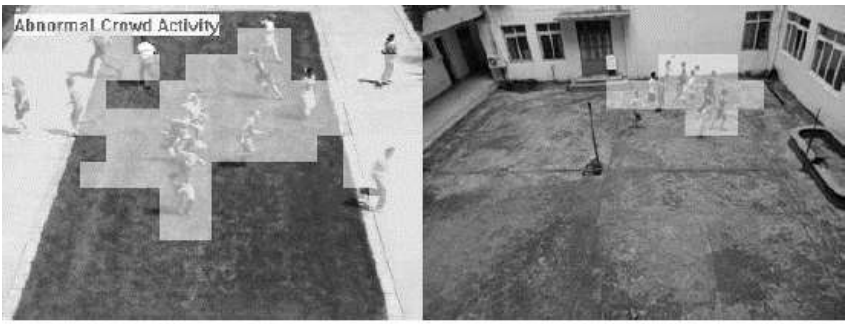


Figure 14. Frames with abnormal behaviors (Wang, 2016).

Person Detector Using Histograms of Oriented Gradients (HOG)

HOG and Support Vector Machine (SVM) approach to human detection has proven to be one of the most popular and successful among other human detection techniques. HOG is a feature descriptor that generalizes the same object from different frames to the same feature descriptor. HOG has made classification of object easier, and the concept of HOG is simpler to understand. HOG focuses on global feature rather than a series of local features. In short, it represents the human in a single feature vector rather than many smaller parts of human. HOG person detector uses sliding window technique to move around the image (Dalal, 2006). A HOG descriptor is then computed at each position of detector window. The image will be subsampled to multiple sizes to help recognize persons at different scales.

Gradient Histograms

The author uses a detection window of 64 pixels wide by 128 pixels tall. Every image is cropped into 64x128 windows. The cells with 8x8 pixels are used inside every detection window.

Figure 15. Small box shows the 8x8 pixels cell in a detection window (Dalal, 2006).

A 9-bin histogram ranges from 0 to 180 degrees is built from 64 gradient vectors.

Block Normalization

The author uses the blocks of size 2 cells by 2 cells with 50% overlap in each detection box.

The block normalization can be performed by linking the histograms into a vector with 4 histograms and 9 bins per histogram components. The vector is then divided by its magnitude to normalize it.

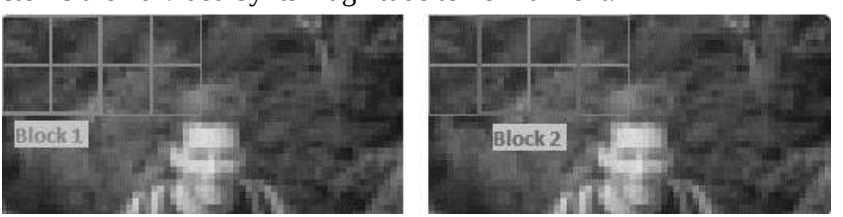


Figure 16. Block normalization (Dalal, 2006).

OUR APPROACH

In this book work, deep learning approach is applied to detect abnormal behaviors. Deep learning has been applied in many fields range from computer vision, machine translation, object recognition, speech recognition, video games, to bioinformatics, etc.

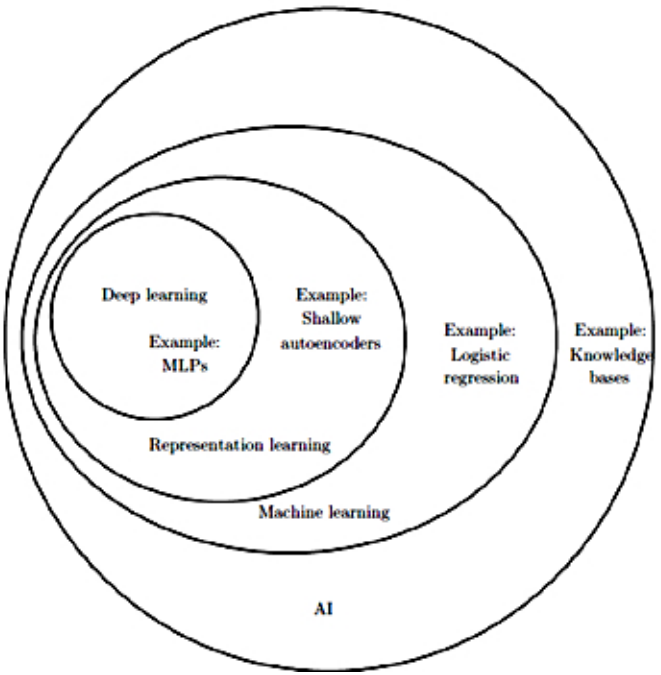


Figure 17. Venn diagram shows that deep learning is a kind of representation learning, which is a kind of machine learning under the field of AI (Goodfellow, 2016).

The basic concepts of representation learning and machine learning will be discussed first before going into deep learning.

Representation Learning

Representation learning is a technique to represent data using features that can be recognized and processed by a computer system. For example,

given an image with different types of fruits, from here the system will represent the fruits inside the image with features such as the color of fruits, the size of fruits, the skin texture of fruits, the shape of the fruits and so on. Representation learning will help choose the best features to be used in machine learning.

Machine Learning

There are two definitions of machine learning, the informal definition is that machine learning is: "the field of study that gives computers the ability to learn without being explicitly programmed." (Arthur Samuel, 1959).

Tom Mitchell provides a more modern definition which is that machine learning is: "A computer program that is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

For example, a task is given to make a computer system that can play tic-tac-toe. From the second definition:

E = experience in playing many games of tic-tac-toe
T = task of playing tic-tac-toe
P = probability that the system will win the next game

Machine learning can be divided into two general classifications: supervised learning and unsupervised learning.

In supervised learning, the sample inputs are given to the system together with their respective outputs. For example, a dataset of fruits is given to the system with their outputs as apple, orange, mango, pear, etc. The features will then be fed into the system to be classified and the result is saved in a model.

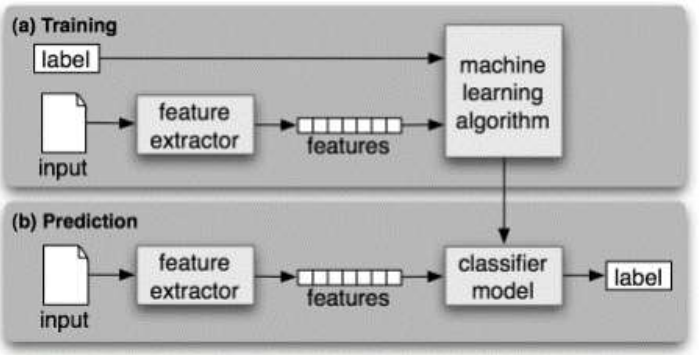


Figure 18. Shows the process of supervised learning, some portions of the dataset will be used for training and the remaining portion will be used for testing (Bird, 2014).

In unsupervised learning, only input samples are given to the system. The system will then need to cluster the data into categories. For example, given a dataset of different types of shirts, the system will need to cluster the data into pattern a, b, c and so on.

Neural Network

Deep learning uses the concept of neural network, which is a network made up from a set of nodes with respective weights. A neural network consists of an input layer, one to many hidden layers and an output layer.

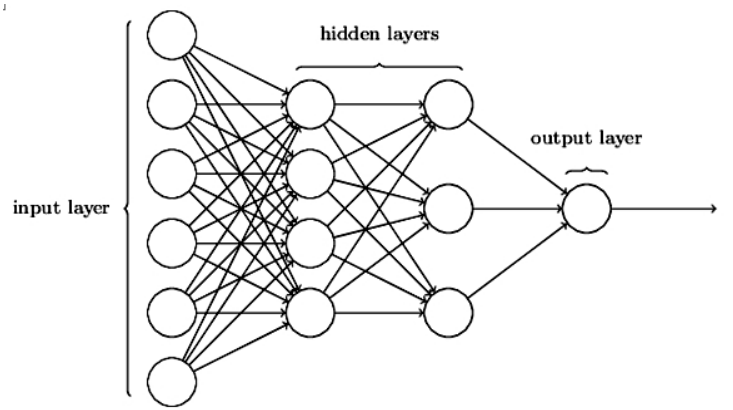


Figure 19. Example of a neural network (Nielsen, 2017).

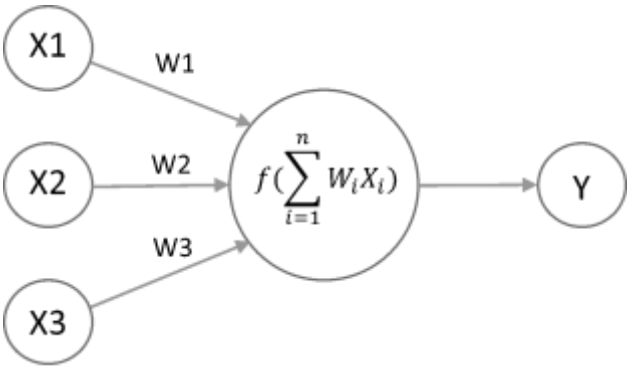


Figure 20. Weighted input summation (Lincoln, 2015).

The function $f(net) = \sum_{i=1}^{n} w_i\, x_i$ is the summation of the product of input value, x_i and respective weight, w_i.

The network architectures can be divided into three types: single-layer feed-forward, multi-layer feed-forward and recurrent.

A single-layer feed-forward network consists of one input layer and one output layer. A multi-layer feed-forward network consists of one input layer, one or more hidden layers, and an output layer.

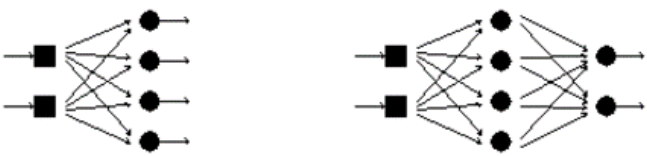


Figure 21. Left shows a single-layer feed-forward network, right shows a multi-layer feed-forward network (Jack, 1991).

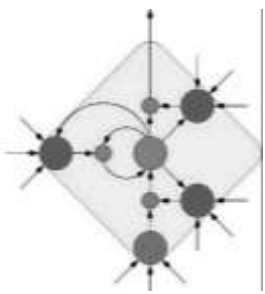


Figure 22. Shows a recurrent network (Schmidhuber, 2017).

A feed-forward network is acyclic, which means there are no cycles of neurons in the network. The data can only be passed from the input layer to the output layer. Hence, the network does not require memory as it will not change even when new data is given.

A recurrent network is cyclic and can have many cycles that are going back and forth in the network from input layer to output layer or the other way round. Thus, memory is needed to record the previous state of the input.

Deep Learning

Deep learning uses the concept of representation learning and machine learning in the form of a neural network. The term "deep" is used to represent the number of layers in the network. The more layers a network contains, the deeper the network will be. Deep learning has proven to achieve state-of-the-art results with high accuracy. For instance, deep learning has outperformed humans in classifying images and in 2016, AlphaGo, an AI computer program that plays the board game GO has won against the world's best GO player.

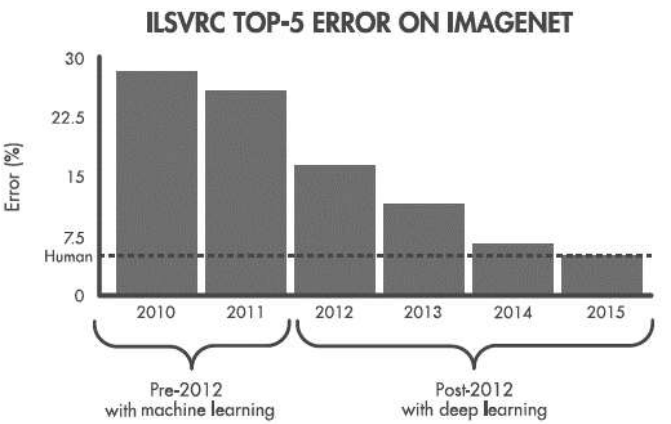


Figure 23. The bar chart shows that deep learning has become more accurate and eventually outperforms humans in classifying images (Mathworks, 2017).

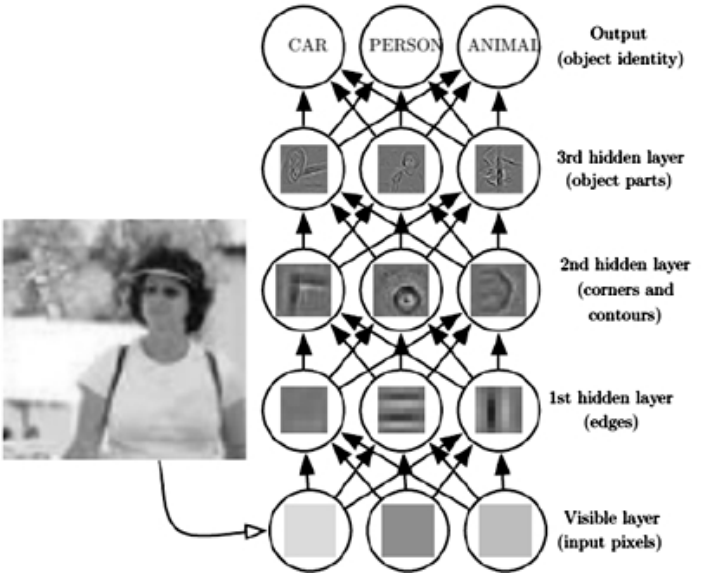


Figure 24. Shows how deep learning learns. (Goodfellow, 2016).

Deep learning learns from the labelled training dataset provided. The input of an image will be mapped into the input layer of the network in pixels form. After that, it will go through feature extraction in the hidden layers such as extracting the edges, corners, contours and object parts of the image. Users have no influence over what features are being selected, the network itself will learn directly from the input samples. Lastly the object will be classified into the correct output (Goodfellow, 2016).

Convolutional Neural Network

In this book work, Convolutional Neural Network (CNN or ConvNet) is used for detecting abnormal behavior as it is one of the most popular deep learning algorithms for image or video classification. CNN consists of two main parts: feature detection layers and classification layers (Mathworks, 2017).

In feature detection layers, the input samples will undergo three operations such as convolution, pooling and Rectified Linear Unit (ReLU).

Convolution layer helps to filter the input image and activate certain features of the image. Pooling layer will carry out downsampling to reduce the number of parameters in the network. ReLU layer maps negative values to zero and ensure there are only positive values in order to allow faster training in the network. These layers will be repeated over many times to learn different features of the image.

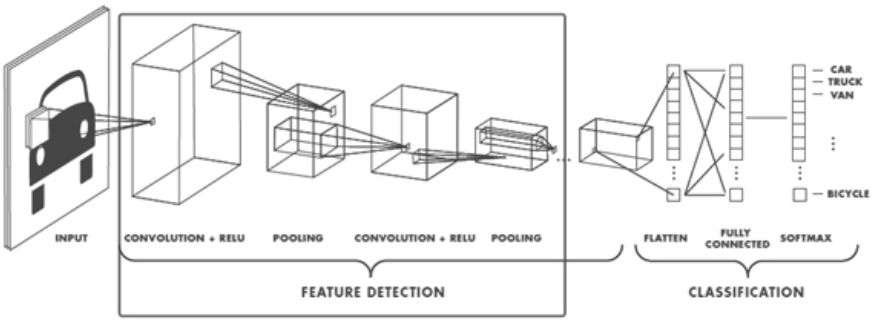


Figure 25. Shows the feature detection layers (Mathworks, 2017).

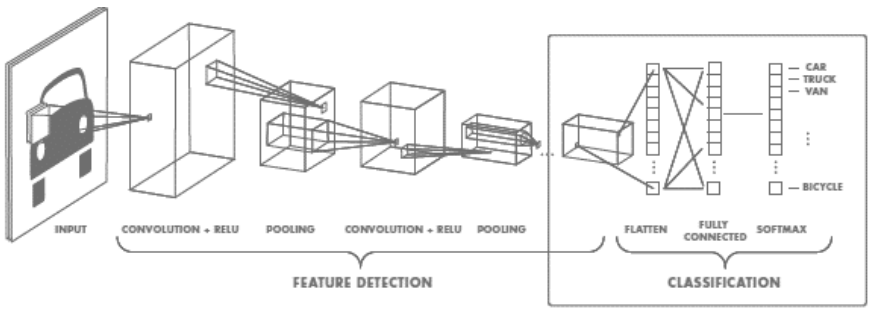


Figure 26. Shows the classification layers (Mathworks, 2017).

After performing feature extraction, the network will perform classification. The fully connected layer (FC) outputs the number of classes to predict. The output vector from FC contains the probabilities for each class of the input after classification. The final layer in the CNN is the softmax layer. This layer helps to provide the output of classification (Mathworks, 2017).

Deep learning often requires large amount of inputs to obtain the best accuracy. It also relies heavily on the computational power and requires high-performance GPU.

When training a deep learning network, one may choose to train the network from scratch or retrain the network using transfer learning approach. Transfer learning may result in faster training as it fine-tunes an existing network to become a new network based on your application.

Experiments

In this book chapter, 2 datasets have been chosen in the abnormal behavior detection using CNN.

Web Dataset

This dataset contains 8 videos of real-life escape panic, clash, fight and 12 videos of normal pedestrians in several places. There are a total of 15766 images; each abnormal class and normal class has 5000 images respectively for training. The images for training and testing are chosen randomly. The experiments had been carried out for 10 times to obtain the average accuracy. This dataset has achieved a high accuracy of 98.49%.

Figure 27. Left shows the images detected for normal behavior, right shows the images detected for abnormal behavior.

UMN Dataset

This dataset contains 11 videos of crowd escape panic. There are a total of 7739 images, each abnormal class and normal class has 700 images respectively for training. The images for training and testing are chosen randomly. The experiments had been carried out for 10 times to obtain the average accuracy. This dataset has achieved a high accuracy of 99.87%.

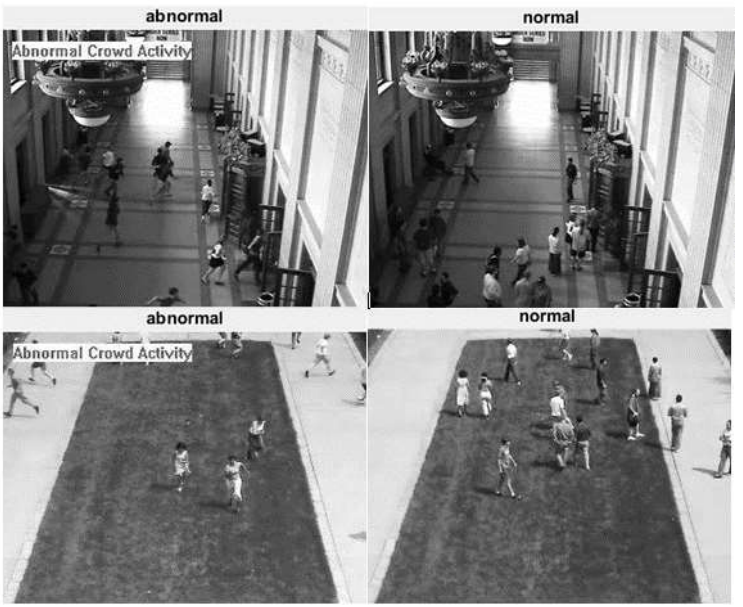


Figure 28. Left shows the images detected for abnormal behavior, right shows the images detected for normal behavior.

Both datasets use small size of images for training to decrease the training time. The second dataset has lesser training data; hence the number of epochs has been increased to obtain better result. One epoch is equal to one forward pass and one backward pass of the training samples in the network. The learning rate of network can be adjusted until the best result is obtained. Lower learning rate may result in slow training time, however higher learning rate tend to overshoot the global minimum during network training.

Table 1. Shows the parameters used in training the network for both datasets

Dataset	Web Dataset	UMN Dataset
Size of images	32x32 pixels	32x32 pixels
No. of training samples	10000	1400
No. of testing samples	5766	6339
No. of epoch	5	100
Learning rate	0.001	0.001
Accuracy	98.49%	99.87%

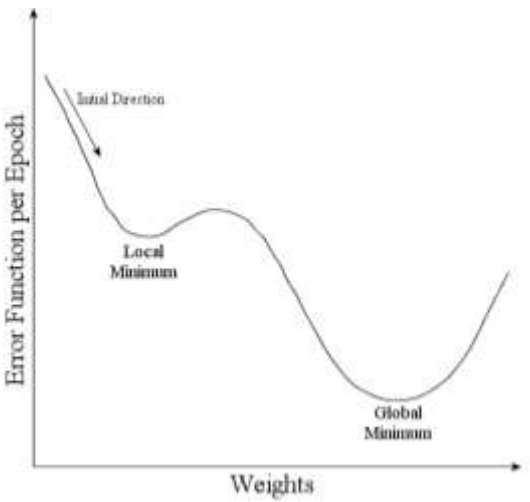


Figure 29. The goal of training algorithm is to reach the global minimum with the least of time used. (H. Corr, 2017).

From the deep learning experiments conducted in this book work, we can get the idea that deep learning actually requires a large amount of input samples to perform well. Other than that, some of the parameters might also affect the accuracy of the result such as:

- Size of the images
- Learning rate
- Number of epochs
- Number of layers

In order to achieve high accuracy, some adjustments to the parameters and the input data are needed. Preprocessing of images should be conducted to enhance the quality of the input samples.

CONCLUSION

This book chapter discusses the usage of deep learning in the context of abnormal behavior. It covers the basic concepts of artificial intelligence, computer vision, other techniques used for abnormal behavior detection, representation learning, machine learning, neural network, deep learning and convolutional neural network approach.

Based on the experiments conducted, CNN shows promising result. It is able to achieve an average accuracy of 99.18% on both Web and UMN datasets. Future work is needed to achieve more accurate result in the least of time and with lesser computational power consumed.

Possible future work could be increase the number of training samples, include more types of abnormal behavior and also employ the method of Deep Convex Network (DCN) which has higher accuracy with less training time needed.

REFERENCES

Bird, Steven, Ewan Klein, and Edward Loper. 2014. "*Natural Language Processing with Python."* Accessed May 15, 2017. http://www.nltk.org/book/ch06.html.

Corr, H. Patrick. "Techniques: The Multi-Layer Perceptron" Accessed July 5, 2017. http://www.qub.ac.uk/mgt/intsys/backprop.html.

Dalal, N., and Triggs, B. 2005. "Histograms of oriented gradients for human detection." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Accessed June 1, 2017. doi: 10.1109/CVPR.2005.177.

Forsyth A., David, and Jean Ponce. 2014. *"Computer Vision: A Modern Approach (2nd Edition)."* London: Pearson.

Fox, Kara, and Dave Gilbert. 2016. *"Terror attacks in developed world surge 650% in one year"* Accessed June 1. http://edition.cnn.com/2016/11/16/world/global-terrorism-report/index.html.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. "*Deep Learning*". Cambridge: MIT Press.

Haixian, Lu, Li, Guo, Shu, Gui, and Jinsheng, Xie. 2012. "Abnormal behavior analysis using LDA". *IEEE Xplore* 96-100. Accessed June 4, 2017. doi: 10.1109/ICALIP.2012.6376593.

Hugh Jack, 1991. *"Application Of Neural Networks To Motion Planning And Control For An Articulated Robot Arm."* Accessed May 7, 2017. http://engineeronadisk.com/V2/hugh_jack_masters/engineeronadisk.html.

Kaiser, Adrien. 2017. *"What is Computer Vision?".* Accessed June 3. http://hayo.io/computer-vision/.

Lincoln, Nicholas. 2015. *"Identifying Subatomic Particles with Neural Networks".* Accessed June 10, 2017. http://2centsapiece.blogspot.my/2015/10/identifying-subatomic-particles-with.html.

Mathworks. 2017. *"Introducing Deep Learning with MATLAB".* Accessed June 5, 2017. https://www.mathworks.com/campaigns/products/offer/deep-learning-with-matlab.html.

Mathworks. 2017. *"Marker-Controlled Watershed Segmentation."* Accessed June 2, 2017. https://www.mathworks.com/help/images/examples/marker-controlled-watershed-segmentation.html?requestedDomain=es.mathworks.com.

Mehran, R., Alexis Oyama, and Mubarak Shah. 2009. "Abnormal crowd behavior detection using social force model." *IEEE Conference on Computer Vision and Pattern Recognition* 935–942. Accessed May 20, 2017. doi: 10.1109/CVPR.2009.5206641.

Nielsen, Michael. 2017. *"Using neural nets to recognize handwritten digits".* Accessed May 5, 2017. http://neuralnetworksanddeeplearning.com/chap1.html.

Rudin, Leonid I., Stanley Osher and Emad Fatemi. 1992. "Nonlinear total variation based noise removal algorithms" *Physica D.* 60: 259–268. Accessed May 20, 2017. doi:10.1016/0167-2789(92)90242-f.

Schmidhuber, J. 2017. *"Recurrent Neural Networks."* Accessed July 1, 2017. http://people.idsia.ch/~juergen/rnn.html.

Smith, S. Stephenson, trans. 2003. *New International Webster's Comprehensive Dictionary of the English Language, Deluxe Encyclopedic Edition.* Florida: Trident Pr Intl.

"Unusual crowd activity dataset of University of Minnesota." Accessed May 1, 2017. http://mha.cs.umn.edu/movies/crowdactivity-all.avi.

Wang, Qiang, Qiao Ma, Chao Hui, Luo Hai, Yan Liu, and Can Long Zhang. 2016. "Hybrid Histogram of Oriented Optical Flow for Abnormal Behavior Detection in Crowd Scenes." *International Journal of Pattern Recognition and Artificial Intelligence*, 30(2): 1–14. Accessed May 21, 2017. doi:10.1142/S0218001416550077.

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 6

SECURITY ISSUES IN WIRELESS SENSOR NETWORKS AND IOT

Jayakumar Vaithiyashankar*
Faculty of Information Science and Technology,
Multimedia University, Malaysia

ABSTRACT

Wireless sensors are gaining tremendous popularity among members of the computing world. Due to the blooming status of the Internet of Things (IoT), the interconnectedness of the world revises every nook and corner of the digital world. Understanding the layout of IoT and wireless sensor networks enhance better implementation strategies. Different types of security issues will arise during the integration of these technologies. Systematic approaches will help solve underlying security loopholes and secure them against possible acts of compromise. Comparisons of various security algorithms and authentication protocols give a clear view about IoT integration with the wireless sensor network.

Keywords: wireless sensor, encryption, IoT, secure routing protocol, 6LoWPAN

* Corresponding Author, Email: jayakumarksrit@gmail.com.

INTRODUCTION

Recently, developments of low-cost electronic components have leveraged the growth of wireless sensor nodes. They consist of electro-mechanical combinations to sense various types of environmental factors, convert them into meaningful signals and transmit them as vital pieces of information (Chavan, Ajit A., and Mininath K. Nighot., 2016). The IoT has become the inevitable technological shift in the modern digital world. Moreover, applying the group of sensor nodes at a large scale will present Internet users with a tremendous scope of various practical applications.

Wireless sensor nodes are able to sense various types of environmental factors; depending upon the sensing components, it could observe temperature, humidity in the atmosphere, movement of vehicles and the pressure of their surroundings, and present real-time parameters of an object like velocity, acceleration, vector direction and volume.

Practical applications of the wireless sensor are numerous and there is no limit for its ability to blend with different domains. It could be used in military applications for secure communication, as a body area network for military personnel, and observe the paramilitary groups, the status of the equipment and the level of backup ammunition. Surveillance of the battlefield becomes easier in cases of the intelligent wireless sensor implementation to face off high-altitude terrain or challenging geographical locations. The antimissile and anti-ballistic launchers could be directed and guided with the help of accurate sensors.

The ultimate integration of a large sensor network could serve as a national level project and plays a vital role in environmental applications. It could be used for forest fire detection, flood alert systems, tsunami detection, precision agriculture, environmental monitoring, natural catastrophic disaster management, etc.

Background

The evolution of the wireless sensor network is studied in conjunction with the development of the electronic component hierarchy. The sensor is

initially used in the industrial application for tracking and monitoring the process chain. It is then slowly implemented in military applications with precise accuracy.

The sensor node consists of the sensor interface; it senses the environmental factors and converts the physical signal into an equivalent electronic signal, which is passed to the analog circuit and integrated with the micro-controller (Ritu Sharma et al., 2010).

This particular micro-controller is specifically designed for each type of sensor and process the sensed signal into the specific frame or packet. It will transmit either raw data or some processed data depending on the architecture of the sensor node. Finally, the radio system is used to transmit the data in exchange for communication, which is equipped with the battery.

IoT and Wireless Sensor Network

The Internet of Things (IoT) is the rapid technological advancement that leverages each and every electronic component in order to communicate with other devices to accomplish meaningful objectives and goals. So, IoT provides the common framework and provisions to communicate with a standardized form that leads to the automated machine-to-machine communication (Ritu Sharma et al., 2010).

Characteristics

The characteristics of the wireless sensor network play a vital role in understanding the functioning of sensors and they are listed as below:

Large Scale

Data gathering and processing will provide meaningful information only if there is a large number of nodes connected together because the sensor nodes are small and they are prone to errors. So, if there is a large number of nodes, then the errors are negligible.

Limited Resources

Due to the tiny size of the sensor node, each resource has very limited resources and limitations, such as the battery, computation power, and transmission range. So, each operation takes place inside the sensor node and becomes precious; they reflect the network lifetime and impact every action (Dener, Murat, 2014).

Redundancy

Redundancy may occur during sensing, storing, and transmitting the sensed data. The degree of redundancy should be kept to a minimal to ensure that the network lifetime should not be sacrificed in terms of processing unnecessary data (Dener, Murat, 2014: Rajeswari, S. Raja and V. Seenivasagam, 2016).

Security

Applications of sensor network spread over the military, medical, and even national level data integration security protocols and tamper-proof communications. Due to the limited resources, the security of the sensor network should be designed and constrained specifically based upon the application. It should override and avoid traditional methods due to the constrained resources (Dener, Murat, 2014: Rajeswari, S. Raja and V. Seenivasagam, 2016).

Security Requirements

The security requirements (Dener, Murat, 2014; Rajeswari, S. Raja and V. Seenivasagam, 2016).of the wireless sensor network are listed in the wireless transmission background and help to pivot basic underlying concepts.

Data Confidentiality

The data collected in the sensor network is sensitive in nature for particular applications like military and medical applications. Hence, the

data should not be revealed to neighbors and high-level confidentiality needs to be maintained. The confidentiality could be maintained by a secure key and encryption algorithms.

Data Integrity

Data transmitted through a network should be secured enough to prevent attackers from altering the original data. Data integrity needs to be maintained to preserve network confidentiality; simultaneously, preventive actions must be taken, such as cyclic codes and encrypted messages.

Data Authentication

While transmitting data, there is the possibility of malicious attackers to behave like the authorized node in that network. They will mimic the same response and style of the normal node. So, there is a crucial step needed to preserve and persist the authenticity of the respective node behavior via cryptography techniques.

Data Freshness

Some applications need the real-time data to immediately respond to the events. With this in mind, the delayed data is not meant to be processed by the sensors and systems mission failure. To ensure data is fresh and conveyed in real-time, the time should be stamped during the data generation.

Availability

Data availability is the capacity of the sensor network to remain active and withstand any type of denial of service attack. The network should have reliability and emergency backup arrangements to maintain its uninterpretable service.

Security Requirements for IOT

The security requirements of the Internet of Things looks almost similar to the sensor network, but there is a slight difference in a more

interconnected nature to the Internet makes it a special case, and careful attention is needed.

Resilience to Attacks

The network should stay away from single point failures; it needs to withstand against the malicious attacks. The self-healing property and self-organizing nature will help the network to re-organize itself from such malicious attacks.

Data Authentication

The data transferred from a source to its destination path may involve a number of intermediate hops. Thus, the data needs to be authenticated by verifying the credentials of data and must ensure it can be matched with the respective local templates. Only a particular amount of data has access to the network.

Access Control

Data accessed by various users at different access privileges in crossover areas leads to data instability. Hierarchy level and priority should be established to make data access comfortable. The access control should be implemented at the granular level for each user category without any collision.

End-User Privacy

At the end, the privacy of the end-user needs to be maintained without any compromise in the system. There should be very minimal interference between the service provider and the end-user.

Security Issues

The security issues are analyzed for both a wireless sensor network and IoT based architecture. Each layer and its functionality will reveal a better

understanding of working principles and related security issues (Gope, Prosanta, and Tzonelih Hwang, 2016).

Wireless Sensor Network Stack

The wireless sensor network protocol stack is depicted in Figure 1. The network protocol stack is a five-layer model as compared to the seven-layer OSI traditional model.

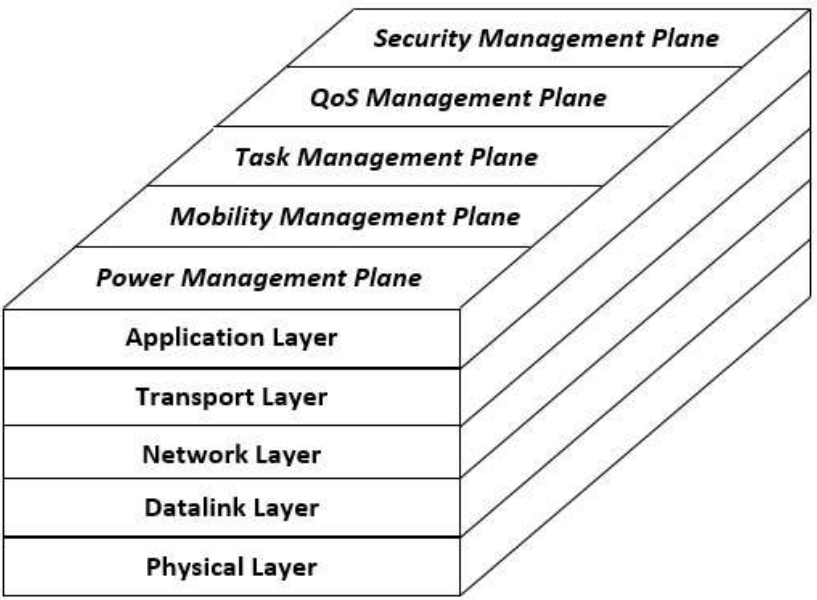


Figure 1. Wireless sensor network protocol stack.

Architecture of IoT Protocol Stack

The architecture of IoT protocol stack is compared with the traditional web stack for the easier comparison among available standards.

<table>
<tr><th></th><th colspan="2">IoT Stack</th><th>Web stack</th></tr>
<tr><td>TCP/IP Model</td><td>IoT Applications</td><td>Device Management</td><td>Web Applications</td></tr>
<tr><td>Data format</td><td colspan="2">Binary, JSON, CBOR</td><td>HTML, XML, JSON</td></tr>
<tr><td>Application layer</td><td colspan="2">CoAP, MQTT,XMPP, AMQP</td><td>HTTP, DHCP, DNS, TLS/SSL</td></tr>
<tr><td>Transport layer</td><td colspan="2">UDP, DTLS</td><td>TCP,UDP</td></tr>
<tr><td rowspan="2">Internet layer</td><td colspan="2">IPv6/IP Routing</td><td rowspan="2">IPv6, IPv4, IPSec</td></tr>
<tr><td colspan="2">6LoWPAN</td></tr>
<tr><td rowspan="2">Network/link layer</td><td colspan="2">IEEE 802.15.4 MAC</td><td rowspan="2">Ethernet(IEEE 802.3), DSL, ISDN, Wireless LAN(IEEE 802.11), Wi-Fi</td></tr>
<tr><td colspan="2">IEEE 802.15.4 PHY/ Physical Radio</td></tr>
</table>

Figure 2. IoT Protocol Stack.

Physical Attack

Jamming

The attack could take place over the interference of the respective radio frequency; this is called jamming. It causes the higher level of noise in the signal and makes it unusable (Dener, Murat, 2014; Rajeswari, S. Raja and V. Seenivasagam, 2016).

Tampering

It is the violation of access over a node by an attacker, thus making it harder to collect the vital data from the nodes and credential information. Tampering reveals that the secret and cryptographic keys that exist in the available packet.

Datalink Layer

Collision

This occurs when two nodes utilize the same frequency at the same time. Collision will result in network congestion and make it unusable during the peak time (Dener, Murat, 2014; Rajeswari, S. Raja and V. Seenivasagam, 2016).

Exhaustion

Frequent collisions will result in the decrease in resource availability, which also makes the network lifetime shorter. The network resources will be depleted at a tremendous rate (Rajeswari, S. Raja, and V. Seenivasagam, 2016).

Unfairness

The attack could take place by holding the whole network service with a modified availability time so that the quality of network service is downgraded.

Network Layer

Selective Forwarding

In selective forwarding, the suspicious nodes will forward the packets selectively based on some constraint results in permanent packet loss. This is highly destructive to the information exchange (Dener, Murat, 2014; Rajeswari, S. Raja and V. Seenivasagam, 2016).

Sinkhole Attack

The sink hole is the metaphor node which acts as a sink for all the nodes to receive data collectively. It is the simplified version of the selective forwarding method. It will produce greater damage to data integrity in the network (Dener, Murat, 2014).

Sybil Attack

The Sybil attack duplicates a node and its behavior over the network at various locations; it makes the network unstable. Redundancy in sensing and data transmission ultimately depletes the precious network resources (Dener, Murat, 2014; Rajeswari, S. Raja, and V. Seenivasagam., 2016).

Wormhole Attack

A wormhole attack is when data packets are received at one end of the node and are reproduced or diverted to some other node without any relevance. It introduces a network overhead and, as usual, the network resources get depleted as soon as possible (Rajeswari, S. Raja, and V. Seenivasagam., 2016).

Hello Flood Attack

The hello packets are flooded into the network at a higher bandwidth frequency to make it seem like every node has this malicious node associated with it. Thus, the routing table is added with an unnecessary false update, resulting in an overhead (Dener, Murat, 2014; Rajeswari, S. Raja, and V. Seenivasagam., 2016)

Encryption Algorithms and Authentication Protocols

The encryption algorithms are used to convert the normal plain text into unidentifiable cyber text with secret keys. The various encryption algorithms and authentication protocols are listed and compared with IoT compatibility (Tellez, Mauricio et al., 2016).

Authentication protocols are analyzed with three important parameters: The cryptographic method, DoS (Denial of Service) Resistance and communication overhead.

By comparing these three parameters, it is easier to interpret a respective protocol as suitable for the specific application or not. Then, evaluating the protocols for IoT compatibility is verified against each protocol such as resilience to attacks, data authentication, and access control and end-user privacy.

Security Protocols

The security protocols for the wireless sensor network are compared and checked against the compatibility with IoT abilities (Xiaomei, Yang,

and Ma Ke., 2016). Four parameters are taken into consideration to decide the IoT compatibility, such as resilience to attacks, data authentication, access control, and end-user privacy.

Table 1. Authentication protocol

Authentication Protocols	Cryptographic Method	DoS Resistance	Communication Overhead	IoT Compatibility
TESLA	MD5	No	Low	No
μ TESLA	MD5	No	Low	No
Multilevel μ tesla	MD5	Yes	Low	No
BABRA	MD5	Yes	Low	Yes
Unbounded key chains	SHA-1	No	Low	Yes
L-TESLA	MD5	No	Low	No
X-TESLA	MD5	Yes	Low	No
TESLA++	MD5	Yes	Low	No
RPT	MD5	No	Low	Yes
Hierarchical key chains	SHA-1	No	Very Low	Yes
Lightweight scheme	SHA-1	No	Very Low	Yes

Table 2. Security Protocol

	SPIN	**LEAP**	**TINYSEC**	**ZIGBEE**	**802.15.4**	**MINISEC**
Encryption	Yes	Yes	Yes	Yes	Yes	Yes
Freshness (CTR)	Yes	No	No	Yes	Yes	Yes
Overhead (Bytes)	8	Variable	4	4,8 or 16	4,8 or 16	4+3
MAC used	Yes	Yes	Yes	Yes	Yes	Yes
Key Agreement	Symmetric Deployed	Pre-Deployed	Any	Trust Center	-	Any
IoT Compatibility	Yes	Yes	No	Yes	Yes	No

Table 3. Security Requirements

	TinySec	SPINS	MiniSec	Lsec	LLSP	LISA	IEEE 802.15.4	LISP
Data Confidentiality	Yes	Yes	Yes	Yes	Yes	Yes	Yes	yes
Data integrity	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Data authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Data freshness	No	Yes	Yes	No	Yes	Yes	Yes	No
Data availability	No	No	No	No	No	No	No	Yes
Implementation	TinyOS (Mica2)	-	TinyOS (TelosB)	No	No	No	TinyOS (micaZ,TelosB)	No
IoT compatibility	Yes	Yes	No	Yes	Yes	No	No	yes

The security requirements are compared side by side with the different security protocols of the wireless sensor network to evaluate IoT compatibility.

DISCUSSION

The comparison of security and authentication protocols of the wireless sensor network with IoT compatibility reveals the details of correlation between security measures with interconnectivity. Whenever there is real-time monitoring and data connectivity in the application, this ultimately results in security checks for each and every data transaction. Suitable data encryption standards and encryption algorithms stand as the safeguard against the security attacks. The design of an algorithm solely depends upon the sensor network design and purpose of the network usage. The requirement of the sensor network application determines the level of security needed for the network. The limited resources and constraints make the designing of a wireless sensor network security challenge-able task possible. A better understanding of the network application and domain knowledge effectively implement security measures and achieve fruitful results. The integration of IoT and a wireless sensor network leverages multidimensional benefits across global data transfer and safer communication to properly deal with security issues.

REFERENCES

Chavan, Ajit A., and Mininath K. Nighot. 2016 “Secure and Cost-effective Application Layer Protocol with Authentication Interoperability for IOT.” *Procedia Computer Science* 78: 646-51. doi:10.1016/ j.procs.2016.02.112.

Dener, Murat. 2014, “Security Analysis in Wireless Sensor Networks.” *International Journal of Distributed Sensor Networks* 10(10): 303501. doi:10.1155/2014/303501.

Gope, Prosanta, and Tzonelih Hwang. 2016, “BSN-Care: A Secure IoT-Based Modern Healthcare System Using Body Sensor Network.” *IEEE Sensors Journal,* 16(5): 1368-376. doi:10.1109/jsen.2015.2502401.

Rajeswari, S. Raja, and V. Seenivasagam. 2016, "Comparative Study on Various Authentication Protocols in Wireless Sensor Networks." *The Scientific World Journal,* 2016:1-16. doi:10.1155/2016/6854303.

Ritu Sharma, Yogesh Chaba, Yudhvir Singh. 2010, “Analysis of Security Protocols in Wireless Sensor Network” *Int. J. Advanced Networking and Applications*, 2(3): 707-713.

Tellez, Mauricio, Samy El-Tawab, and Hossain M. Heydari. 2016, “Improving the security of wireless sensor networks in an IoT environmental monitoring system.” *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, doi:10.1109/sieds.2016.7489330.

Xiaomei, Yang, and Ma Ke. 2016, “Evolution of wireless sensor network security.” *2016 World Automation Congress (WAC)*, doi:10.1109/ wac.2016.7583032.

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 7

FINGER VEIN BIOMETRICS: THE FUTURE FOR A MOBILE AUTHENTICATION SYSTEM

Ahmad Syarif Munalih[1,*] and William Ardianto[2]
[1]Information Security Lab, MIMOS Bhd, Kuala Lumpur, Malaysia
[2]The Center of Applied Data Science, Kuala Lumpur, Malaysia

ABSTRACT

Today, mobile devices such as mobile phones, tablets and notebooks play essential roles in our daily lives. Most of our daily activities, like long distance communication via phone calls and messaging, documents production, and personal and confidential information storage are performed using mobile devices. Thus, mobile devices require a security system to allows only authorized users to access them. A traditional way of doing this is by implementing password authentication or grid lock pattern authentication, which is available in every smart phone. However, password and grid lock pattern are not convenient to the user since he/she needs to memorize the password and grid pattern. In addition, many users make mistakes when drawing the grid pattern lock especially when the

[*] Corresponding author, Email: asyarifm@yahoo.co.id.

pattern is drawn using a single hand. Several biometric techniques, such as face recognition and fingerprint recognition have been implemented to overcome these problems. Face recognition has been successfully deployed due to the wide availability of cameras in mobile devices. Fingerprint implementation is also popular with the availability of fingerprint scanners in high-end mobile devices. The security of the mobile devices has been enhanced with these biometric techniques. However, both the facial and fingerprint characteristics are visible externally and this makes them vulnerable to spoof attacks. For example, pictures of a person's face can be easily obtained from social media like Facebook and Instagram. Spoofing can be made by showing the pictures, videos, or 3D virtual models of the faces to the camera. A German hacker recreated a fake fingerprint using only pictures to deceive the fingerprint recognition system. To address these weaknesses, human authentication using features located inside the human body is a favorable solution. Finger vein recognition is one of the promising alternatives. Hitachi and Fujitsu are two big companies that are actively researching finger vein technology. Recently, they have created a small and thin finger vein scanner for mobile devices. With the invention of the compact finger vein scanner, smart devices embedded with finger vein technology will soon be available on the market.

Keywords: biometrics, finger vein, fingerprint, iris, face recognition, mobile authentication

INTRODUCTION

In the digital era, user authentication plays an important role in people's daily lives. Many day-to-day activities such as performing banking transactions, accessing emails and social media (Facebook, Twitter, etc.), and unlocking mobile devices require user authentication. Typically, the users are authenticated based on something they know such as passwords; something they possess like tokens, keys or smart cards; or something they are, which are their biometric characteristics.

Passwords are known as the most popular method for user authentication. The user is required to input his/her username and password before the access privilege is granted. However, passwords are not reliable enough as a simple and short password can be easily guessed

and are susceptible to easy spoofing. In contrast, a complex password is difficult to remember. Besides, passwords can be stolen by observers during the password entering process, and authenticity is subsequently invalid.

Meanwhile, token-based authentication is mostly used in access control. A token encapsulates the user's information in a compact device which is highly efficient. However, serious consequences will arise when the token is lost or stolen. Moreover, the token can be shared among users, hence it is not able to provide non-repudiation and effective authorization.

Biometrics is another authentication method that measures and identifies users based on their physiological and behavioral traits. Physiological biometrics refer to physical characteristics such as fingerprints, vein patterns, iris patterns, retina patterns, facial features, palm prints, and hand geometry. On the other hand, behavioral biometric traits refer to the user's behavioral patterns such as gait, signature, or keystroke dynamics. Biometrics are related to the unique physiological and behavioral characteristics of an individual, hence, it is difficult to counterfeit. Besides, these biological traits cannot be shared, forgotten, and are non-repudiative (Jain, Ross, and Pankanti, 2006). Hence, biometrics is known as a strong measure in personal authentication.

Finger Vein and Other Existing Biometrics

Any human physical and behavioral characteristic can be used as a biometric feature as long as its meets the following requirements (Jain, Ross, and Prabhakar, 2004):

- Universality: Every human has this characteristic.
- Distinctiveness/uniqueness: Every human has sufficient differences making it possible to distinguish between each other using this characteristic.
- Permanence: The characteristic does not change with time.

- Collectability: How easy the acquisition of the biometric characteristic can be performed.
- Performance: How good the accuracy of the system is and how fast the system can complete all the process (from acquisition until the matching process).
- Acceptability: How well a human can accept the use of this biometric characteristic in his/her daily life.
- Circumvention: How easily a fraudulent method can fool the system.

Nonetheless, each biometrics has its own advantages and disadvantages. Each biometric usage varies in its capabilities and effectiveness in addressing the specified application requirements and purposes. Table 1 outlines a comparison among the different biometric methods in terms of requirements (Jain, Bolle, and Pankanti, 1999).

Among the biometric traits, iris recognition, face recognition, and fingerprint recognition appear to be the most popular biometrics due to the rich information available and simple usage. These biometrics have been implemented in a wide variety of applications such as identification card, passport, and mobile devices. However, the iris, face and fingerprint features lay outside the human body and are exposed to the public. This makes them susceptible to forgery attack (spoofing), especially the face feature.

Face recognition is vulnerable to spoofing as the recognition system can be straightforwardly circumvented via presentation of a photograph or video recording of the genuine user (Biggio et al., 2012). The proliferation of high-resolution cameras, social media websites and phones equipped with both high-end cameras and screens, have made such impersonation easier. It is also possible to construct three-dimensional face models from multiple photos (“Virtual U: Defeating Face Liveness Detection by Building Virtual Models from Your Public Photos | USENIX”, 2017).

Table 1. Table of Comparison among Different Biometric Techniques (Jain, Bolle and Pankanti, 1999)

Biometrics	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	High	Low	Medium	High	Low	High	Low
Fingerprint	Medium	High	High	Medium	High	Medium	High
Hand Geometry	Medium	Medium	Medium	High	Medium	Medium	Medium
Hand Vein	Medium	Medium	Medium	Medium	Medium	Medium	High
Iris	High	High	High	Medium	High	Low	High
Retinal Scan	High	High	Medium	Low	High	Low	High
Signature	Low	Low	Low	High	Low	High	Low
Voice	Medium	Low	Low	Medium	Low	High	Low
DNA	High	High	High	Low	High	Low	Low
Gait	Medium	Low	Low	High	Low	High	Medium
Ear	Medium	Medium	High	Medium	Medium	High	Medium

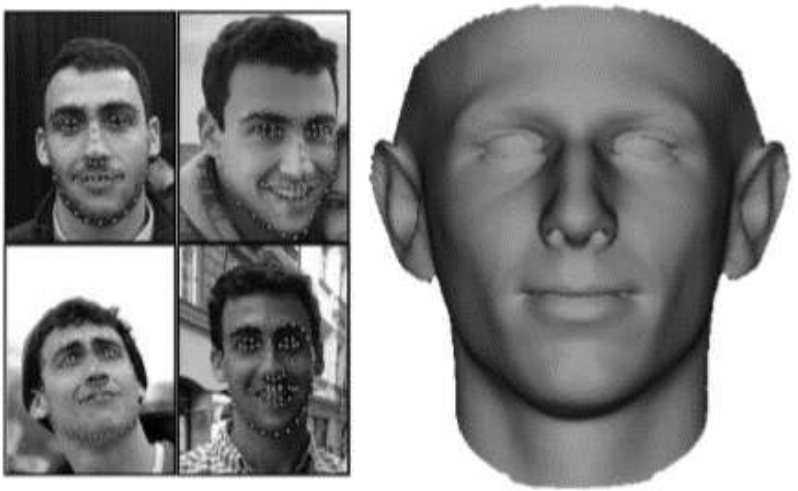


Figure 1. 3D facial model (right) built from facial landmarks extracted from 4 images (left) ("Virtual U: Defeating Face Liveness Detection by Building Virtual Models from Your Public Photos | USENIX", 2017).

Fingerprint can also be easily spoofed as the fingerprint marks can be easily obtained from a smooth surface such as glass, metal, and even the sensor itself. The fingerprint marks can be discovered using techniques such as powder and brush. Spoofing is also possible through the usage of fingerprints from the dead fingers, or recreation of the fingerprints from stolen templates. Lately, fingerprints are spoofed through the images taken by high-resolution cameras (Kulkarni and Patil, 2015). The same problem also happens to iris recognition where the system can be spoofed using presentation of a photograph or video of the genuine user without reconstruction of a proper clone (Hern, 2017).

Vein-based biometrics is a favorable method to solve these problems. Vein-based biometrics uses the vascular vein pattern inside the human body for user authentication. Since the vein resides inside the human body, it has several advantages as compared to the others. For example, vein pattern is hard to counterfeit. The skin condition does not hinder clear images from being acquired during the acquisition process. Besides, vein-based biometrics is contactless during the authentication process. Thus, there is no hygienic issue to addresses in relation to body health.

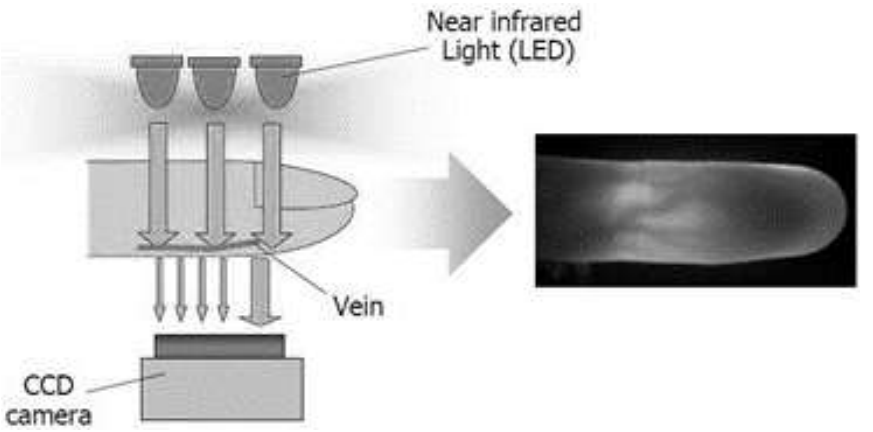


Figure 2. A Finger Vein Image Acquisition System Using Near-infrared LED and CCD Camera Developed by Hitachi.

The most common vein-based biometrics are finger vein and hand/palm vein. In order to capture vein line inside the finger/palm, a near infrared camera is required. Figure 1 shows a finger vein image acquisition device developed by Hitachi. In comparison to hand or palm vein systems, finger vein appears to have more merits because the hand or palm vein system requires a sufficiently large camera to capture the vein pattern from the whole hand (Lee, Jung, and Kim, 2011).

Finger Vein Application

There are a few companies that actively research in vein-based biometrics. A live finger vein application was developed in late 2005 in a Japanese bank. The bank started to adopt finger vein biometric solution developed by Hitachi for their automated teller machines (ATM). The system has helped to alleviate problems such as card or PIN loss and forgery. Finger vein authentication is suitable to be implemented in public space because the users do not need to touch the sensor during the authentication process. There is also no trace of biometric evidence since the sensor is contactless.

Figure 3. Finger vein ATM Machine (Patrick Collinson, 2017).

Figure 4. Finger vein Time and Attendance (T&A) Systems.

In 2014, Poland has become the first country in Europe to introduce a network of "finger vein ID" cash machines. The two thousand new ATMs are installed in bank branches and supermarkets across the country. In 2016 Qatar Commercial Bank (QCB) also implement the first finger vein system in Qatar.

The implementation of vein-based biometrics is not only limited to ATMs. There are many other finger vein biometric applications in the market. One of them is Time and Attendance (T&A) solution provided by a company called TimeTarget in Australia. The T&A system has been installed at more than 3,000 sites in Australia and New Zealand (Matsui et

al. 2012). Meanwhile, finger vein biometrics has also been implemented in healthcare industry in the US. Finger vein authentication units have been incorporated in the medical data management systems used at several hospitals in Ohio.

BIOMETRICS IN MOBILE DEVICES

Today, mobile devices such as smartphone, tablet and notebook have become an essential need for almost everyone in this modern life. Daily activities like calling a friend, buying necessities from online shops, transferring money via online banking, and storing private data are performed using mobile devices. Thus, there is a rising need for higher privacy and security protection in mobile devices.

Password has been implemented for a long time in many mobile devices. Swipe pattern has also been widely implemented in smartphones, tablets and some notebooks. The latest method to secure mobile devices is biometrics. A few biometric techniques have been implemented in smartphones lately. Face recognition, fingerprint recognition, and iris recognition are among those techniques.

Face Recognition in Mobile Devices

Face recognition is the first biometric technique to be implemented in mobile device. Many companies including Samsung, Google and Microsoft compete to develop the best face recognition application. Google has officially added the ‘Face Unlock’ system in Android 4.0 in 2011. Samsung has also implemented face recognition in Galaxy S4 and the latest Samsung Galaxy S8. Microsoft introduced the ‘Windows Hello’ system in Windows 10. In addition, numerous third party face recognition applications such as BioID (Android & iPhone), FaceVault (iPhone), and Visidon applock (Android) can also be found.

Figure 5. Face Recognition in Android 4.0.

Face recognition is convenient for mobile device users. The user only needs to capture their faces to access their mobile devices. However, face recognition is vulnerable to spoofing attack. For example, the Android 4.0 face recognition system can be fooled by a captured face picture. In order to overcome that problem, Google added liveness detection by checking blinking eyes in Android 4.1. However, the liveness detection system can still be fooled by editing the captured image with opened and closed eyes (Amadeo, 2017). The same happens to Samsung face recognition system in which their latest Galaxy S8 can also be fooled with the captured pictures (MARCIANOTECH, 2017). Microsoft with Windows Hello perhaps offers a better application since the spoofing tricks by photo, video and mask could not work against the system. Windows Hello uses dual cameras to create a virtual 3D model of the registered user's face. However, there is always a cost-benefit to added hardware. The hardware vendors need to decide whether there is enough demand from the market to add specialized components like IR cameras or structured light projectors in the device ("Virtual U: Defeating Face Liveness Detection by Building Virtual Models from Your Public Photos | USENIX", 2017).

Fingerprint Recognition in Mobile Devices

With higher demand for enhanced security in mobile devices, manufactures start to embed fingerprint scanner in their products. Fingerprint scanner has become a standard specification for the latest mobile devices. To support embedded fingerprint scanner hardware, fingerprint applications and services have been developed. Apple introduced touchID which has been implemented since iPhone 5s. Android has also introduced fingerprint API since Android 6.0. Microsoft uses the same 'windows hello' service for face recognition which is available in Windows 10.

Fingerprint recognition in mobile devices has become more and more popular due to accurate, convenient, and fast performance. Based on a recent projection (Capsule, 2017), 50% of the smartphone shipments will have a fingerprint sensor by 2019. Moreover, the implementation of fingerprint recognition is not only to unlock a mobile device but also to secure a mobile payment authorization process (Zhang, Chen, and Wei, 2017).

Figure 6. Authorizing an Electronic Transaction using Fingerprint in a Smartphone.

However, spoofing is still a serious problem for any fingerprint recognition system, including the one implemented in mobile device. In 2013, Apple's TouchID sensors was spoofed by a Geman hacker named Jan Kissler within 24 hours of the release of iPhone 5S. Some researchers from Michigan State University also managed to spoof two android phones, Samsung Galaxy S6 and Huawei Honor 7 (Cao and Jain, 2016). They showed that creating a fake fingerprint is not as complex as those shown in movie. Apart from that, Jan Krissler has successfully recreated the fingerprints of the German Minister of Defence, Ursula von der Leyen. Krissler demonstrated that he used a photo of Von der Leyen's thumb taken at a distance of 3 meters away with a 200er-Objektiv lens at a news conference in October. This photo was combined with several other photos taken from other angles to recreate Von der Leyen's fingerprint. Krissler used a commercial software product called VeriFinger to synthesize these photos into a fully imaged fingerprint, which was then used to fool biometric security devices like Apple's Touch ID fingerprint scanner found in iPhone 5S, iPhone 6 and 6 Plus, iPad Air 2 and iPad Mini 3. Krissler claimed that other image-processing software aside from VeriFinger could be used as well. Anil Jain and Kai Cao created fake fingerprints by using a normal inkjet printer and a conductive silver ink and a type of photo paper manufactured by a Japanese manufacturer called AgIC. They only used a Brother printer that costs about $400 on Amazon to fool the fingerprint system (Cao and Jain, 2016). These attacks show the vulnerability of fingerprint system in mobile devices.

Iris Recognition in Mobile Devices

Some mobile device manufactures have implemented iris recognition. The mobile-based iris recognition systems available in market include Samsung Galaxy Tab, Samsung Galaxy S8, Microsoft 950 XL, and Vivo X5Pro. Iris recognition also aims to raise the security level in mobile devices.

Figure 7. User Unlocks Samsung Galaxy S8 using Iris Recognition.

The same German hacker who defeated Apple touch ID has managed to defeat the iris recognition in Samsung Galaxy S8 using only a printed eye image. The German hacker even said that "The security risk to the user from iris recognition is even bigger than with fingerprints, as we expose our irises a lot" (Hern, 2017).

FINGER VEIN BIOMETRIC FOR FUTURE MOBILE AUTHENTICATION SYSTEMS

The three biometric techniques in mobile devices discussed above face similar problems in which they are not robust against a spoofing attack. An attacker is able to recreate face, fingerprint and iris simply with taking pictures of the biometrics without knowledge of the authorized user. This is all because the three biometrics are exposed to the public externally.

Finger vein biometrics is an attractive solution to this problem. The finger vein feature is located inside the human body and cannot be seen with naked eyes. Finger vein can only be observed with near infrared camera as hemoglobins contained in the blood are able to absorb infrared light. Moreover, finger vein recognition has some favourable

characteristics as compared to the other biometric characteristic such as immunity to counterfeit and active liveness (Lian, Rui, and Chengbo, 2008; Wen and Liang, 2010). Finger vein system is also user-friendly as finger vein scanning is performed in almost the same manner as fingerprint scanning.

To implement vein recognition in mobile device, a small and portable sensor to capture the finger vein is required. Hitachi and Fujitsu have been actively doing research in finger vein to develop a small sensor to capture finger vein. Figure 8 and Figure 9 show the sensors developed by Hitachi and Fujistu (Hitachi, 2017; Fujitsu Laboratories Ltd., 2017).

Figure 8. Finger vein sensor developed by Hitachi. (Hitachi., 2017)

Figure 9. Palm vein Sensor Developed by Fujistu. (Fujitsu Laboratories Ltd., 2017)

With the availability of small finger vein sensors, a higher security level can be achieved in mobile devices. Finger vein biometrics has everything required to be the next biometric solution for mobile devices. In few years' time, mobile devices with embedded finger vein scanner will be a leading biometric solution in the market.

CONCLUSION

Biometrics systems are in place to overcome the limitations of traditional authentication methods such as passwords and tokens. Biometrics offer high universality, distinctiveness, permanence, collectability, performance, acceptability, and circumvention characteristics. Among all the available biometrics, the face, irises and fingerprints offer high reliability, low-cost and usability. However, face and iris recognition systems are highly vulnerable to spoof attack. Face and iris recognition systems can be easily fooled by high quality photographs or videos without the need for proper clone reconstruction. The fingerprint system is vulnerable to latent fingerprint marks left on glass or metal. It is also possible to fool a fingerprint system through the usage of the fingerprints on dead fingers or fingerprint recreation from stolen templates.

Vein-based biometric systems have been proposed to overcome the limitations of the face, iris, and fingerprint systems. Since vein-based biometrics uses a vascular vein pattern inside the human body for authentication, the vein feature is not prone to counterfeiting or spoofing. Besides, the skin condition does not hinder clear images from being taken, and the vein-based systems are touchless. Finger vein biometrics has been successfully applied in many areas such as banking transactions, ATMs, and attendance systems. There are many opportunities to implement a vein-based system in mobile devices in the near future.

REFERENCES

Amadeo, R. (2017). *"Galaxy S8 Face Recognition Already Defeated with a Simple Picture | Ars Technica."* Accessed July 19. https://arstechnica.com/gadgets/2017/03/video-shows-galaxy-s8-face-recognition-can-be-defeated-with-a-picture/.

Biggio, B., Z. Akhtar, G. Fumera, G. L. Marcialis, and F. Roli. (2012). "Security Evaluation of Biometric Authentication Systems under Real Spoofing Attacks." *IET Biometrics* 1 (1): 11–24. doi:10.1049/iet-bmt.2011.0012.

Cao, K., and A. K. Jain. (2016). "Hacking Mobile Phones Using 2D Printed Fingerprints." *MSU-CSE-16-2. MSU Technical Report.* http://www.passwordresearch.com/papers/paper484.html.

Capsule, R. (2017). *"Fingerprint Sensors Market in Smart Mobile Devices 2012-2019."* Accessed July 19. https://www.marketresearch.com/land/product.asp?productid=8918844&progid=87404.

Fujitsu Laboratories Ltd. (2017). *"Fujitsu Develops World's Smallest and Slimmest Palm Vein Biometric Authentication Sensor Deployable in Tablet Devices - Fujitsu Global."* Accessed July 19. http://www.fujitsu.com/global/about/resources/news/press-releases/2012/0501-01.html.

Hern, Alex. (2017). *The Guardian*. Accessed July 14. http://www.theguardian.com/profile/alex-hern.

Hitachi. (2017). *"Hitachi Develops a 3mm Thin-Type Finger Vein Authentication Module."* Accessed July 19. https://phys.org/news/2009-09-hitachi-3mm-thin-type-finger-vein.html.

Jain, A. K., R. Bolle, and S. Pankanti. (1999). *Biometrics - Personal Identification in Networked Society*. Kluwer Academic Publisher. http://www.springer.com/in/book/9780387285399.

Jain, A. K., A. Ross, and S. Pankanti. (2006). "Biometrics: A Tool for Information Security." *IEEE Transactions on Information Forensics and Security* 1 (2): 125–43. doi:10.1109/TIFS.2006.873653.

Jain, A. K., A. Ross, and S. Prabhakar. (2004). "An Introduction to Biometric Recognition." *Circuits and Systems for Video Technology, IEEE Transactions On* 14 (1): 4–20. doi:10.1109/TCSVT.2003.818349.

Kulkarni, Samruddhi S., and Hemprasad Y. Patil. (2015). "Survey on Fingerprint Spoofing, Detection Techniques and Databases." *IJCA Proceedings on National Conference on Advances in Computing* NCAC 2015 (7): 30–33.

Lee, Eui Chul, Hyunwoo Jung, and Daeyeoul Kim. (2011). "New Finger Biometric Method Using Near Infrared Imaging." *Sensors (Basel, Switzerland)* 11 (3): 2319–33. doi:10.3390/s110302319.

Lian, Z., Z. Rui, and Y. Chengbo. (2008). "Study on the Identity Authentication System on Finger Vein." In *2008 2nd International Conference on Bioinformatics and Biomedical Engineering*, 1905–7. doi:10.1109/ICBBE.2008.805.

Marcianotech. (2017). *Galaxy S8 Bloqueo Facial Puede Ser Burlado Con Foto ! [Galaxy S8 Facial Lock Can Be Mocked With Photo!]* Accessed July 19. https://www.youtube.com/ watch?v=S3rCOZNqYq0& feature=youtu.be.

Matsui, Y., A. Sawada, S. Kaneko, Y. Nakamaru, R. Ahluwalia, and D. Kumar. (2012). *"Global Deployment of Finger Vein Authentication." Hitachi Review*.

Patrick Collinson. 2017. "Forget fingerprints – banks are starting to use vein patterns for ATMs." *The Guardian*. Accessed July 14. https://www.theguardian.com/money/2014/may/14/fingerprints-vein-pattern-scan-atm.

"Virtual U: Defeating Face Liveness Detection by Building Virtual Models from *Your Public Photos | USENIX."* (2017). Accessed July 14. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/xu.

Wen, X. B., and X. Z. Liang. (2010). "Research on Enhancing Human Finger Vein Pattern Characteristics Based on Adjacent Node Threshold Image Method." In *2010 Fifth International Conference on*

Frontier of Computer Science and Technology, 552–56. doi:10.1109/FCST.2010.84.

Zhang, Y., Z. Chen, and T. Wei. (2017). "Fingerprints on Mobile Devices: Abusing and Leaking." *Blackhat*.

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 8

ANDROID DEVICE MISPLACEMENT REMEDY VIA BLUETOOTH-ENABLED TECHNOLOGY

Siew-Chin Chong and Kaven Raj S/O Manoharan***
Faculty of Information Science and Technology,
Multimedia University, Melaka, Malaysia

ABSTRACT

Mobile devices have become a part of our lifestyle in today's modern era. As we are heavily dependent on electronic mobile devices, misplacement of our gadgets can cause a lot of inconveniences and can be a risk to our personal privacy. The inconveniences caused and security concern are huge enough to cause the rise of solutions to handle misplacement of devices, specifically smartphones. This solution is a self-help mobile application which is compatible with Android operating systems of smartphones. It leverages on existing Bluetooth technology through the pairing of devices between the phone and another device (anchor device). This app not only has a function to track the location of a lost device and to lock it remotely, if necessary, it has the benefits of incorporating preventative functions such as alarm and SMS notification when a certain distance between the paired devices exceeds a safety limit, which can be preset.

* Corresponding Author Email: chong.siew.chin@mmu.edu.my.

Keywords: Android, Bluetooth, mobile devices, pairing, misplacement

INTRODUCTION

The advancement of technology in mobile devices has brought a great impact to our daily lifestyle. Everyday new mobile applications are developed to serve the fast-growing demands from the users. Smartphone has become a companion to the users and can be functioned as a personal computer. Recently, iOS - or Android - operated smartphones are dominated the mobile phones market sales (A. Livingston., 2004). In addition, more than 90 percent of Bluetooth enabled smartphones are expected to be Smart Ready devices by the Year of 2018 (Skyrocketing Demand for Bluetooth Appcessories for Latest Phones. 2017). There are billions of Bluetooth products on the market that work with the smartphones to collect information to make our lives better and more productive. On the other hand, the satellite-based navigation system, known as Global Positioning System (GPS) has become an essential element in our daily activities. GPS helps in locating a person precisely anywhere in the world, under any weather conditions, 24 hours per day, without having to be too much technically literate. There is no subscription fee or setup charges to use GPS. According to the statistics, 900 million mobile phones that incorporated GPS were sold globally in 2012 (The Economic Benefits of GPS., 2017).

Since the smartphones are always a tool to bring along with us daily, the possibilities of losing or misplacing our smartphones are undeniably high if precaution steps are not taken efficiently. When one loses his or her smart device, the inconvenience is not merely financial but it also incurs inconvenience through loss of productivity, intellectual property through files stored in these devices, data breaches and data loss of contacts. As such, the actual value of the loss of a device is higher than the cost of the device itself. Statistics from Kensington show that up to 70 million smartphones are lost each year (with only 7 percent recovered) (Kensington's Infographics., 2017). According to the study from Lookout

Mobile Security (Lookout Mobile Security., 2017), 44% of smartphones were stolen or lost due to the owners who forgot and left the smartphones in public places, as shown in Figure 1. The survey by Lookout also shows that people are willing to pay a ton of cash to retrieve their stolen smartphones and 68% of the phone theft victims would put themselves in danger to regain their smartphones. To those victims, it is not the device itself that is so valuable but the data stored inside the smartphones.

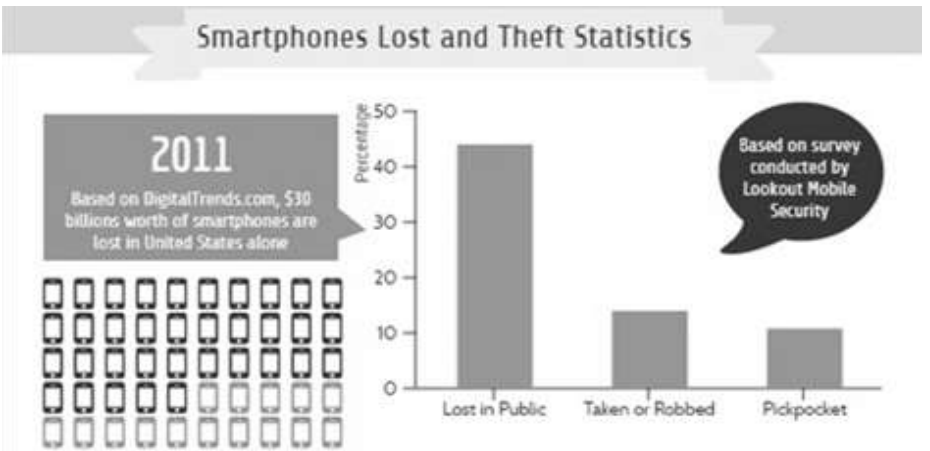


Figure 1. Smartphones Lost and Theft Statistics by Lookout Mobile Security.

Hence, various solutions are proposed to find the lost or misplaced phones. One of the solution is through the use of either web applications or mobile applications that locate the misplaced devices, eg: Life360 (Find My Phone – Life360., 2017), Google's Android Device Manager (Google Play Store, 2017a), Plan B (How-to Geek., 2016) and others. These applications normally utilize the satellite and internet connection to track down the devices remotely.

However, most of the existing applications only assists the users to track and locate the lost or misplaced smartphones without taking into consideration on any preventive measure. Thus, a better solution is to prevent the users from forgetting to bring along their smartphones in the first place rather than taking action after losing it. An ounce of prevention is worth a pound of cure. The proposed Android app, dubbed "Don't

Forget Me" is designed in a way to alert the users in the first place so that they will always remember to bring along their smartphones before their devices are misplaced. This is the preventive measure included in the proposed solution, which is different from the existing applications. Nevertheless, the proposed app is also equipped with tracking ability. The app allows user to set a Bluetooth connection with any other Bluetooth enabled device and if the Bluetooth connection is disconnected, an alarm and SMS notification are triggered to notify the users about the misplaced device. Even if the device is missing or stolen, the proposed app is able to track the GPS location of the device, as well as to lock the device remotely with password. This is to secure the device from breaching by thief. These are the possible remedies after the device is misplaced or lost. The overview of the proposed app is illustrated in Figure 2. The proposed app furnishes the before and after solutions to minimize the chances of losing the device and the confidential data in the device.

The advantageous features of the proposed app is listed as follow:

- Short Message Service (SMS) notification using SmsManager Android libray: This function enables user to enter the favourite third party's phone number and the text message that needs to be sent when triggered. During the situation where the project application detects a disconnected Bluetooth connection, this application automatically sends the SMS to the favourite third party's phone notifying him/her about the misplaced device.
- Alarm Notification using MediaPlayer Android library: This function is triggered when the application detects a disconnected Bluetooth connection. User will be notified by a loud preset ringtone in the application, even the phone is set to silent or vibrate mode. The alarm can be disabled when the user retrieves the misplaced device.
- Locating and Tracking using Global Positioning System (GPS) and Open Device Manager (ODM). The exact location of the device can be tracked through the latitude and longitude GPS coordinates. Moreover, the user can also track their lost device using the

website built for the proposed app. Firstly, user needs to register the device, and then the user can search for his/her particular device on the map of the website. The ODM enables the user to remotely lock the misplaced device with password in order to safeguard the data in the device from breaching by intruders.

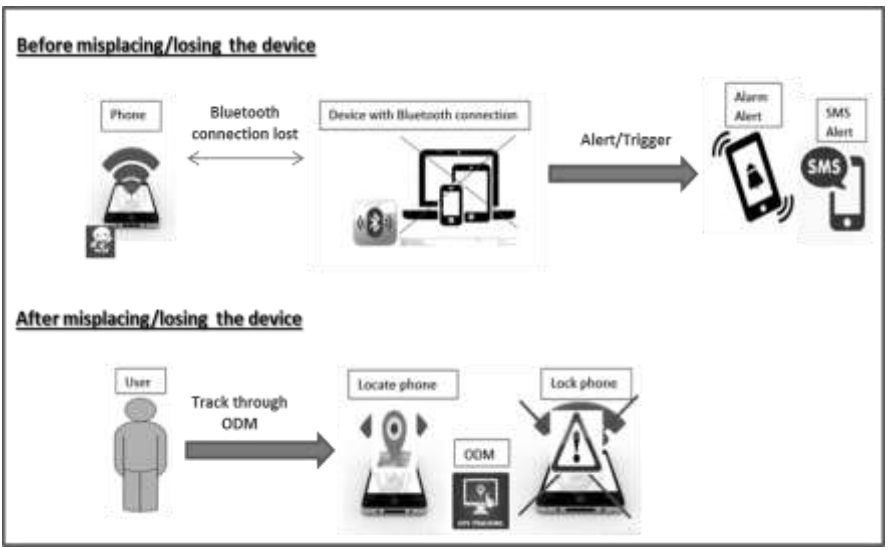


Figure 2. Overview of the Proposed Application - "Don't Forget Me."

This chapter offers an overview of the solution designed to solve the problem of the misplaced or lost Android devices. The rest of the chapter is organized as follows. In the next section, the existing mobile applications related to the scope of this chapter will be briefly reviewed. The proposed application design is discussed in the section of "Proposed System Design." Then, the implementation process of the proposed application is explained. After that, the application is tested with various testing scenario and the results are reported. Finally, the conclusion is also given at the last section.

RELATED WORKS

Smartphones are becoming ubiquitous in our culture nowadays. New technologies are being invented and implemented in smartphones in order to satisfy various customers' needs. The usefulness of smartphones have increasingly allowed users to perform more tasks efficiently in daily basis. According to a recent study in 2014, Malaysia has ten million active smartphone users, which is 140% in mobile penetration and having a lead against other country such as Indonesia, Thailand and even United States (TechCrunch, 2015).

Smartphones are so attached to our lives that we not only use them to communicate to people but also is useful to us for internet surfing, business transaction, entertainment and many more. The smartphones we carry along in our pockets all day have become more important and powerful then the most expensive and well-designed automobiles. However, the stolen or misplaced smartphone cases have been on the rise lately. Losing your precious smartphone will end you up in a miserable day. With the estimation rise of 6.1 billion smartphone users globally by the year 2020, precautious steps should be taken to reduce the number of missing smartphones.

Various mobile applications have been designed or innovated that would have improved and influenced the daily activities of human life. Android, as one of the leading mobile operating system, has encouraged the large community of developers to use the open-source code to enhance the existing applications as well to create what is currently topping in the market demand. According to Sundar Pichai, Android's Senior Vice President, Android currently has over 1 billion users and is still the dominant mobile platform (Engaget, 2014).

There are plenty of mobile applications developed on different platforms with the aim to track the location of the smart devices. However, majority of them serve solely on finding the missing phone after the device is lost. This chapter introduces and compares some of the latest well-known Android and Mac iOS applications related to the missing device solutions.

a) Plan B (Google Play Storeb, 2017): One of the most popular apps in the Google Playstore developed by Lookout Mobile Security. It uses the GPS to locate the misplaced or lost phone. This application will trigger an email to the Gmail account of the misplaced or lost phone location. User can relocate the lost phone by texting "locate" to the phone number and the latest location is again sent to the Gmail account. It locates the registered device for every ten minutes. The strength of the app is that it allows user to find their phones without installing it beforehand and it is able to send SMS to the missing phone in order to get the location. However, there is no security feature installed in it and it is vulnerable to stealing confidential information. Figure 3 shows some samples of the graphical user interfaces of Plan B.

Figure 3. Sample Graphical Interfaces of Plan B.

b) AntiDroidTheft (Google Play Storeb, 2017): This is an Android app developed by Zobo Technologies to view the position of the lost or misplaced smartphones through GPS. User needs to create an account and provide the email and password for the app. Another feature of this app is child tracking, which the activities of the child can be traced. It would be helpful in case of kidnapping by locating the location of the person. However, this app does not enable the alarm for searching purpose and the process of tracking is not user-friendly. In addition, the app would have installed to another smartphone in order to make a reverse look up. Some of the users commented that the GPS feature is somehow not very accurate. Samples of interfaces of AntiDroidTheft are shown in Figure 4.

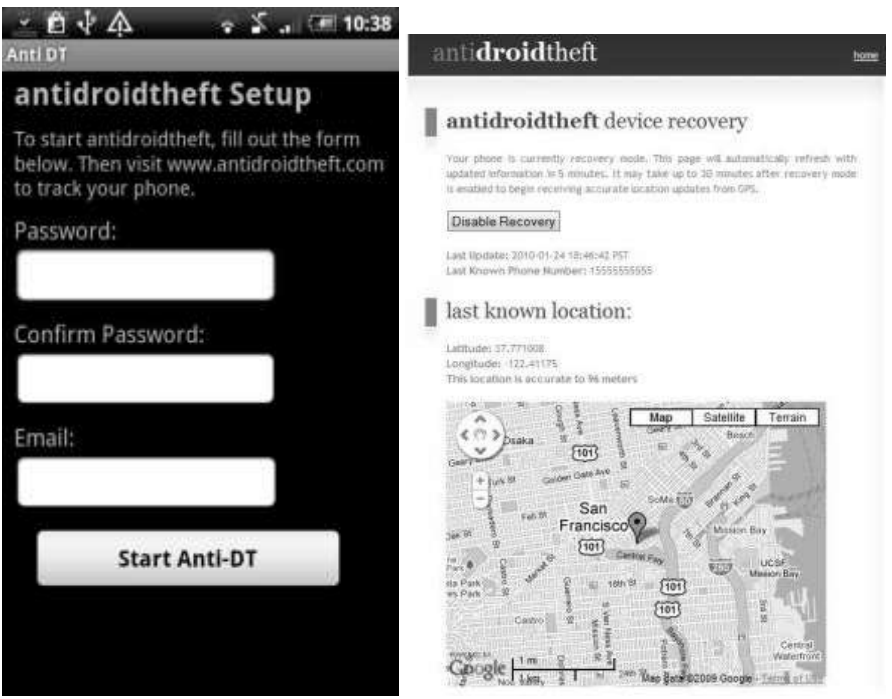


Figure 4. Sample Graphical Interfaces of AntiDroidTheft.

c) FoneHome (FoneHome, 2017): This is the mobile application built on the iOS platform by Appmoys LLC. The main feature of the app is to track the location of the device on the map. In case a thief has stolen the device, it takes the picture of the thief and the picture can be viewed remotely by the user. A loud siren is activated for searching purpose. It also can wipe the device and permanently remove the data it contains. The weaknesses of the app are the compatibility issues and the heavy consumption of the battery as it runs in the background of the device. It has the difficulty to pair with other iOS devices. Figure 5 presents some screenshots of FoneHome.

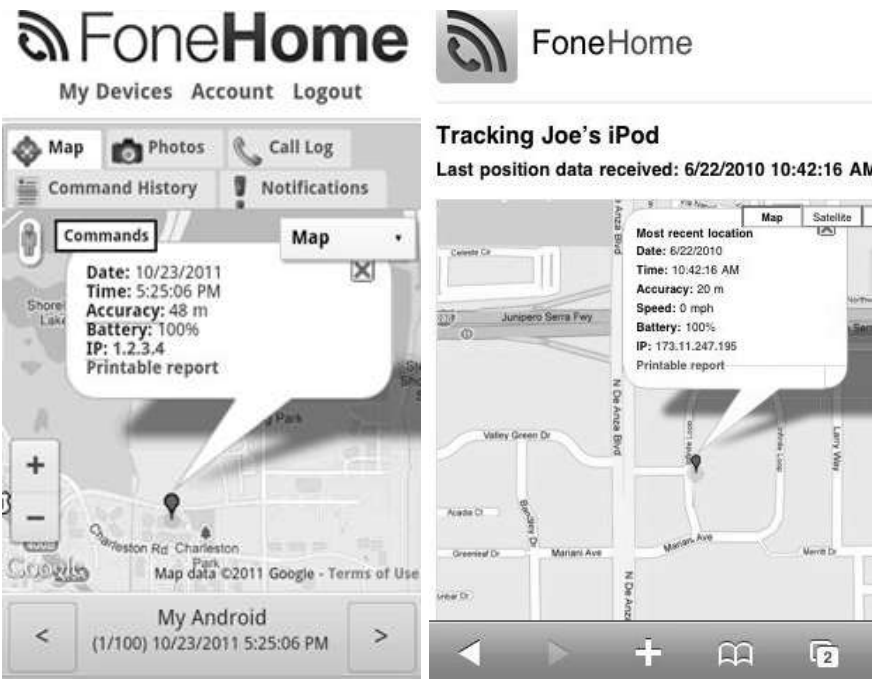


Figure 5. Sample Screenshots of FoneHome.

d) GadgetTrak (GadgetTrak, 2017): This iOS application is developed by WestinTech to allow users to find and protect the

information in the device. It tracks the device on a map and remotely locks it so that thief would not be able to steal the valuable information in the device. If the device is nearby, it would play a sound to notify the location of the device. User is able to display message showing to whom to contact in order to retrieve the missing device. In spite of that, this app requires plenty of information for tracking purpose. The sample design of GadgetTrack can be viewed in Figure 6.

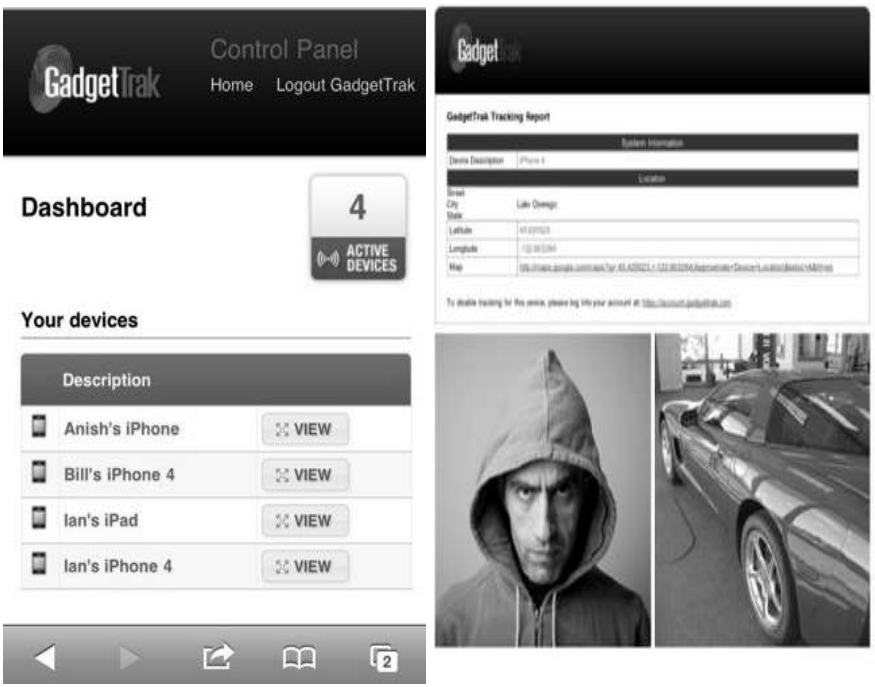


Figure 6. Sample Screenshots of GadgetTrak.

e) GPS Phone Tracker Pro (Google Play Store, 2017c): This Android application is developed by Iready and the main purpose of the app is to report the real-time whereabouts of the user's friends or family members. Through the app's website, it can assist in finding the lost devices. It accesses GPS

Android data to pinpoint the location of missing devices. Through the use of satellites, it triangulates the exact location of every phone registered to the user's account. Figure 7 illustrates some graphical user interfaces of GPS Phone Tracker Pro.

Figure 7. Sample Graphical User Interfaces of GPS Phone Tracker Pro.

f) SeekDroid (SeekDroid, 2017): This is another Android device tracker to track the lost devices developed by GT Media. It has the feature to lock the device to secure the data if it has been stolen. It locates the device via text message and set off an alarm with a custom message. It is able to control multiple devices with one account. In addition, it keeps a 30 day history of the devices' location. The only weakness of this app is the accuracy of locating. Samples of screenshots can be viewed in Figure 8.

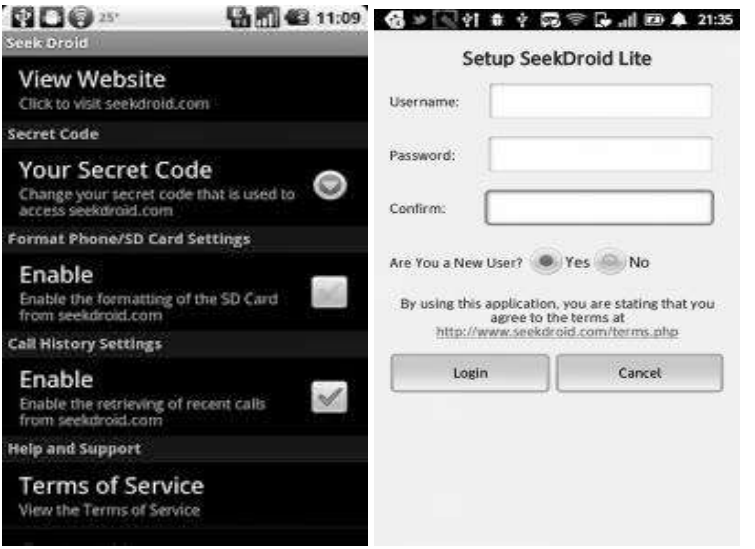


Figure 8. Sample Graphical User Interfaces of SeekDroid.

Table 1 shows the comparisons between the studied existing apps and the proposed app. Different features or strengths such as bluetooth capability, security password, navigation accuracy, GPS tracking, alarm notification, SMS notification, remotely locking feature and operating system are compared among these existing apps. Due to the free nature of Android platform, there are also a lot of other tracker apps available in the market (Google Play Store, 2017d) such as Find My Device , Where's My Droid, Lost Android and others, in order to fulfill the increasing market demand in protecting everyone's valuable device.

PROPOSED SYSTEM DESIGN

This section details on how the proposed Android application is developed and demonstrates a clear picture of the tool's functions as well as the processes in diagrammatic representations. Figure 9 illustrates the system architecture of the overall idea.

Table 1. Comparison on different features of various existing device tracking applications and the proposed solution

Features	Plan B [8]	AntiDroid Theft [12]	Fone Home [13]	Gadget Trak [14]	GPS Phone Tracker Pro [15]	Seek Droid [16]	Proposed app – Don't Forget Me
Bluetooth Capability	No	No	No	No	No	No	Yes
Navigation Accuracy	No	No	No	No	No	No	Yes
Bluetooth Capability	No	No	No	No	No	No	Yes
Navigation Accuracy	No	No	No	No	No	No	Yes
GPS Tracking	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Alarm Notification	No	No	No	No	No	Yes	Yes
SMS Notification	Yes	No	No	No	No	Yes	Yes
Security Password	No	Yes	No	No	No	No	Yes
Lock Phone Remotely	No	No	No	Yes	No	Yes	Yes
Operating System	Android	Android	iOS	iOS	Android	Android	Android

To implement the proposed application, there are two important factors of the prototype to be concerned with: the client and the server applications, which is elaborated in the next sub-section. The major components and the main algorithms are to be discussed in the rest of the sub-sections.

The proposed app – "Don't Forget Me" is designed with the following characteristics:

- Use Bluetooth pairing and connection to trigger an alarm to notify the user from forgetting the Android device.

- Able to trigger alarm although the device is in Silent mode or Vibration mode.
- Use SMS notification to send information about the ODM which can be used for tracking and locating scenarios.
- Able to track the missing or misplaced device by using GPS technology with combination of ODM.
- To secure device remotely with a password to prevent theft from breaching valuable information in the misplaced device.
- Real-time and responsive application.

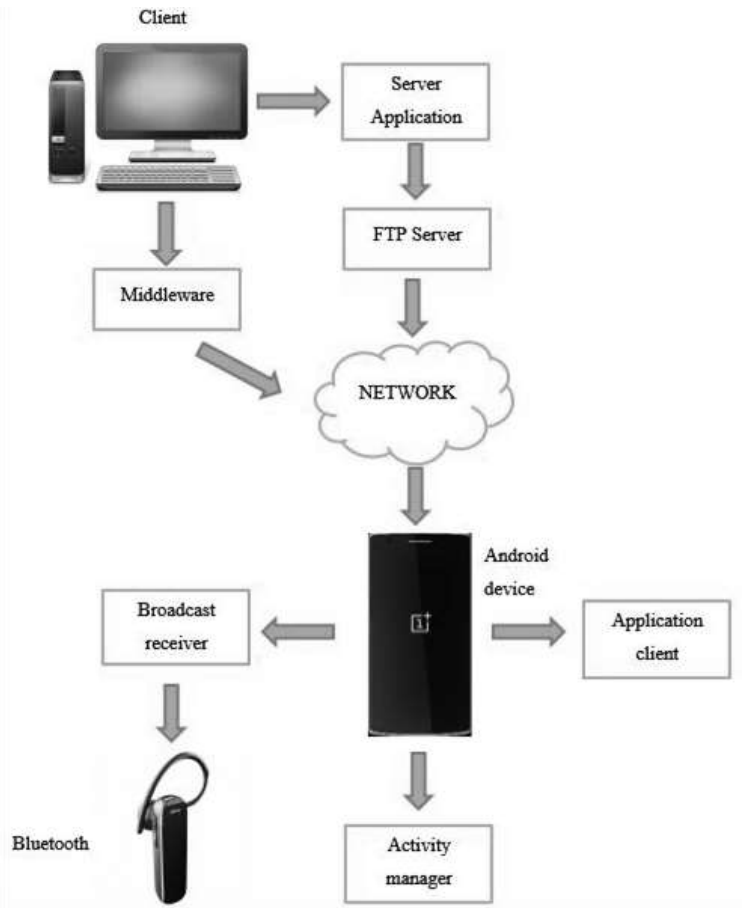


Figure 9. System Architecture of the Proposed Application.

The Client and Server Application

The client and server application flow diagram is depicted in Figure 10.

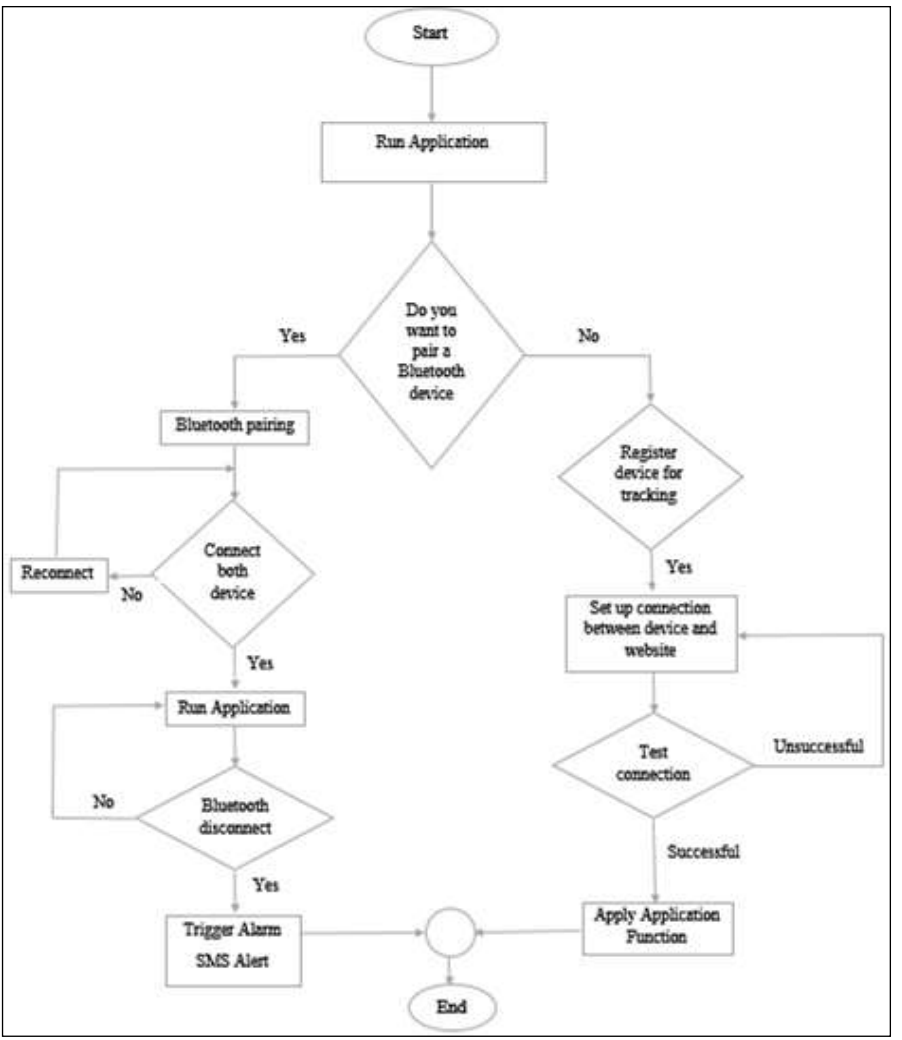


Figure 10. The Client and Server Application Flow Diagram.

The user can have the options to pair with Bluetooth or to track for lost device. If the user chooses to pair with the Bluetooth, the left option of the flow should be followed, which is the client side of the application. The user is allowed to enable the pairing, unpairing, connecting or disconnecting to a particular Bluetooth device. If the Bluetooth connection is disconnected, the application will trigger the alarm notification and send the formatted SMS to the preset third party. The third party could be any of the user's family members, friends or others, whoever is reachable to be alerted about the device misplacement.

User will be directed to the server side of the application if the user selects to search for the misplaced or missing device. Here the user is able to remotely track and lock the device with a password to fail any attempt of the theft in breaching the important information stored in the device.

Application Scenarios

In this section, a series of possible scenarios are described in accordance with the applicability of the proposed app.

Figure 11 exemplifies a scenario when a user has the phone installed with this proposed app named "Don't Forget Me" and tries to connect to another device with Bluetooth connection enabled. The proposed app allows the pairing and enables the Bluetooth connection. When the app is activated, any incoming connection request from the Bluetooth devices nearby will be listened. Once the Bluetooth request is accepted, the Bluetooth connection is established between both devices. The app will regularly check for the distance of the Bluetooth connection range. If the distance of both the devices goes out of the permitted range, the app will try to reconnect the devices. The approximate range of the Bluetooth connection is about 10 meters or 30 feet.

Figure 12 shows the situation where a connection is lost between the phone and the other Bluetooth device. The app will automatically trigger the alarm alert and the SMS alert function. The alarm alert function works by using the default alarm tone set by the user. The alarm will ring

continuously until the user acquires back the misplaced phone and disables it. The app will bypass the silent mode or vibration mode whenever the alarm alert function is triggered.

For the SMS alert function, it works with the desired third party's phone number being saved into the application beforehand. Apart from the alarm alert function as discussed, the SMS function will also be triggered after the Bluetooth connection is lost. A text message containing the details of the login information to the ODM will be sent to the third party's phone. With the required information to login to ODM, user is able to track down and lock the phone remotely.

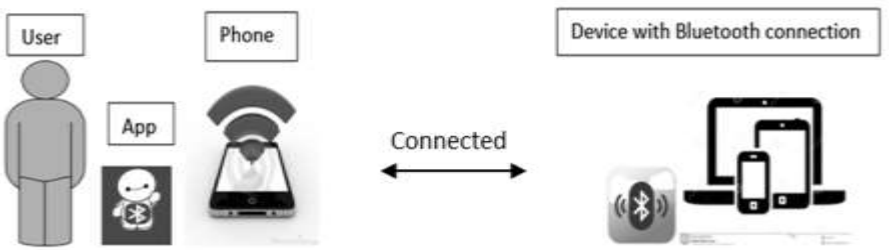


Figure 11. Enabling Connection Between Bluetooth Devices.

Figure 12. The Scenario of Bluetooth Disconnection of The Proposed Application.

Figure 13 demonstrates another scenario when the phone has been stolen or misplaced accidentally. In order to trace the phone, the ODM comes to the play. ODM is an open source application which allows the user to access to the website for locating the missing phone. It also allows the phone to be secured with a password remotely.

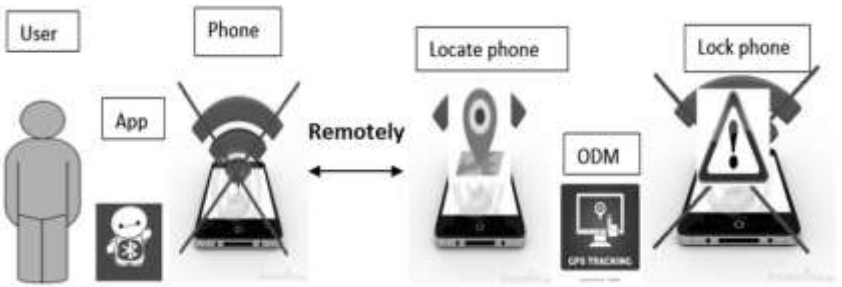


Figure 13. The Scenario of Bluetooth Disconnection of The Proposed Application.

First, user would need to register his/her phone with the ODM and enable the GPS function for tracking and prevention purpose. If the user has accidentally lost his/her phone, the ODM application can be used to provide the exact coordinate of the missing device. With the coordinate details, user is able to trace down the location of the phone easily.

ODM has another function to allow the user to remotely control the phone e.g.: lock the phone with any password set by the user. This is very much useful to avoid the intrusion of the thief towards the private and important information in the device. The thief would be asked to enter the password in order to access the phone. Besides, user can opt for another function to protect the private information in the device by wiping off all the data completely from the device.

Code Snippets of the Client Application

There are some major activities governed the client-side of the proposed application such as the alarm alert activity, SMS alert activity and device tracking activity. In this section, the codes for each major

activity will be introduced. All the code snippets are written in Java Programming language and are being run by Eclipse software.

The proposed app uses the package of BluetoothDevice and BluetoothSocket to detect for any Bluetooth connection and disconnection, as displayed in Figure 14.

```
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
```

Figure 14. The Imported Bluetooth Package.

An intent is created to start the connection and disconnection action listener. Once the application detects the connection, it will toast a message of connected. If the application detects a disconnection, the alarm and SMS will be triggered and sent automatically. Figure 15 presents the code snippet of creating intent for connection and disconnection.

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

intentFilter filter1, filter;

    filter1 = new IntentFilter(android.bluetooth.BluetoothDevice.ACTION_ACL_DISCONNECTED);

    filter2 = new IntentFilter(android.bluetooth.BluetoothDevice.ACTION_ACL_CONNECTED);

    this.registerReceiver(mConnectionReceiver, filter1);
    this.registerReceiver(mConnectionReceiver, filter2);
}
```

Figure 15. Creating Intent for Connection and Disconnection.

The proposed application uses MediaPlayer to trigger alarm if the Bluetooth function detects a disconnection. Figure 16 shows the code for implementing MediaPlayer. The default ringtone is preset and triggered by the function mediaPlayer.start(). The user can disable the alarm after the missing device has been retrieved.

```
private final BroadcastReceiver mConnectionReceiver = new BroadcastReceiver()
{
    @Override
    public void onReceive(Context context, Intent intent) {
      String s = intent.getAction();
      if (BluetoothDevice.ACTION_ACL_CONNECTED.equals(s)){
        showToast("Connected");
      }
      else if (BluetoothDevice.ACTION_ACL_DISCONNECTED.equals(s)){
        showToast("Disconnected");
        mediaPlayer.start();
        sendSMSMessage();
      }
    }
  };
```

Figure 16. Alarm Alert Activity.

```
protected void sendSMSMessage() {
    Log.i("Send SMS", "");


String phoneNo = txtphoneNo.getText().toString();
    String message = txtMessage.getText().toString();

    try {
      SmsManager smsManager = SmsManager.getDefault();
      smsManager.sendTextMessage(phoneNo, null, message, null, null);
      Toast.makeText(getApplicationContext(), "SMS sent.",
      Toast.LENGTH_LONG).show();
    }
      catch (Exception e) {
      Toast.makeText(getApplicationContext(),
      "SMS failed, please try again.",
      Toast.LENGTH_LONG).show();
      e.printStackTrace();
    }
  }
```

Figure 17. SMS Alert Activity.

The proposed app uses the SmsManager package in the Android library to handle the SMS operation in sending messages to a mobile device. Figure 17 exhibits the code for implementing SMS alert activity. In the event of the user had accidentally misplaced their device and the Bluetooth connection is disconnected, the function will automatically send the text message to the desired third party's mobile device notifying that the user has lost the phone and ask their assistance to find the misplaced phone. The function accepts both phone number string and text message string from the user and then send it to the third person.

```
public GetLocation()
   {
      gps_enabled = true;
      network_enabled = true;
      locationListenerGps = new LocationListener() {
         final GetLocation this$0;
         public void onLocationChanged(Location location)
         {
            timer1.cancel();
            locationResult.gotLocation(location);
            lm.removeUpdates(this);
            lm.removeUpdates(locationListenerNetwork);
         }
         {
            this$0 = GetLocation.this;
            super();
         }
      };
locationListenerNetwork = new LocationListener() {
         final GetLocation this$0;
         public void onLocationChanged(Location location)
         {
            timer1.cancel();
            locationResult.gotLocation(location);
            lm.removeUpdates(this);
            lm.removeUpdates(locationListenerGps);
         }
```

Figure 18. Device Tracking Activity.

In order to trace the location of the misplaced device, the proposed app utilises the packages such as LocationManager and LocationListener. User would need to register and login to the ODM website that has been set up

for locating and tracking purposes. After that, the user is able to track down to secure the device from remote site. Figure 18 presents the code snippet for device tracking activity. Here the function is extracted from the open source ODM. The application will enable the GPS to locate the misplaced device. After the location information is gathered, the tracking function can be started.

SYSTEM IMPLEMENTATION

This section explains the functionalities and the user interface of the proposed application.

Prototype Functionalities

Practically, the proposed application pairs with another Android device through the Bluetooth connection. The main objective of establishing the Bluetooth connection is to prevent the user from forgetting to bring along the device. If the Bluetooth connection between both paired devices is disconnected due to the permitted range, an alarm alert is activated to notify the user that the device was not brought along with them. In the event that the user is not aware of the alarm alert, the SMS notification will also be triggered to send the saved text message to the desired third person's phone number. The tracking and locating the missing device can be done through the ODM website.

There are three major functions that can be performed by the installed client-side application on the device such as:

- Device Bluetooth Pairing and Unpairing Activity
- Device Enable and Disable Bluetooth Connection Activity
- SMS Notification Activity

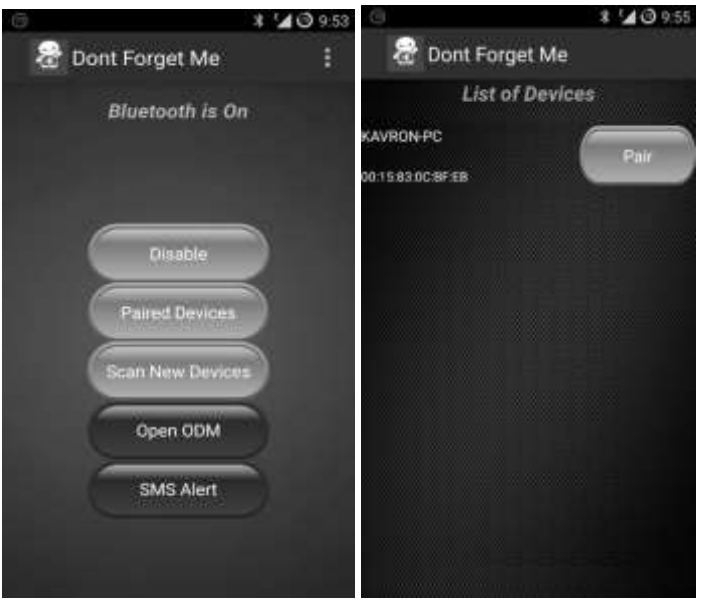
For the server-side application, the functions that are installed on the computer are:

- User has to surf the website on his/her computer.
- Locating and tracking the misplaced device.
- Securing the misplaced phone remotely with password.

Graphical User Interface

a. Make a connection to a Bluetooth device

Figure 19 shows (i) the device enable Bluetooth connection activity (ii) device Bluetooth pairing activity. Click on the "Enable" button to enable Bluetooth in the device. Once the Bluetooth is enabled, there is a text on the top of the screen stating "Bluetooth is On."



(i) (ii)

Figure 19. (i) Enable Bluetooth Connection Activity, (ii) Bluetooth Pairing Activity.

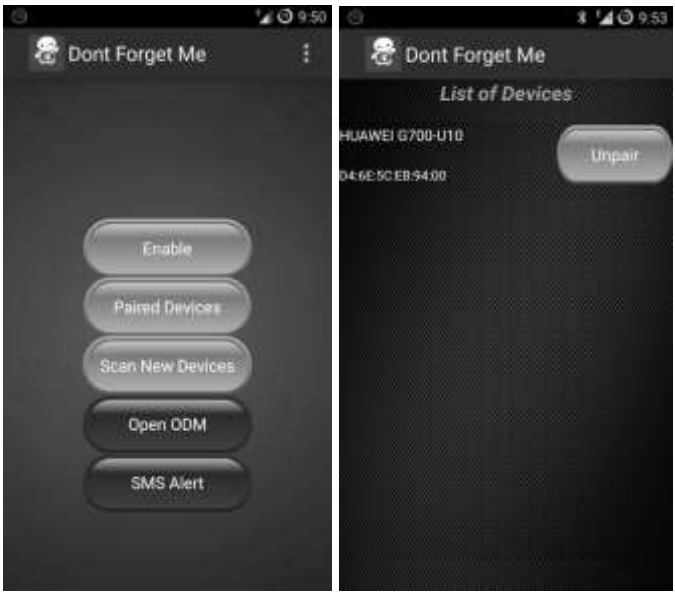
To pair with the other device, click on the “Paired Devices” button to view the already paired Bluetooth devices with the current device. User needs to click on the desired Bluetooth device to establish a Bluetooth connection between both devices. In order to scan for new Bluetooth device to be paired with, click on the “Scan New Devices” button. The available Bluetooth devices in range will be listed. Once the application has detected the device that wanted to be paired with, click on the “Pair” button next to the device name and the pairing process will begin.

b. Turn off Bluetooth or make disconnection to a Bluetooth device

To turn off the Bluetooth function in the device, click on the “Disable” button at the home screen and the text “Bluetooth is Off” is displayed at the top of the screen as illustrated in Figure 20 (i). To unpair a Bluetooth device, select the particular device to be unpaired and the application will automatically terminate the connection of both devices. This can be seen in Figure 20 (ii).

c. Registering SMS details

User is allowed to enter the favourite third party’s phone number as the receiver of the SMS notification. Click on the “SMS Alert” button and user will be directed to the page to enter the phone number and text message. The phone number can be preset, for example “0123456789” and the text message can be designed for example “Don’t Forget Me!” in the blank space provided. Once user has filled in both phone number and message column, user is required to click on the “Register” button to save the information. Figure 21 displays the SMS Notification page.



(i) (ii)

Figure 20. (i) Disable Bluetooth Connection Activity, (ii) Bluetooth Unpairing Activity.

Figure 21. SMS Notification Page.

d. Registering device to ODM

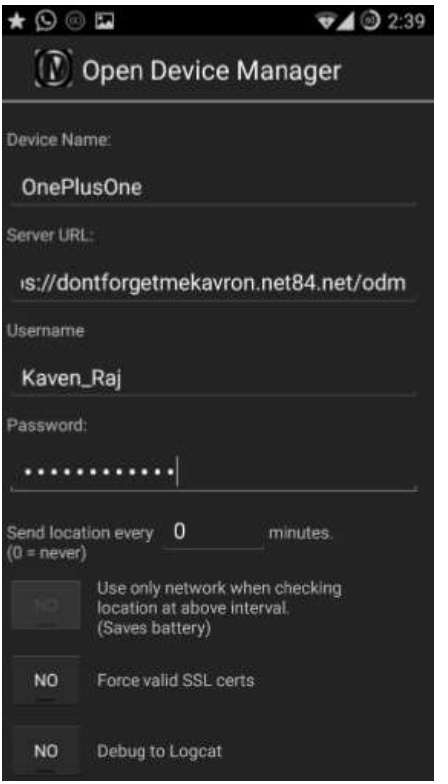


Figure 22. ODM Registration Page.

In order to register the device to the ODM website, click on the "Open ODM" button and the user will be directed to a new screen enabling the user to enter the details. After that, click on the "Save Setting" button and the device is ready for tracking purpose. The screenshot of ODM is shown in Figure 22.

e. Locating device using ODM

Locating, tracking and securing the device can be done remotely through the ODM website, as displayed in Figure 23. User is able to get the longitude and latitude coordinate of the misplaced device for tracking

purposes. User is also able to lock the device remotely to protect the leakage of the sensitive data to the intruder. It can also wipe off the data from the device if needed.

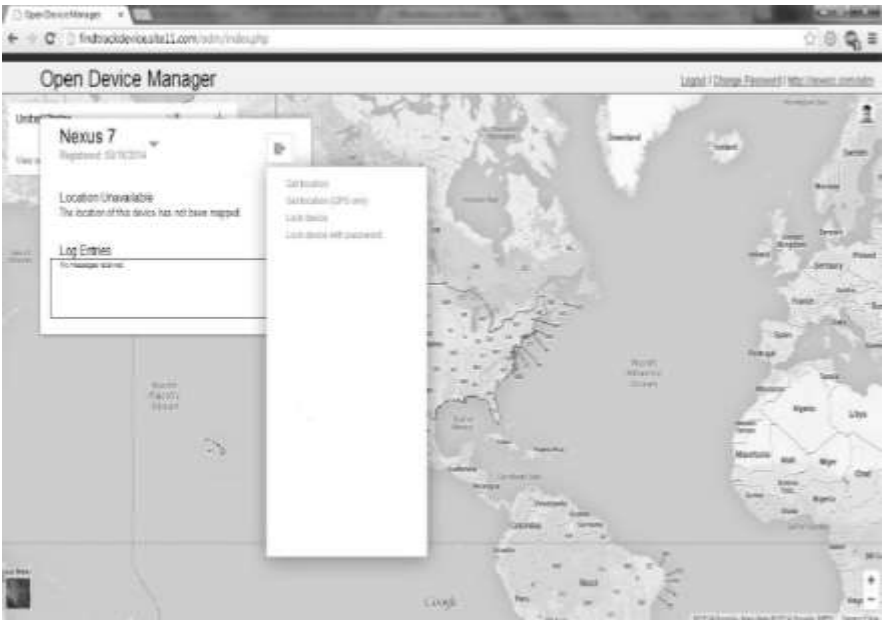


Figure 23. Tracking and Locating Feature in Open Device Manager Website.

TESTING RESULTS

Comprehensive testing is required on the application to ensure that the entire application will flow efficiently and reliably. In another words, a good application should be able to produce the desired outcome in an effective manner. In this testing process, several software tests are conducted in different types of scenarios. Each scenario should yield an expected output. If there are any unexpected errors and bugs, they will be repaired accordingly.

Unit test is conducted on individual modules to assess the fitness of the application towards the intended purpose. After the individual modules

pass the unit test, the individual modules are combined and sent for integration test. Integration test is performed to evaluate the performance, functions and reliability of the application. Results of the test cases are demonstrated in Table 2. The result shows that the system is performing well and is reliable as all of the outcome is as expected.

Table 2. Test Cases of "Don't Forget Me"

Test Case	Expected Output	Result
Turn On Bluetooth	Bluetooth enabled	Pass
Turn Off Bluetooth	Bluetooth disabled	Pass
Scan For New Device	Found new devices	Pass
Unable To Find New Device	No Device is listed	Pass
Pairing New Device	Device is paired successfully	Pass
Unable To Pair New Device	Pairing failed	Pass
Unpairing With Paired Device	Devices are unpaired	Pass
Unable To Unpairing Paired Device	Device are failed to unpair	Pass
Connect To Device	Device is connected	Pass
Unable To Connect To Device	Device is not connected	Pass
Disconnecting To Device	Device is disconnected	Pass
Unable To Disconnect With Device	Device is not disconnected	Pass
Register SMS Number and Message	Third party's SMS number and message are registered	Pass
Unable To Register SMS Number and Message	SMS number and message are not registered	Pass
Entering User SMS Message	SMS Message is saved	Pass
Unable to Enter User SMS Message	SMS Message is not saved	Pass
Entering User SMS Phone Number	Phone number is saved	Pass
Unable to Enter User SMS Phone Number	Phone number is not saved	Pass
Send User SMS Notification	SMS Notification is received	Pass
Unable To Send User SMS Notification	SMS Notification is not received	Pass
User Alarm Notification	Phone rings loudly	Pass
Turn Off Alarm Notification	Phone does not ring	Pass

Test Case	Expected Output	Result
User Alarm Notification in Phone's Silent Mode	Phone rings loudly	Pass
User Alarm Notification in Phone's Vibrant Mode	Phone rings loudly	Pass
The Distance of Two Paired Devices is in the Permitted Bluetooth Range	Phone does not ring	Pass
The Distance of Two Paired Devices is out of the Permitted Bluetooth Range	Phone rings loudly	Pass
Register Device With ODM	Device is registered and is able to login	Pass
Unable To Register Device With ODM	Device is not registered and fail to login	Pass
Track Device With ODM	Device's location tracked	Pass
Unable To Track Device With ODM	No information on device	Pass
Wipe Off with ODM	Device is reset to default setting and memory is cleared.	Pass
Lock the Phone with ODM	New password is saved and the phone is locked	Pass

Additionally, a survey has been conducted on the public to collect the user feedback towards the usefulness and stability of the proposed application. 200 potential users were randomly targeted for the survey. Among them, 182 participants had provided their feedback. Most of the participants own a/more than one Android devices, including smartphone, tablet and pad. 92.3% of them have the experience in downloading Android apps frequently. This reflects that the Android apps are getting popular among the smart device users.

The survey requires the participants to have a try on the proposed application. Almost 94.5% of the participants feel comfortable with the proposed idea and 163 of them would like to keep this application in their device for prevention purpose. 77% of the participants are agreed that the overall performance of the application is satisfactory. 10% of the 182 participants would like to have the automatic video recording feature when

the alarm of the application is triggered and 50% of them would like to be informed on the updates of the proposed application.

CONCLUSION

Seeking the optimum prevention solution of missing one's belongings is always important to everyone. Smart devices especially smartphones have now become an essential tool that manage our lifestyle. This "Don't Forget Me" Android application fulfills the needs to prevent, detect and secure the Android devices from being stole or misplaced. This application not only provides the prevention before losing or misplacing the device, but provide also the solution after losing the device to the Android users. This invention is based on two technologies: a Bluetooth system and a GPS. The Bluetooth technology has the role of establishing a connection with another Bluetooth device. For the latter, it is used to track down the location of the smart device and also remotely secure the device with new password via the assistance of ODM website.

Practically, this proposed application pairs with another device using Bluetooth connection. This Bluetooth connection is established in order to prevent the user from forgetting to bring along the device. If the Bluetooth connection between the paired devices is disconnected, an alarm is triggered to alert the user that the device was not brought along with them. In addition, a SMS with the ODM's URL is automatically sent to a preset third person to enable them for tracking purpose. User will also be allowed to track and locate the missing device via the ODM website. User is allowed to remotely lock the misplaced device with password in order to secure the device from breaching by strangers. Compared to most of the existing applications in the market which only track devices after the devices are gone missing, this proposed application considers both the remedies for the users before and after the device is lost.

The proposed application can be improved in the future to achieve a bigger market area by making it compatibility to other operating systems such as iOS and Window devices. Furthermore, the application can be

upgraded to be able to track and locate within the device without requiring to set up a webhosting server. Also, a system can be used to replace the auto SMS notification such as auto email notification function. As suggested by the participants of the survey, an auto video recording feature can be implemented in order to capture the person who intentionally or unintentionally grab the misplaced device.

REFERENCES

Engaget. 2014. *"Android still the dominant mobile OS with 1 billion active users."* Available at: https://www.engadget.com/2014/06/25/google-io-2014-by-the-numbers/ (Accessed 24 May 2017).

Find My Phone – Life360. 2017. Available at: https://www.life360.com/find-my-phone/ (Accessed 5 May 2017).

FoneHome. 2017. Available at: https://www.myfonehome.com/ (Accessed 23 May 2017).

GadgetTrak. 2017. *"GadgetTrak for iOS Security."* Available at: http://www.gadgettrak.com/ (Accessed 22 May 2017).

Google Pay Store. 2017. *"Where's My Droid."* Available at: https://play.google.com/store/apps/details?id=com.alienmanfc6.wheresmyandroid&hl=en (Accessed 23 May 2017).

Google Play Store. 2017. *"Lost Android."* Available at: https://play.google.com/store/apps/details?id=com.androidlost&hl=en (Accessed 23 May 2017).

Google Play Store. 2017a. *"Plan B – Track Lost Phone."* Available: https://play.google.com/store/apps/details?id=com.planbnew&hl=en (Accessed 24 May 2017).

Google Play Store. 2017b. *"GPS Phone Tracker Pro."* Available at: https://play.google.com/store/apps/details?id=com.fsp.android.c&hl=en (Accessed 23 May 2017).

Google Play Store. 2017c. *"Find My Device."* Available at: https://play.google.com/store/apps/details?id=com.google.android.apps.adm&hl=en (Accessed 23 May 2017).

Google Play Store. 2017d. *"AntiDroidTheft."* Available at: https://play.google.com/store/apps/details?id=com.android.antidroidtheft&hl=en (Accessed 24 May 2017).

How-to Geek. 2016. *"How to Find Your Lost or Stolen Android Phone."* Available at: https://www.howtogeek.com/170276/how-to-locate-your-lost-or-stolen-android-phone-and-wipe-if-necessary/ (Accessed 5 May 2017).

Kensington's Infographics. 2017. Available at: https://www.kensington.com/us/us/4596/infographics (Accessed 5 May 2017).

Livingston, 2004, "Smartphones and other Mobile Devices: the Swiss Army Knives of the 21st Century". *Educase Quarterly (EQ)*, 27(2) :46-52

Lookout Mobile Security, 2017, Available at: https://www.lookout.com/insights (Accessed 5 May 2017).

SeekDroid. 2017. Available at: https://www.seekdroid.com/ (Accessed 22 May 2017).

Skyrocketing Demand for Bluetooth Appcessories for Latest Phones. 2017. Available at: https://www.bluetooth.com/what-is-bluetooth-technology/where-to-find-it/mobile-phones-smart-phones. (Accessed 4 May 2017).

TechCrunch. 2015. *"6.1B Smartphone Users Globally By 2020, Overtaking Basic Fixed Phone Subscriptions."* Available at: https://techcrunch.com/2015/06/02/6-1b-smartphone-users-globally-by-2020-overtaking-basic-fixed-phone-subscriptions/ (Accessed 23 May 2017).

The Economic Benefits of GPS. 2017. Available at: http://gpsworld.com/the-economic-benefits-of-gps/ (Accessed 5 May 2017).

YAHOO! News. 2014. *"With 140% mobile penetration, Malaysia has 10m smartphone users."* Available at: https://sg.news.yahoo.com/140-mobile-penetration-malaysia-10m-smartphone-users-084900024.html. (Accessed 23 May 2017).

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 9

A LABELED NETWORK-BASED ANOMALY INTRUSION DETECTION SYSTEM (IDS) DATASET

Nicholas Ming Ze Lee, Shih Yin Ooi*, Yong Kian Lee and Ying Han Pang
Faculty of Information Science and Technology, Multimedia University, Melaka, Malaysia

ABSTRACT

Over the past decades, many researches were devoted to the field of network intrusion detection system (IDS) through the machine learning approach. One of the classical datasets being used over the years is DARPA 1999 dataset. However, its obsolete traffic logs are disputable since they may not be able to well representing the current network attacks. A set of labeled IDS dataset is a necessity when machine learning approach is in place. As such, this chapter presented a new IDS dataset with two different kinds, (i) Type I – avail with raw packet features, and (ii) Type II – avail with network connection features. Type I presents a

* Corresponding author, Email: syooi@mmu.edu.my.

data corpus of 8,201,274 raw packets, whereas Type II consists of 694,461 connection records. This chapter describes the network IDS data corpus, the way of collecting, and the categories of network attacks. Alongside of these, a baseline performance with J48 decision tree (Weka package) is also presented.

Keywords: network-based intrusion detection system, anomaly detection, dataset collection, classification

INTRODUCTION

There has been much interest devoted to the field of computer network intrusion detection from the machine learning community. A set of well-labeled intrusion detection system (IDS) dataset is necessity, especially to evaluate the performances of specific classifiers in detecting anomalies from the observed network traffic logs. IDS dataset is a collection of live network traffics, and should captured both normal as well as malicious (anomaly) network traffics. However, the logs may expose the organizations' network architectures if they are not handled properly, thus, not many IDS datasets are made available publicly. Furthermore, some of them are heavily anonymized and lacking of statistical characteristics, which will make them incompetent to reflect the current trends (Shiravi, 2012).

One of the most widely used benchmark IDS datasets throughout these years is KDD'99 dataset (Creech & Hu, 2013; Tan et al., 2014; Kaskar et al., 2014; Ooi et al., 2014; Aggarwal & Sharma, 2015; Moustafa & Slay, 2015; Aissa & Guerroumi, 2015; Folino & Pisani, 2016). KDD'99 dataset was re-developed from 1998 DARPA Lincoln Lab dataset, containing four categories of attacks: (i) Denial of Service attack(DoS), (ii) User to Root attack(U2R), (iii) Remote to Local attack(R2L), and (iv) Probing Attack. In view of it was collected on year 1998, many are doubting its reliability because most of the logged attacks are obsolete and not be able to reflect current network threats anymore (Ray 2013; Zuech et al., 2015). Furthermore, the attack platform of DARPA'98 was towards UNIX

machines only, and there are no attacks performed on the Windows machines (Lippmann et al., 2000). An updated version of this dataset was named as NSL-KDD dataset by Tavallaee et al. (2009), which has been released after they removed the duplicate records in KDD'99 dataset.

There are another two popular network-based IDS datasets which have been made available publicly. One of them is the Kyoto dataset. The traffic logs were captured through several in-house honeypots, during the period of November 2006 to August 2009. The honeypots were hosted in several platforms, including Windows machines (Windows XP SP2, Windows Vista) and Unix/Linux machines (MacOS X). All of the captured data was examined through Clam Antivirus, Ashula and Semantic Network Security 7160 (Song et al., 2011). In year 2012, Shiravi et al. (2012), contributed an Information Security Centre of Excellence (ISCX) 2012 IDS dataset, which contains approximately 1,512,000 packets. They were collected in the period of seven days, consisting normal traffics, internal attacks, HTTP DoS, Distributed DDoS, and brute-forced SSH.

In this chapter, we aim to contribute a network-based IDS dataset and will be made entirely free for research purpose. To our best knowledge, this collected dataset is the largest network-based IDS dataset when considering the number of non-duplicated packets, as well as the variety of attack tools. In total, the attacks are performed through the usage of seven probing tools and fifteen DoS/ DDoS tools.

DATA COLLECTION

The overall datasets are collected on four platforms, including Kali Linux, Windows XP, Windows 7 and Ubuntu. Most of the attacks are simulated from Kali Linux machine, targeting towards Windows and Ubuntu machines. For logging purposes, three open-sourced sniffers are used, which are (i) TCPDump/ WinDump, (ii) Wireshark, and (iii) TCPtrace. A general features construction flow is depicted in Figure 1.

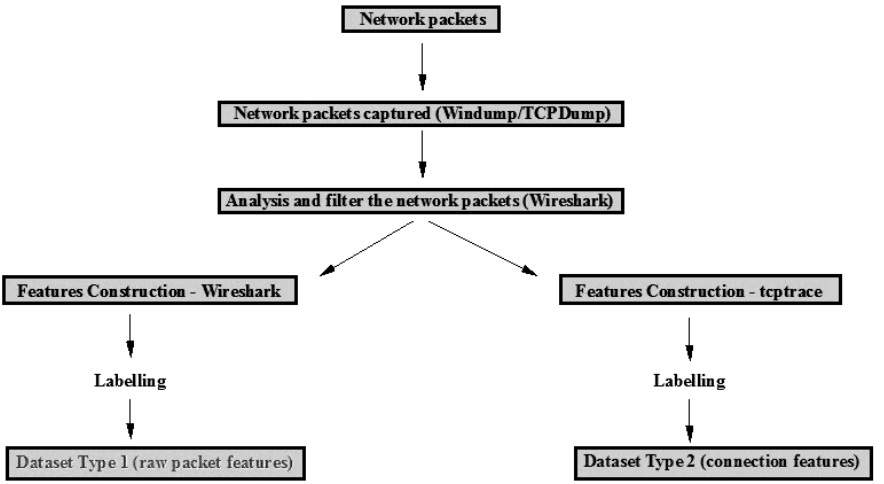


Figure 1. Overview of features construction flow.

TCPDump / WinDump

TCPDump is an open source command-line network tool which is widely used for sniffing and capturing network traffics. Its compliment version which supports Windows platforms is known as WinDump. TCPDump version 3.9.5 was installed in Ubuntu machine, whereas WinDump version 3.9.5 was installed in Windows XP and Windows 7 respectively. A snapshot of WinDump command line is shown in Figure 2.

Wireshark

Wireshark is one of the most powerful network protocol analyzers, and available freely as an open source. It supports multiple platforms including Windows, Linux, OS X and so forth. Wireshark provides several useful features such as live capture and offline analysis, flexible display filter and able to process various file formats including .pcap, .cap, .dmp and etc. It is able to output the result in several file extensions, such as XML,

PostScript, CSV and plain text format. In this work, Wireshark was used to analyze the useful network packers and filter the duplicated and unwanted network packets. A snapshot of Wireshark interface is shown in Figure 3.

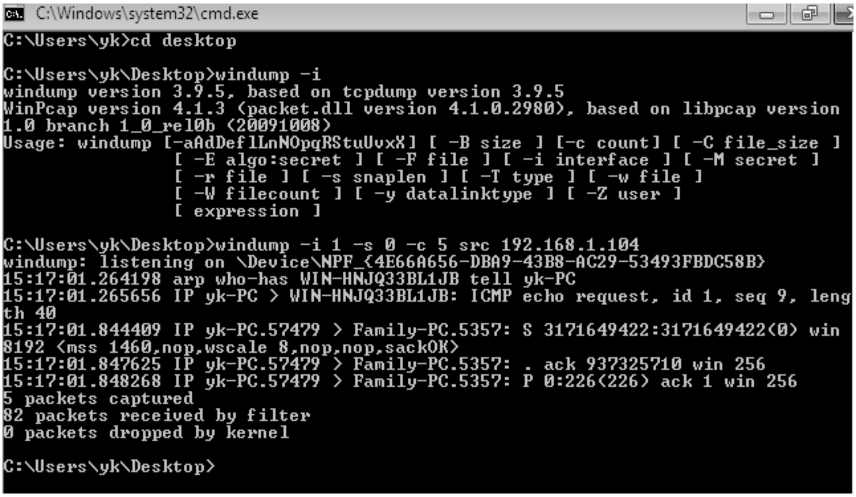


Figure 2. Snapshot of Windump in command utility.

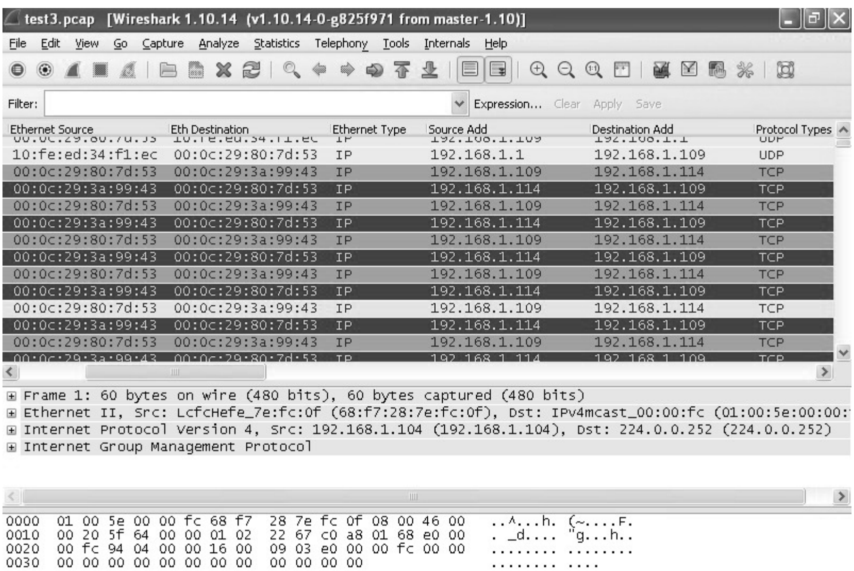


Figure 3. Snapshot of Wireshark interface.

Tcptrace

Tcptrace tool is a utility which is able to provide analysis to the TCP dump files. It can process the dump files such as those from TCPDump and WinDump. Tcptrace was installed in out Kali Linux machine. It is mainly used to generate the connection features, such as bytes and segments sent, retransmissions, round trip times, window advertisements, throughput, and so forth. The interface of tcptrace is shown in Figure 4.

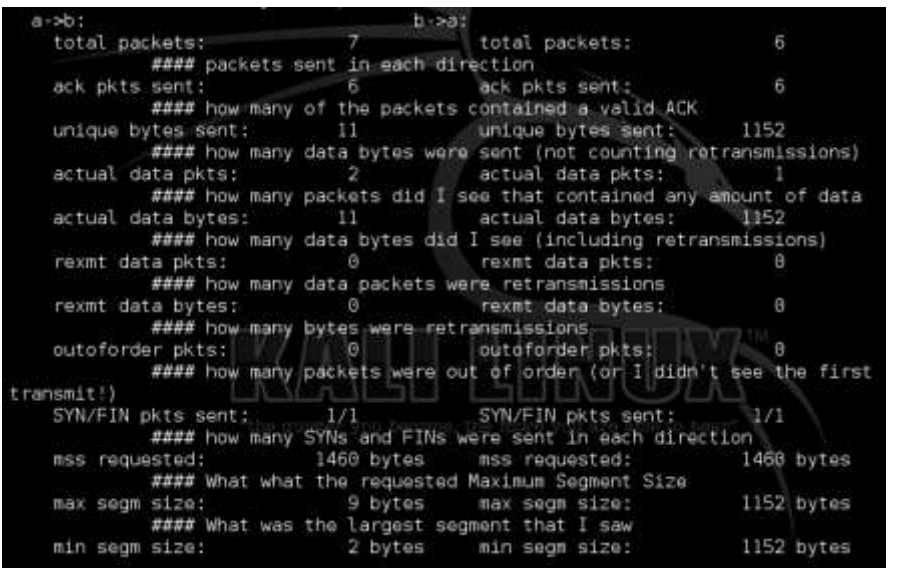


Figure 4. Snapshot of tcptrace utility.

Data Collected

Two types of dataset are collected: Dataset Type I – which contains 8,201,274 raw packet features (with no duplicated data). All of these features can directly extracted from network packet headers. For example, source address, destination address, source port and destination port. Wireshark is adopted to construct these features. The full list of features is shown in Table 1.

Table 1. Features of Dataset Type I

No	Features	Description
1	Packet Length	The size of the packet as seen "on the wire."
2	Arrival Time	(hh:mm:ss:fraction) packets arrived.
3	Ethernet Source	Source MAC address.
4	Ethernet Destination	Destination MAC address.
5	Ethernet Type	The Ethernet type: IP.
6	IP Version	First header of the IP packet. IPv4 = 4
7	IP Header Length	Internet header length. (Min: 20bytes, Max: 60bytes)
8	Total Length	Entire packet size.
9	ID	This number is for packet fragmentation purpose.
10	Time to live	TTL used to limit the packets lifetime.
11	Protocol Types	Protocol types: TCP, UDP.
12	Source Address	Source IP address.
13	Destination Address	Destination IP address.
14	Source Port	Source port.
15	Destination Port	Destination port.
16	Sequence Number	First byte in packet (Unique number).
17	Ack Number	TCP packet's acknowledgement number.
18	Flags	TCP flags (SYN, ACK, RST)
19	Header Length(TCP)	Minimum size: 20bytes and Maximum size: 60bytes.
20	Windows Size	Also known as receive buffer. Example: win 16560.
21	Checksum	Packet's TCP checksum values.
22	MSS Value	Maximum segment size. For example: <mss 555>, sent by source to tell destination not data more than 555 bytes.
23	Data Length	TCP data length; UDP data length.
24	Label	Label of the packets. Example "nmap (SYN Scan)"

On the other hand, Dataset Type II contains 694,461 connection features (with no duplicated data). TCPtrace is adopted to trace the

`.pcap`file and several connection features are generated, such as ACK packets sent, received retransmission, total packet sent and so forth. However, the connection features (TCP stream) are only available for TCP protocol (due to its connection-based architecture) and not available for other connectionless protocols like UDP and SCTP. The full list of features is shown in Table 2.

Table 2. Features of Dataset Type II

No.	Features	Description
1	Host_a	Each connection has two hosts. Example: host a to host b
2	Host_b	
3	Port_a	Source port and Destination port
4	Port_b	
5	First_packet	First and last packet in the connection
6	Last_packet	
7	Total_packets_a2b	Total packets sent in each direction
8	Total_packets_b2a	
9	Resets_sent_a2b	Number of the packets contain RST
10	Resets_sent_b2a	
11	ack_pkts_sent_a2b	Number of the packets contain ACK
12	ack_pkts_sent_b2a	
13	pure_acks_sent_a2b	Number of the packets contain valid ACK
14	pure_acks_sent_b2a	
15	unique_bytes_sent_a2b	Number of bytes were sent (not counting retransmission)
16	unique_bytes_sent_b2a	
17	actual_data_pkts_a2b	Number of packets that contained any amount of data
18	actual_data_pkts_b2a	
19	actual_data_bytes_a2b	Number of data bytes (including retransmission)
20	actual_data_bytes_b2a	
21	rexmt_data_pkts_a2b	Number of packets were retransmission
22	rexmt_data_pkts_b2a	
23	rexmt_data_bytes_a2b	Number of bytes were retransmission
24	rexmt_data_bytes_b2a	
25	outoforder_pkts_a2b	Number of packets were out of order

No.	Features	Description
26	outoforder_pkts_b2a	
27	pushed_data_pkts_a2b	Number of packets contain PUSH
28	pushed_data_pkts_b2a	
29	SYN/FIN_pkts_sent_a2b	Number of SYN and FIN were sent
30	SYN/FIN_pkts_sent_b2a	
31	urgent_data_pkts_a2b	Number of packets contain URG
32	urgent_data_pkts_b2a	
33	urgent_data_bytes_a2b	Number of data bytes of URG packets
34	urgent_data_bytes_b2a	
35	mss_requested_a2b	The requested Maximum Segment Size
36	mss_requested_b2a	
37	max_segm_size_a2b	Largest segment
38	max_segm_size_b2a	
39	min_segm_size_a2b	Smallest segment
40	min_segm_size_b2a	
41	avg_segm_size_a2b	Average segment
42	avg_segm_size_b2a	
43	max_win_adv_a2b	Largest window advertisement that was sent
44	max_win_adv_b2a	
45	min_win_adv_a2b	Smallest window advertisement that was sent
46	min_win_adv_b2a	
47	zero_win_adv_a2b	Number of times that sent a zero-sized window advertisement
48	zero_win_adv_b2a	
49	avg_win_adv_a2b	Average window advertisement that was sent
50	avg_win_adv_b2a	
51	initial_window_bytes_a2b	Number of bytes in the first window (before the first ACK)
52	initial_window_bytes_b2a	
53	initial_window_pkts_a2b	Number of packets in the first window (before the first ACK)
54	initial_window_pkts_b2a	
55	missed_data_a2b	Number of data bytes were in the stream that we didn’t see
56	missed_data_b2a	
57	truncated_data_a2b	Number of data bytes been truncated
58	truncated_data_b2a	
59	truncated_packets_a2b	Number of packets been truncated
60	truncated_packets_b2a	

Table 2. (Continued)

No.	Features	Description
61	data_xmit_time_a2b	Number of time data retransmit
62	data_xmit_time_b2a	
63	idletime_max_a2b	Maximum and Minimum of idle time
64	idletime_max_b2a	
65	throughput_a2b	The data throughput (Bytes/second)
66	throughput_b2a	
67	Label	Example: normal or nmap (SYN Scan)

The main aim of this study is to generate a network-based IDS dataset, thus, we are focusing on two types of attacks: probing and Denial-of-Service (DoS). To make up the completeness of this dataset, three types of activities are logged in the period of 1 year, including the normal traffics, probing traffic patterns and DoS traffic patterns. The traffic patterns may appear differently when the attacks are initiated by using different tools. Thus, a number of probing tools and DoS tools are employed in this study to simulate the attacks.

Normal / Non-Malicious Traffics

To differentiate the malicious traffics from the normal one, it is important to have some normal traffics in the dataset for a machine learner to learn. The normal traffics were collected from some normal activities conducted in WWW environment, such as web surfing, as well as logging in and out from some accounts (sensitive data has been removed in the dataset, we are only interested on the traffic patterns). All of these "benign" traffics were captured by WinDump (for Windows machines) or TCPDump (for Ubuntu machine). An example is shown in Figure 5.

```
C:\Documents and Settings\Administrator\Desktop>windump -i 1 -s 0 -w xpNORMAL.pc
ap host 192.168.1.114
```

Figure 5. Example of command to capture packets.

Probing

Probing, or also known as scanning is one of the crucial stages during penetration testing or illegal hacking. Through the probing, many useful information can be discovered from the victim machines, i.e., operating system, port statuses, services, as well as vulnerabilities of the system. In this dataset, seven probing tools are used: Nmap (encompassing TCP full-connect scan, stealthy scan, UDP scan, SCTP INIT scan, null scan, FIN scan, Xmas scan, ACK scan, Window scan, Maimon scan, SCTP COOKIE ECHO scan, IP Protocol scan, and Idle scan), Acunetix Web Vulnerability Scanner, GFI LanGuard, Nessus Vulnerability Scanner, OpenVAS, Retina Network Community, and Shadow Security Scanner. Each of the aforementioned scanners was used to scan the Ubuntu, Windows XP, as well as Windows 7, all traffic patterns (i.e., the way the scanner requesting from tested machines, and how the tested machines respond were all recorded). In this dataset, each log is labeled with its type of scanning, as well as the tools used.

Nmap

Nmap (Network Mapper) is a security scanner which commonly used to discover the hosts and services on a computer network. It is written by Gordon Lyon and released from its official web: https://nmap.org. Nmap is generally used to determine the available hosts, detecting the running services, grabbing the operating systems, and many more. Nmap is preloaded in Kali Linux, and several scan types can be accomplished with a series of commands. Snapshots of TCP full-connect scan, stealthy scan, UDP scan, SCTP INIT scan, null scan, FIN scan, Xmas scan, ACK scan, Window scan, Maimon scan, SCTP COOKIE ECHO scan, IP Protocol scan, and Idle scan are shown in Figure 6 – 18 respectively.

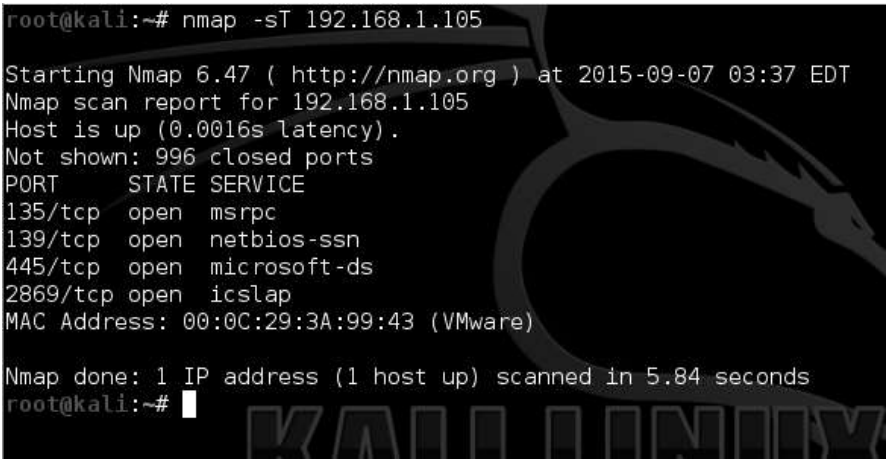


Figure 6. TCP full-connect scan.

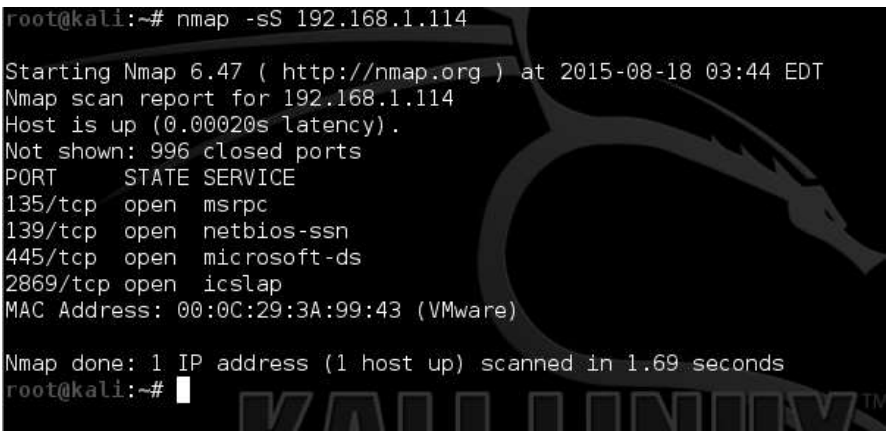


Figure 7. TCP SYN stealth scan.

Figure 8. UDP scan.

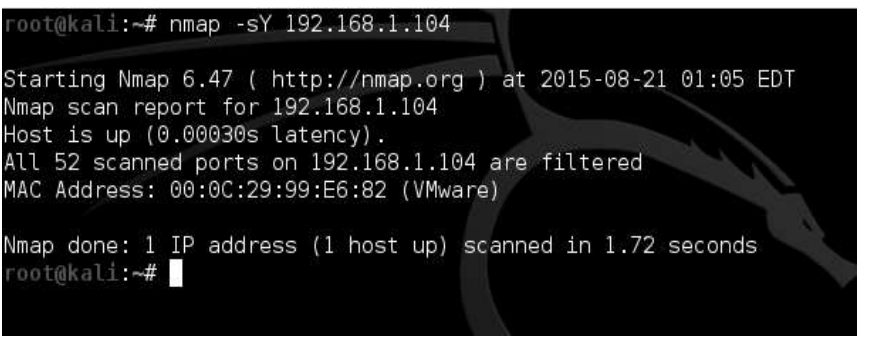


Figure 9. SCTP INIT scan.

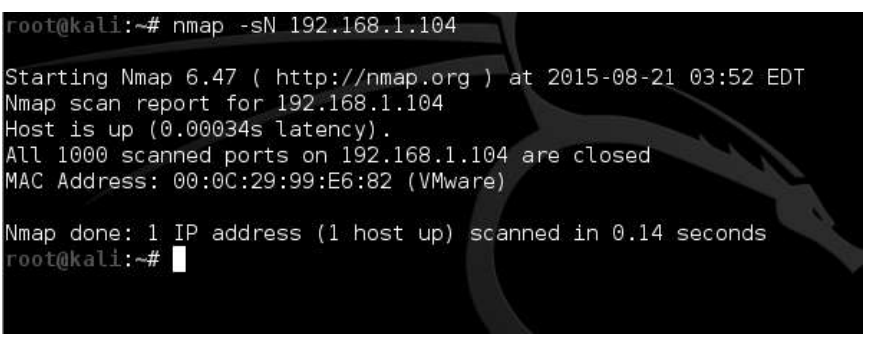


Figure 10. Null scan.

Figure 11. FIN scan.

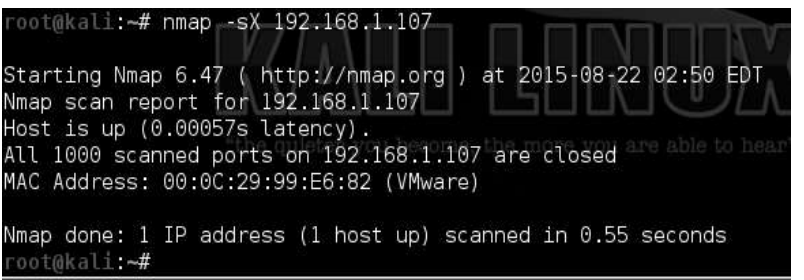


Figure 12. XMAS scan.

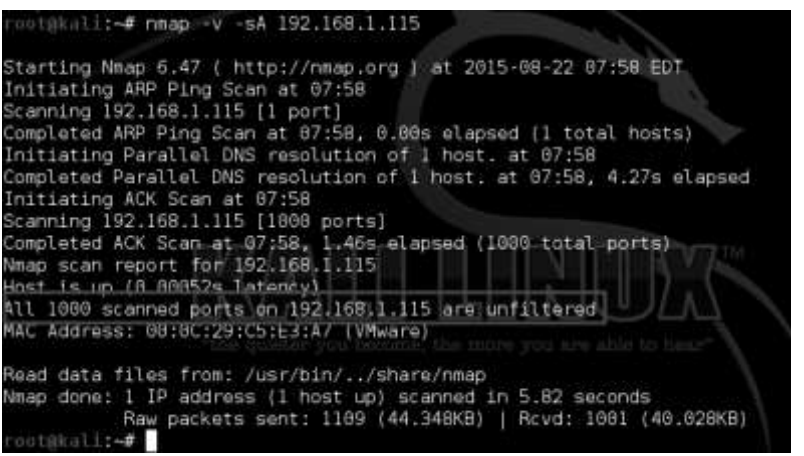


Figure 13. ACK scan.

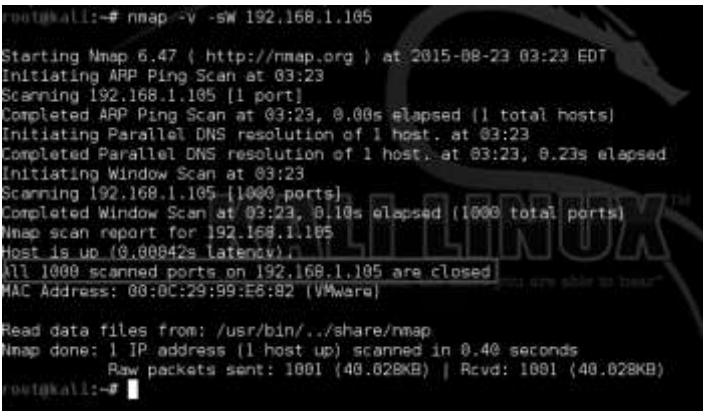


Figure 14. Window scan.

Figure 15. Maimon scan.

Figure 16. SCTP COOKIE ECHO scan.

Figure 17. IP protocol scan.

Figure 18. Idle scan.

Acunetix Web Vulnerability Scanner (WVS)

Acunetix Web Vulnerability Scanner is a tool which commonly used to discover and identify security holes in the web application. Acunetix WVS will constantly crawling the website and automatically analyzes the web application. This tool able to detect a number of vulnerabilities such

as cross site scripting (XSS), weak password strength, weak authentication pages, SQL injection, etc. To monitor how it scans the server, we have set up a website hosted on Apache server for the data collection purpose. Partial results from Acunetix WVS is shown in Figure 19. The traffic patterns were logged from the moment we initiated the Acunetix WVS, until the scanning completed.

Figure 19. Example of Acunetix WVS interface.

GFI Languard

GFI LanGuard is a vulnerability scanner that designs to scan and detect vulnerabilities. It contains powerful features like patch management for various operating systems, vulnerability assessment, network and software

auditing and so forth. GFI LanGuard 2015 was adopted for the data collection purpose. Five types of scanning were performed: Full Scan, Full TCP and UDP Scan, Full Vulnerability Assessment, Port Scanner and Top SANS 20 Vulnerabilities (as depicted in Figure 20 – 24).

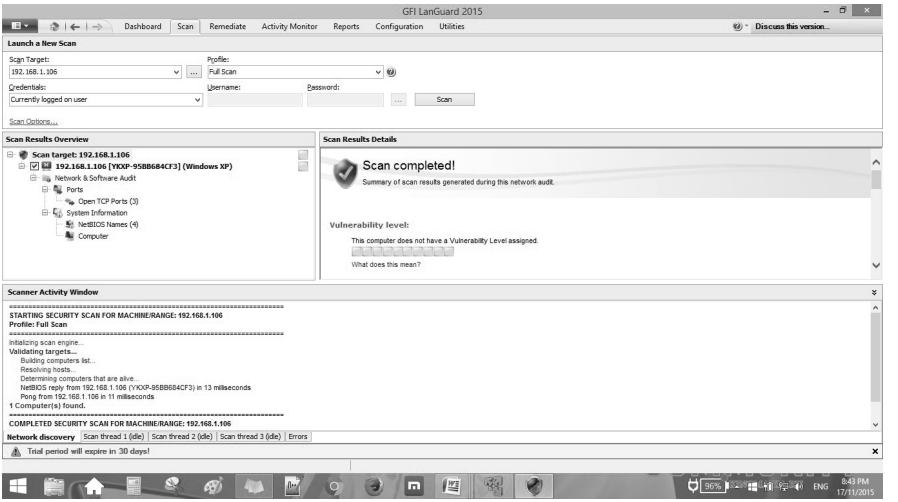


Figure 20. Full scan.

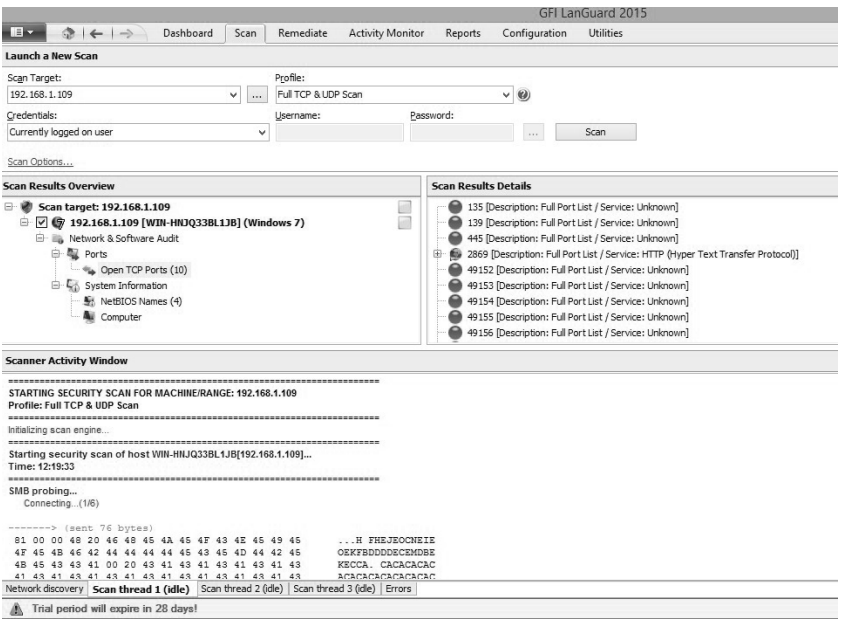


Figure 21. Full TCP and UDP scan.

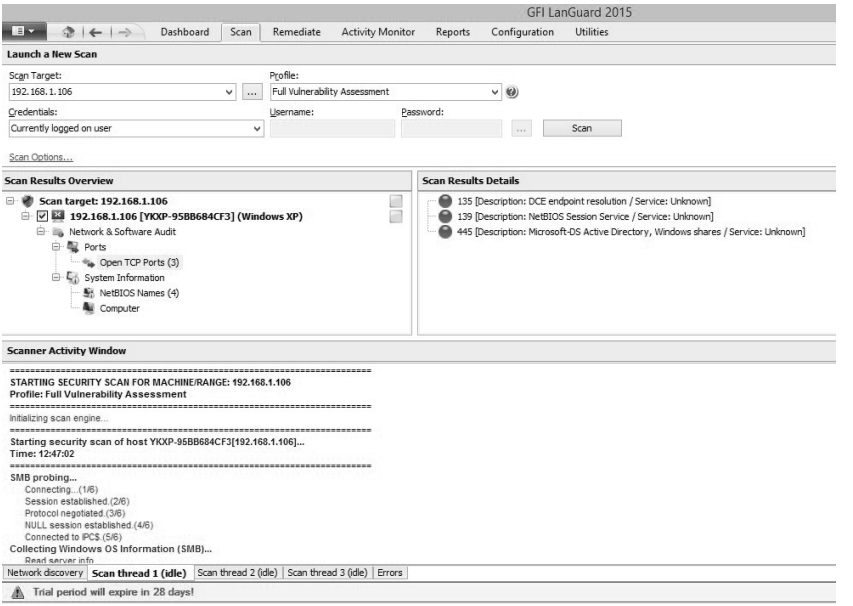


Figure 22. Full vulnerability assessment.

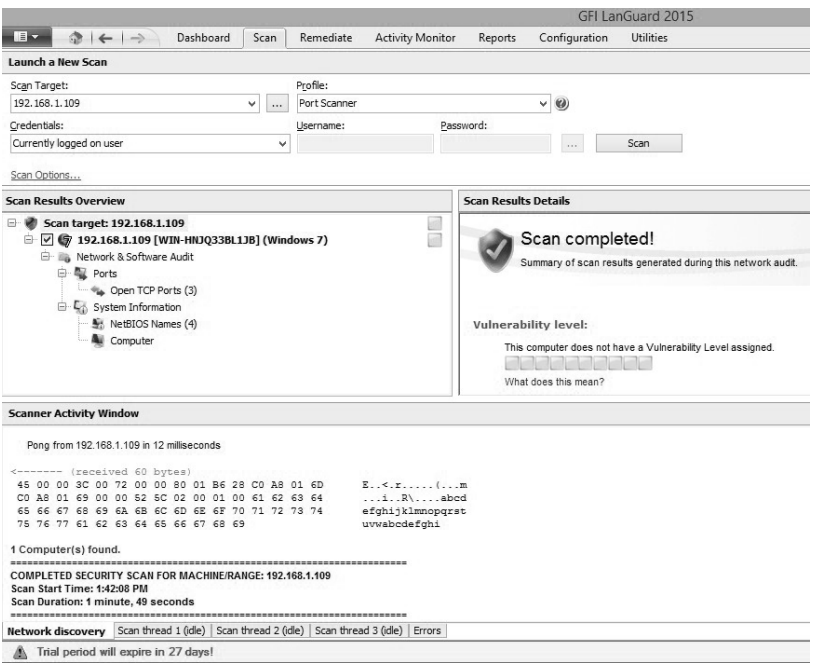


Figure 23. Port scanner.

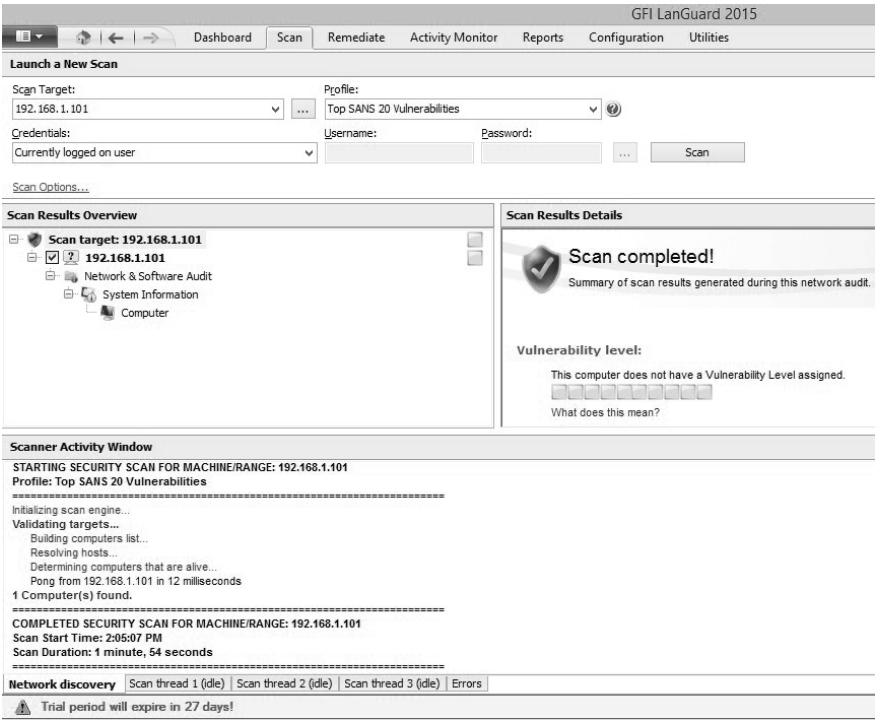
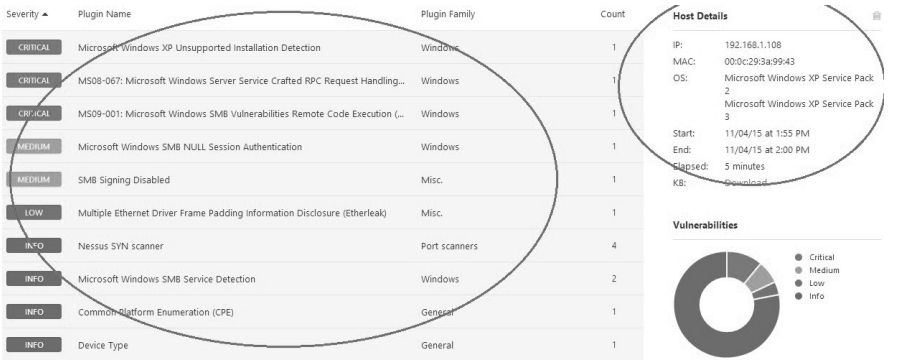


Figure 24. Top SANS 20 vulnerabilities.

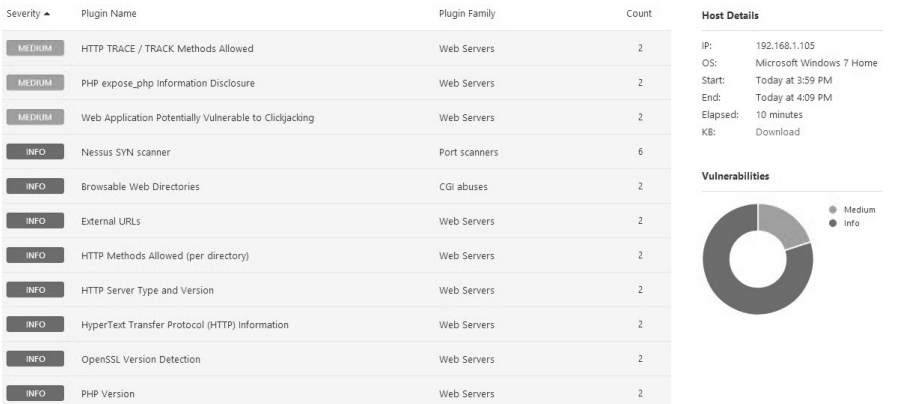
Nessus Vulnerability Scanner

Nessus Vulnerability Scanner is a remote vulnerability scanner developed by Tenable Network Security (released at: www.tenable.com). Nessus is able to perform various scanning modules including network scan, credentialed patch audit, host discovery, mobile device scan and many more. Nessus supports multiple operating systems like Windows, Mac and Linux. For data collection purpose, Nessus Home edition (ver 6.5.2) was adopted. Two types of scanning were observed: Basic Network Scan (sample is shown in Figure 25) and Web Application Tests (sample is shown in Figure 26).



Severity	Plugin Name	Plugin Family	Count
CRITICAL	Microsoft Windows XP Unsupported Installation Detection	Windows	1
CRITICAL	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handling...	Windows	1
CRITICAL	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution (...	Windows	1
MEDIUM	Microsoft Windows SMB NULL Session Authentication	Windows	1
MEDIUM	SMB Signing Disabled	Misc.	1
LOW	Multiple Ethernet Driver Frame Padding Information Disclosure (Etherleak)	Misc.	1
INFO	Nessus SYN scanner	Port scanners	4
INFO	Microsoft Windows SMB Service Detection	Windows	2
INFO	Common Platform Enumeration (CPE)	General	1
INFO	Device Type	General	1

Figure 25. Basic network scan.



Severity	Plugin Name	Plugin Family	Count
MEDIUM	HTTP TRACE / TRACK Methods Allowed	Web Servers	2
MEDIUM	PHP expose_php Information Disclosure	Web Servers	2
MEDIUM	Web Application Potentially Vulnerable to Clickjacking	Web Servers	2
INFO	Nessus SYN scanner	Port scanners	6
INFO	Browsable Web Directories	CGI abuses	2
INFO	External URLs	Web Servers	2
INFO	HTTP Methods Allowed (per directory)	Web Servers	2
INFO	HTTP Server Type and Version	Web Servers	2
INFO	HyperText Transfer Protocol (HTTP) Information	Web Servers	2
INFO	OpenSSL Version Detection	Web Servers	2
INFO	PHP Version	Web Servers	2

Figure 26. Web application tests.

OpenVAS

OpenVAS also known as Open Vulnerability Assessment System. It is an open source vulnerability scanner and commonly used to determine the security vulnerabilities of a system. The limitation of OpenVAS is that it only support Linux and not compatible on Windows platform. Figure 27depicts the OpenVAS implementation.

Vulnerability	Severity	Host	Location	Actions
DCE Services Enumeration	5.0 (Medium)	192.168.1.104 (WIN-HNJQ33BL1JB)	135/tcp	
DCE Services Enumeration	5.0 (Medium)	192.168.1.104 (WIN-HNJQ33BL1JB)	135/tcp	
TCP timestamps	2.6 (Low)	192.168.1.104 (WIN-HNJQ33BL1JB)	general/tcp	
CPE Inventory	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	general/CPE-T	
ICMP Timestamp Detection	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	general/icmp	
OS fingerprinting	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	general/tcp	
Traceroute	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	general/tcp	
Using NetBIOS to retrieve information from a Windows host	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	137/udp	
SMB on port 445	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	139/tcp	
SMB NativeLanMan	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	445/tcp	
SMB log in	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	445/tcp	
SMB on port 445	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	445/tcp	
SMB Brute Force Logins With Default Credentials	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	445/tcp	
SMB Brute Force Logins With Default Credentials	0.0 (Log)	192.168.1.104 (WIN-HNJQ33BL1JB)	445/tcp	

Figure 27. OpenVAS scanning.

Retina Network Community

Retina Network Community is a vulnerability scanning tool developed by eEye Digital Security. It identifies vulnerabilities like configuration issue, zero-day vulnerabilities and missing patches. Retina Network Community version 5.23.1.3050 was adopted for data collection purpose. Sample snapshot is shown in Figure 28.

Shadow Security Scanner

Shadow Security Scanner is a computer network security vulnerability scanner that available with over 5000 audits. Shadow Security Scanner can provide a reliable detection on a wide range of security vulnerabilities. Shadow Security Scanner analyzes the data collected and locates the vulnerabilities in system upon the completion of scanning. However, it is only compatible with Windows platforms. Three types of scanning were accomplished in our work: Complete Scan, Only NetBIOS Scan and SANS/FBI Top 20 Scan. An example is shown in Figure 29.

NetBIOS/SMB Information Disclosure

Audit ID:	17601
Risk Level:	Information
PCI Level:	Low (Pass) - Default
CVSS Score:	N/A
BugtraqID:	N/A
CVE:	N/A
CCE:	N/A
Exploit:	No
Context:	UDP:137
Result:	Success
Tested Value:	RBC T WB FB578400
Found Value:	FB57840000000001000000020434B4141414141414141414141414141414141414141414141414141414141414141414141414100002100010000000000770457494E2D484E4A513333424C314A420004005 74F524B47524F5550202020202020000840057494E2D484E4A5 13333424C314A42200400574F524B4752 (truncated...)

ICMP Timestamp Request

Audit ID:	3688
Risk Level:	Information
PCI Level:	Low (Pass) - Denial of Service
CVSS Score:	0 [AV:L/AC:L/Au:N/C:N/I:N/A:N]
BugtraqID:	N/A
CVE:	CVE-1999-0524
CCE:	N/A
Exploit:	No
Context:	N/A
Result:	Success
Tested Value:	1
Found Value:	1 - ICMP timestamp request is allowed

Figure 28. Retina Network Community screenshot.

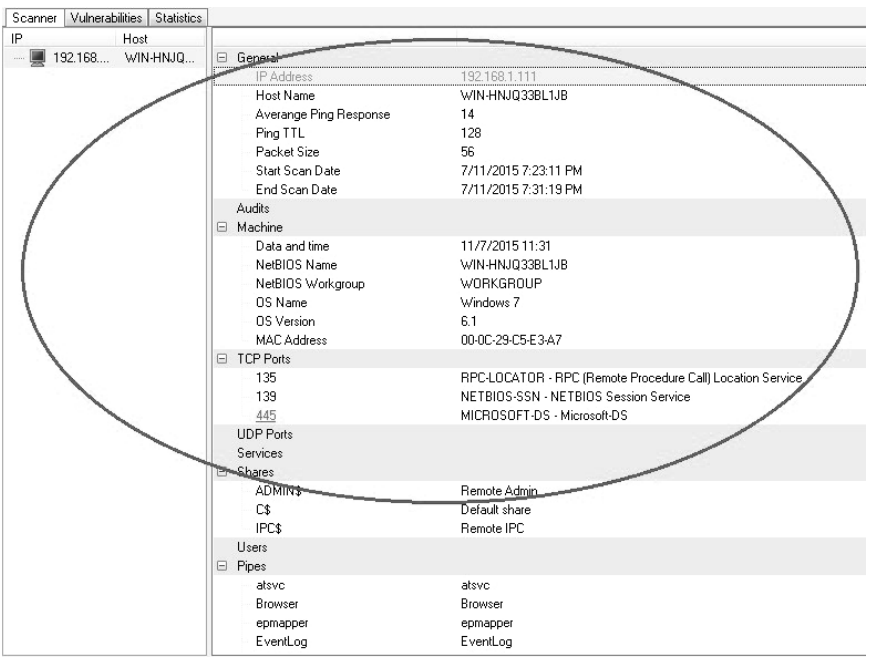


Figure 29. Complete scan.

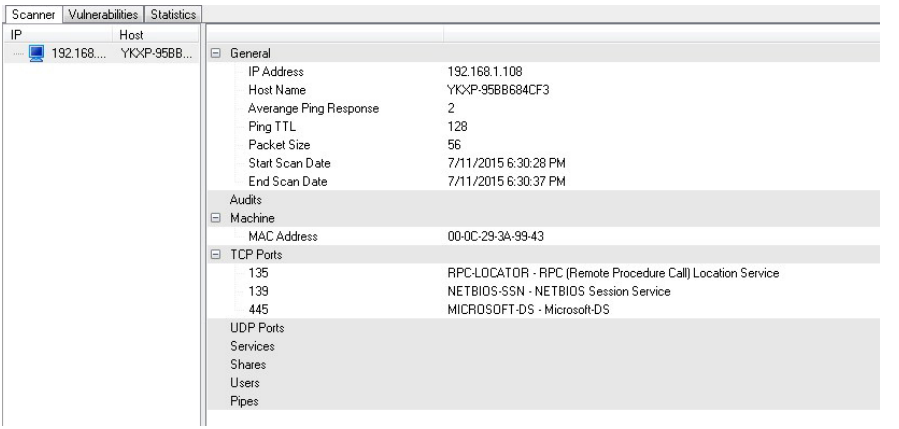


Figure 30. Only NetBIOS scan.

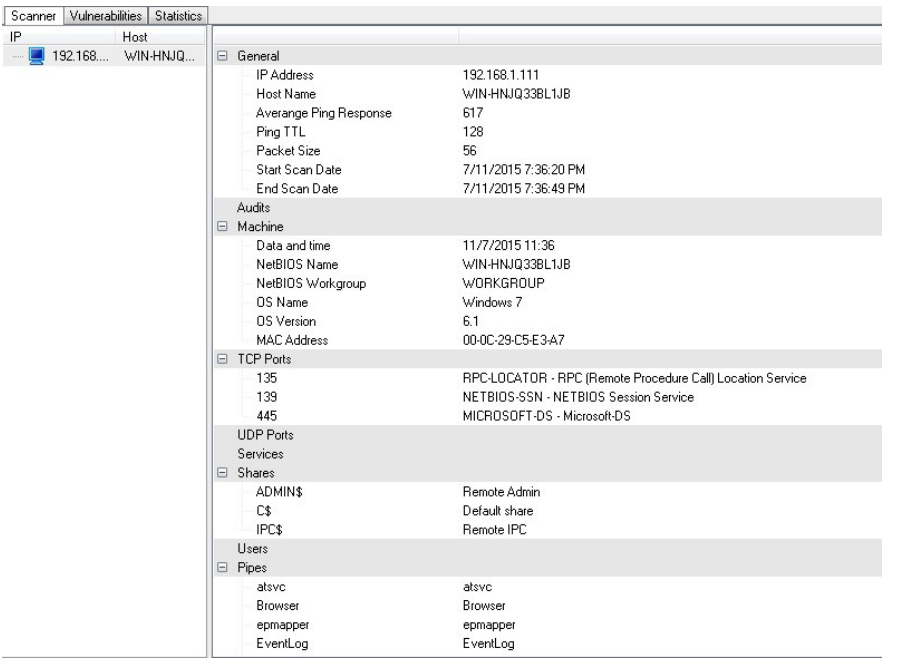


Figure 31. SANS/FBI Top 20 scan.

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS)

DoS and DDoS attacks are the attacks to consume the maximum bandwidth and make the network resources unavailable to the legitimate users. The common situation will be flooding the web server by attempting many requests or connections, and make the server cannot respond to other users temporarily, or permanently. Fifteen DoS and/or DDoS tools used in this study are Anonymous-DoS, BanglaDos, BFF DoS, DoSHTTP, Good Bye, Janidos v3, Low Orbit Ion Cannon (LOIC), OWASP HTTP DoS Tool, PHP DoS, Siege, Smurf6, Sprut, T50, Iaxflood, and Inviteflood. All of them are depicted in Figure 32 – 46. All DoS/ DDoS attacks were launched towards a make up Apache server hosted in Ubuntu as well as Windows 7 respectively. The usage of DoS/ DDoS tools is pretty straightforward, almost of the aforementioned tools are available with GUI interface. To launch the attacks, it only required the target IP address. In this dataset, each log is labeled, as well as the tools used.

Figure 32. Anonymous-DoS.

Figure 33. BanglaDoS.

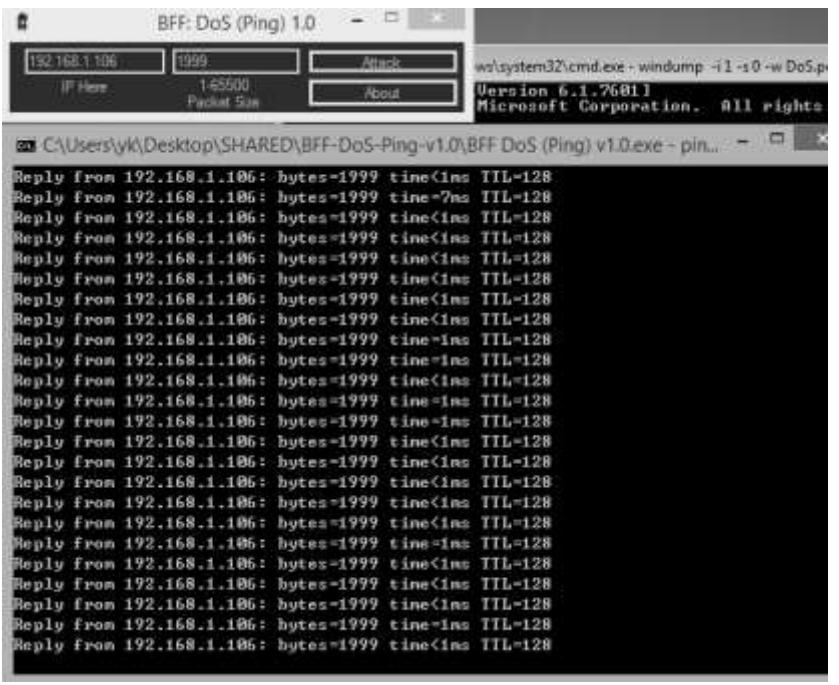


Figure 34. BFF DoS.

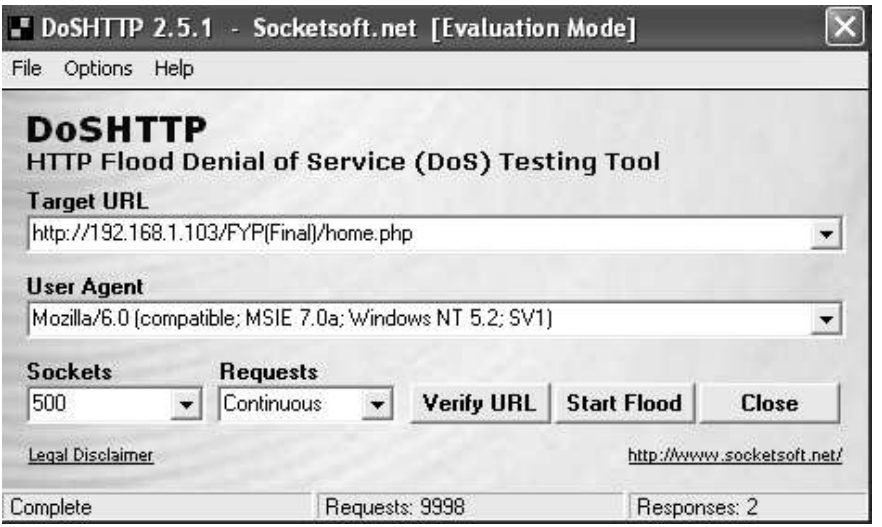


Figure 35. DoSHTTP.

Figure 36. Good Bye.

Figure 37. Janidos v3.

Figure 38. LOIC.

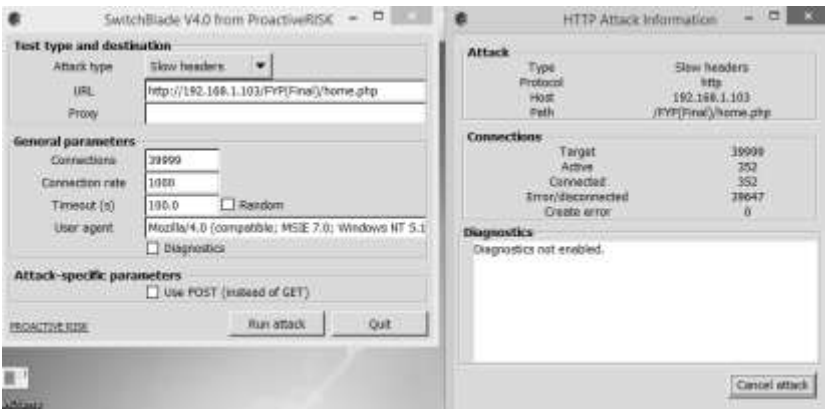


Figure 39. OWASP HTTP DoS tool.

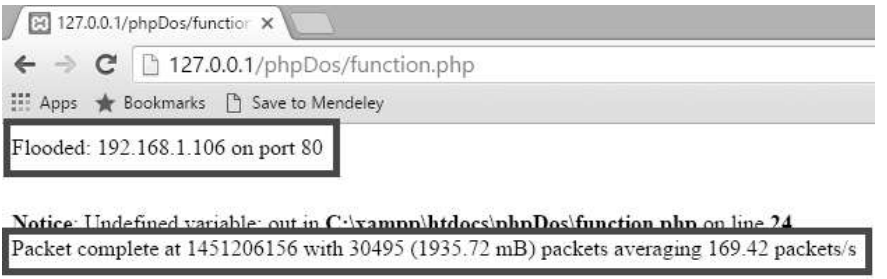


Figure 40. PHP DoS.

Figure 41. Siege.

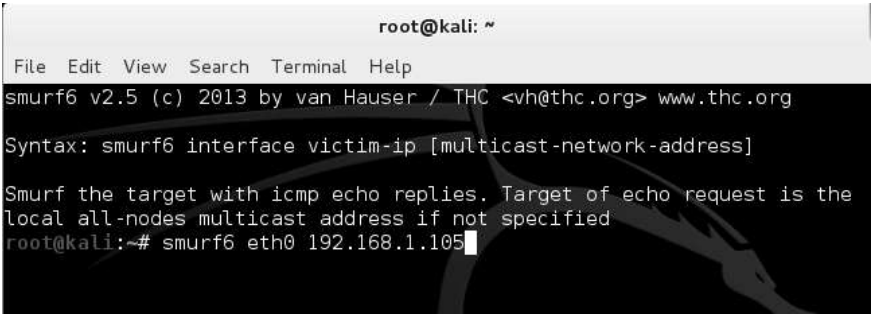


Figure 42. Smurf6.

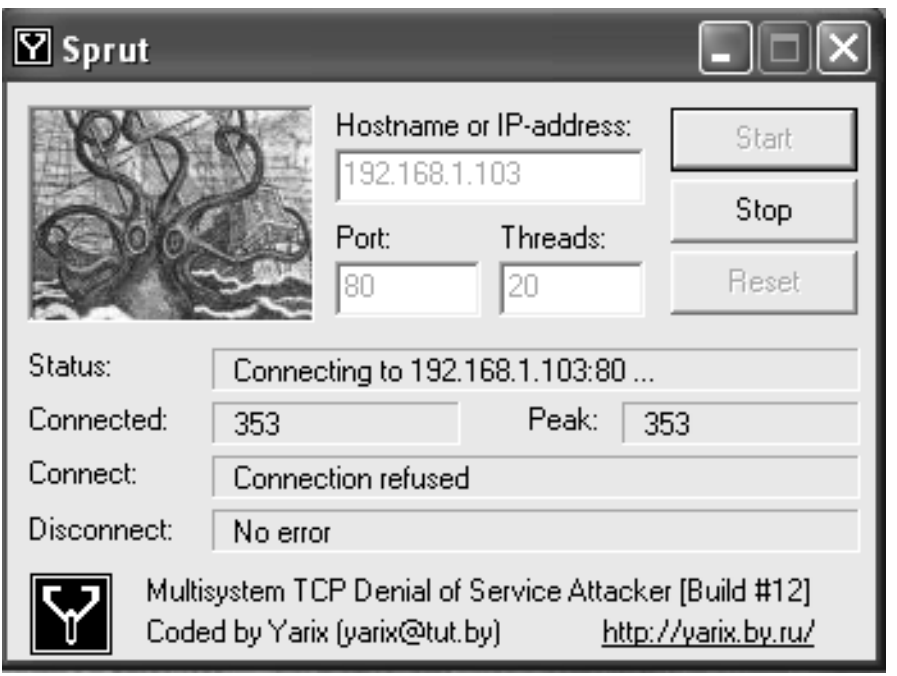


Figure 43. Sprut.

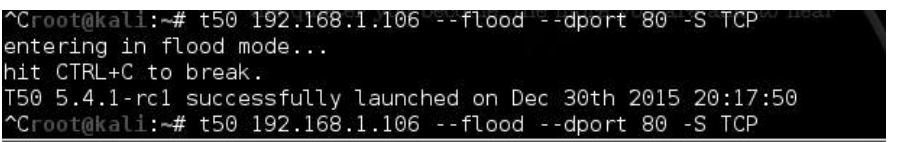


Figure 44. T50.

Figure 45. Iaxflood.

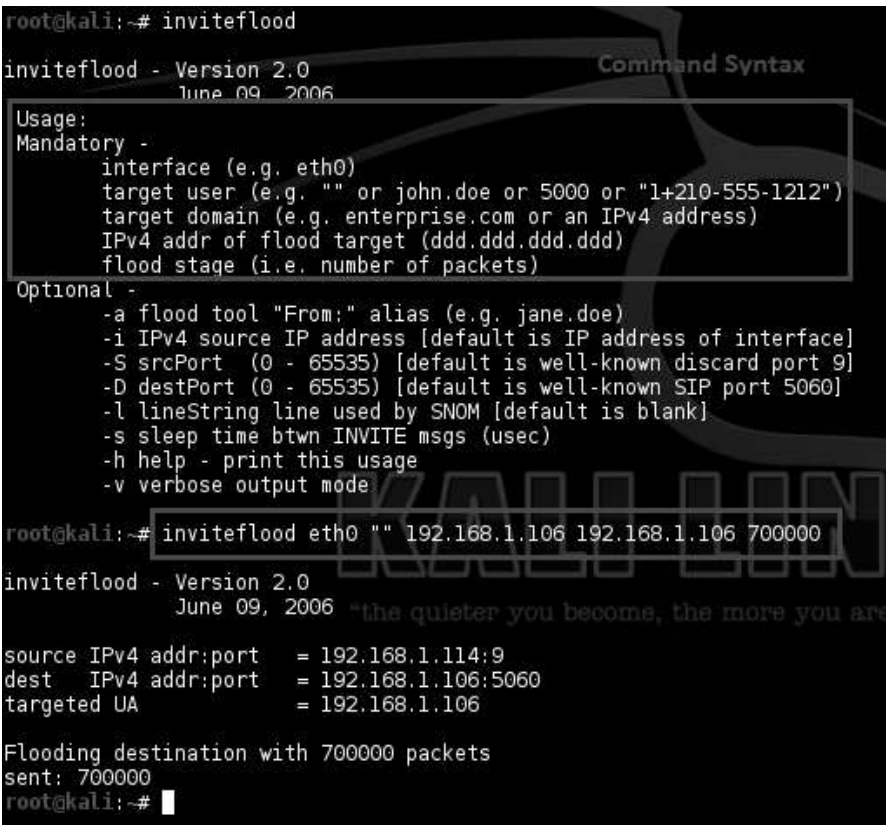


Figure 46. Inviteflood.

PRELIMINARY EXPERIMENTAL EVALUATIONS

To ensure the usability of the collected dataset, a benchmark classifier – which is J48 (or widely known as C4.5) from Weka package has been adopted in classifying the dataset. The experimental evaluation in this chapter is based on a 10-fold cross validation (for Type III error) (Kohavi, 1995), where the full dataset is subdivided into 10 subsets of data. Testing and training are performed 10 times repeatedly on a rotation basis, and in a discrete manner. Each time, a subset is left out to serve as testing set, while the rest of 9 subsets constitute the training set. All experiments are conducted in the environment of an Intel Core i7-2920XM 2.50GHz processor PC and 16GB RAM with a Windows 7 operating system. All attributes in the 11 datasets are used without applying any additional attribute reduction algorithms. The results for Dataset Type I is shown as below:

```
Classifer: Pruned J48
Dataset: Dataset Type I
Number of leaves:   142519
Size of the tree:   143676

Time taken to build model: 1361.6 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances   8186069       99.8146 %
Incorrectly Classified Instances 152050.1854 %
Kappa statistic0.998
Mean absolute error0.0001
Root mean squared error0.0081
Relative absolute error0.3002 %
Root relative squared error5.5335 %
Total Number of Instances  8201274
```

Although the detection rate is considered high, which is 99.8146%, however, in reality, this may not be consider good when looking on the

number of attacks which are missed to get detected (15205). One single attack might ruined up the overall security defense of an organization, thus, we postulated that many researches should be done to improve the detection rate. The same experimental settings also applied to Dataset Type II, and the results are shown below:

```
Classifer: Pruned J48
Dataset: Dataset Type II
Number of leaves:   119
Size of the tree:   173

Time taken to build model: 103.19 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances   694438      99.9967 %
Incorrectly Classified Instances  230.0033 %
Kappa statistic          1
Mean absolute error        0
Root mean squared error      0.0014
Relative absolute error      0.0051 %
Root relative squared error    0.8731 %
Total Number of Instances694461
```

Both datasets are working well, and we also packaged them in `.csv` format, as well as `.arff` format to support their usages in different data mining tools. The dataset can be acquired by contacting the corresponding author, and they are entirely free for research purpose.

CONCLUSION

In this chapter, a new and modern network-based IDS dataset is contributed. A series of real-life network attacks like probing and DoS/ DDoS attacks are conducted on a number of legal resources. The dataset is filtered from unwanted and duplicated traffics, and all attacks have been

labelled as per accordingly. The usability our dataset has been carried out by using the J48 decision tree from Weka package. For the future direction, this network-based IDS dataset will be expand to cover more challenges, i.e., by adding some noises, fake alarms, etc.

REFERENCES

Aggarwal, P. and S. K. Sharma, "Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection. (2015). "*Procedia Comput. Sci.*, vol. 57, pp. 842–851.

Aissa, N. B. and M. Guerroumi, "Semi-supervised Statistical Approach for Network Anomaly Detection. (2016). "*Procedia Comput. Sci.*, vol. 83, no. Fams, pp. 1090–1095.

Creech, G. and J. Hu, "Generation of a new IDS test dataset: Time to retire the KDD collection. (2013). " *IEEE Wirel. Commun. Netw. Conf. WCNC*, pp. 4487–4492.

Folino, G. and F. S. Pisani, "Evolving meta-ensemble of classifiers for handling incomplete and unbalanced datasets in the cyber security domain. (2016). "*Appl. Soft Comput. J.*, vol. 47, pp. 179–190. ~~, 2016.~~

Kaskar, J., R. Bhatt, R. Shirsath, and A. Ids. (2014). "A System for Detection of Distributed Denial of Service (DDoS) Attacks using KDD Cup Data Set," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 3551–3555.

Kohavi, R. (1995). "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, *International Joint Conference on Artificial Intelligence (IJCAI)*, 1995, vol. 5, pp. 1137–1143.

Lippmann, R., J. W. Haines, D. J. Fried, J. Korba, and K. Das, (2000). "1999 DARPA off-line intrusion detection evaluation, "*Comput. Networks*, vol. 34, no. 4, pp. 579–595.

Moustafa, N. and J. Slay, "Creating Novel Features to Anomaly Network Detection using DARPA-2009 Dataset. (2015). "*Proc. 14th Eur. Conf. Cyber Warf. Secur. ECCWS*, no. April, pp. 204–212.

Ooi, S. Y., S. C. Tan, and W. P. Cheah, "LNCS 8836 - Anomaly Based Intrusion Detection through Temporal Classification. (2014). " *Lect. Notes Comput. Sci. (LNCS), 21st Int. Conf. Neural Inf. Process. (ICONIP 2014)*, pp. 612–619.

Ray, L., "Training And Testing Anomaly-Based Neural Network Intrusion Detection Systems. (2013). "*Int. J. Inf. Secur. Sci.*, vol. 2, no. 2, pp. 57–63.

Shiravi, A., H. Shiravi, M. Tavallaee, and A. A. Ghorbani. (2012). "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374.

Song, J., H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao. (2011). "Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation," *Proc. First Work. Build. Anal. Datasets Gather. Exp. Returns Secur. - BADGERS '11*, pp. 29–36.

Tan, Z., A. Jamdagni, X. He, P. Nanda, and R. P. Liu. (2014). "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 447–456.

Tavallaee, M., E. Bagheri, W. Lu, and A. A. Ghorbani. (2009). "A detailed analysis of the KDD CUP 99 data set," *IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, no. Cisda, pp. 1–6.

Zuech, R., T. M. Khoshgoftaar, N. Seliya, M. M. Najafabadi, and C. Kemp. (2015). "A New Intrusion Detection Benchmarking System," *Proc. Twenty-Eighth Int. Florida Artif. Intell. Res. Soc. Conf.*, no. McHugh, pp. 252–255.

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 10

SEAMLESS BIOMETRICS FOR A SMART OFFICE

Soh Ting Yong and Michael Goh Kah Ong*
Faculty of Information Science and Technology,
Multimedia University, Melaka, Malaysia

ABSTRACT

The use of biometric technology to provide strict security to a smart office has become rampant recently. However, most of the existing biometric applications require some extent of interaction from the users. For example, the current fingerprint recognition system requires the users to touch the sensor; and a face recognition system requires proper position of the head for precise recognition. These requirements may sometimes interrupt the users in their daily routines, e.g., having to stop by to be verified by the biometric system. As such, this project develops a seamless biometric solution that does not require any user interaction. The biometric system can recognize the users through their natural walking behaviour. In other words, the system is be able to provide non-intrusive and seamless access experiences to the user.

Keywords: seamless biometrics, gait recognition, pattern recognition, computer vision

* Corresponding author, Email: michael.goh@mmu.edu.my.

INTRODUCTION

Biometrics is a technology that analyzes human's unique behavioral or physical characteristics for personal identification, verification, access control, and surveillance. Biometrics is a term derived from the ancient Greek words: "Bio" means life and "metric" means measure (What are biometrics? 2017) & (Biometrics, 2017).

Traditional authentication methods like keys or passwords have many problems such as forgotten or stolen passwords, key lost, and accounts hacked. On the other hand, biometrics is a more secure, convenient, reliable and effective technique without requiring keys or cards for entry access or personal identification. Biometrics has been applied in various applications like aviation, banking, construction, government, health-care, military, smart application, transportation industry and workplaces. Besides, biometrics has also been used in law enforcement to identify suspects in crime scenes. In addition, many commercial products and applications have been developed for biometric payment and authentication such as iPhone iris scanner, M2SYS DriverTrackTM, Zwipe Access card, PayPal, Biyo Wallet, PayTango and Danal (Trader, 2013).

Biometric identifiers can be categorized into two main types that are behavioural-based and physiological-based. Behavioural characteristics refer to the behaviour patterns of a person which include gait, signatures, keystrokes, voice and speech patterns. Physiological characteristics are related to an individual's physical trait like DNA, ear, face, fingerprints, hand geometry, iris, odour, palm veins, and retina. Some other biometrics such as temporary tattoos, brainwave signal, and even heartbeats are also being studied for personal authentication.

Biometrics has been deployed in many countries including China, United States, India, United Kingdom, Germany, and Canada. Biometrics is often utilized to enhance national border security and verify the identity of foreign visitors at the immigration department too. Some countries have also incorporated biometric technology with identity card, credit cards, passports or visa to strengthen security. Biometrics is commonly

implemented in both government programs and private sectors (DriverTrack™, 2017).

BIOMETRIC SOLUTION FOR A SMART OFFICE

A growing number of smart office applications and products are using biometric technology for personal authentication. There are some examples of existing applications and products for smart offices in the market.

Iris Recognition (By IRIS ID)

Iris ID Systems Inc. (formerly known as LG Electronics, Iris Technology Division), is a world class provider for iris biometric. It has developed many iris recognition and face camera systems since 1997. One of the biometric applications developed by Iris ID for smart office is the Iris Scanner products - Iris ID iCAM TD100 series (IRIS ID, 2017).

Figure 1. Iris Recognition Product- iCAM TD100.

Most of the handheld iris recognition systems require the user to remain completely motionless in order to obtain high resolution iris images. As compared to the handheld iris recognition devices, iCAM TD100 provides the capability to capture iris images while the person or the device is in motion. The device collects high quality ISO standard compliant iris images of a subject within an optimum capture distance in

no more than one second. Besides, iCAM TD100 has dual iris captures, which are iris image capture and face image capture.

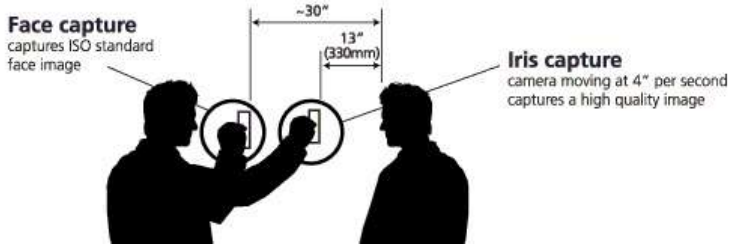


Figure 2. Settings illustration for iCAM TD100.

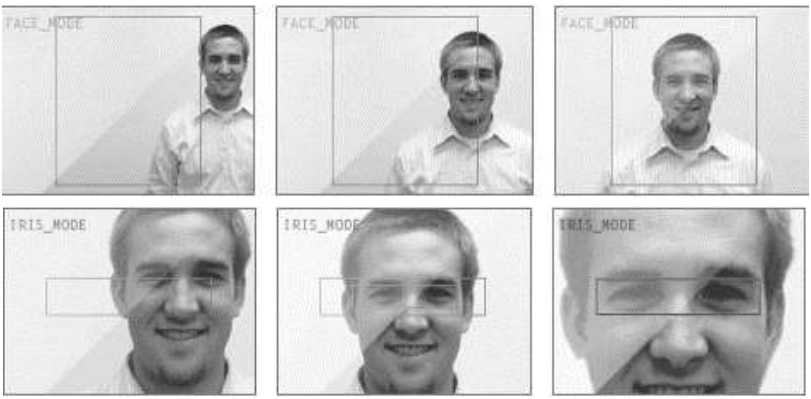


Figure 3. Face mode (1st row) and Iris mode (2nd row) of iCAM TD100.

Access Control/Time and Attendance (TNA): eNBioAccess-T9 (NITGEN's product)

eNBioAccess-T9 is one of the cutting edge fingerprint and face access controllers developed by NITGEN. It has a superior matching engine equipped with a 5 inch touch colour LCD and a built-in camera. It also provides multifactor authentication capability which includes face, fingerprint, radio frequency card and personal identification number (NITGEN, 2017).

Figure 4. eNBioAccess-T9.

Figure 5. System Configuration of eNBioAccess-T9.

Hikvision is a world class supplier of video surveillance products and solutions. Hikvision's product suite includes Smart IP cameras, access control and alarm systems, and other elements of highly complex security systems. HIKVISION DS-2CD4132FWD-IZ is a security camera developed by Hikvision to enhance workplace security.

Network Camera DS-2CD4132FWD-IZ (Hikvision's product)

HIKVISION DS-2CD4132FWD-IZ has many security features. It is integrated with smart IR, smart codec and smart focus (motorized VF lens with focus power for both daylight and night conditions), smart face detection and smart audio detection. It is also equipped with alarm triggers that alert the user of potential issues such as motion detection, alarm interference, camera disconnection from the internet, or low storage space. The product also enables connection to a facial detection camera via LAN for live video feed anywhere and anytime (HIKVISION DS-2CD4132FWD-IZ, 2017).

BIOCAM 300 (ZKAccess's Product)

Biocam300 is the world first HD IP camera that implements ZK's facial recognition algorithm. The users can be recognized from a database comprising four hundred faces within one second. The built-in infrared light source enables the users to be detected and recognized from a distance of 3.81 meter (BIOCAM300, 2017.).

Figure 6. BIOCAM 300.

PROPOSED SOLUTION

This project implements a gait recognition system to realize seamless biometric solution. A subject's walking pattern is extracted from a distance for personal authentication. An administrator is required to control and manage the smart office system. The administrator may manage one or many offices. Each of the smart offices can accommodate up to eight users. The gait recognition system enables the identity of the subject be ascertained without touching any recording probe.

First, each of the smart office users must provide their information and gait patterns for registration. The system records the gait patterns and store the patterns in a database. A camera sensor is placed on top of the office's door. The camera starts to capture the subject's gait pattern when the subject walks towards the office door (entrance of smart office). The subject's gait pattern will be sent to the server and the server directly saves

the gait pattern into a computer (Raspberry Pi) for recognition. If the subject is identified as a genuine user, the office door will be unlocked. At the same time, the appliances in the office will be automatically switched on. Before leaving the office, the user may remotely control the door and deactivate the appliances.

In this work, three core components are required to implement the gait recognition system: Kinect for Windows, a server, and Raspberry Pi. The Kinect sensor is used to capture the gait pattern data. The server controlled by the administrator is used to save the subject's gait data to the computer (Raspberry Pi). Raspberry Pi is used to execute the recognition process. The device is also used as the database to store and save all the user's gait patterns. At the same time, it also serves as a user control interface for the smart office users. The figure below illustrates the schematic for the smart office environment.

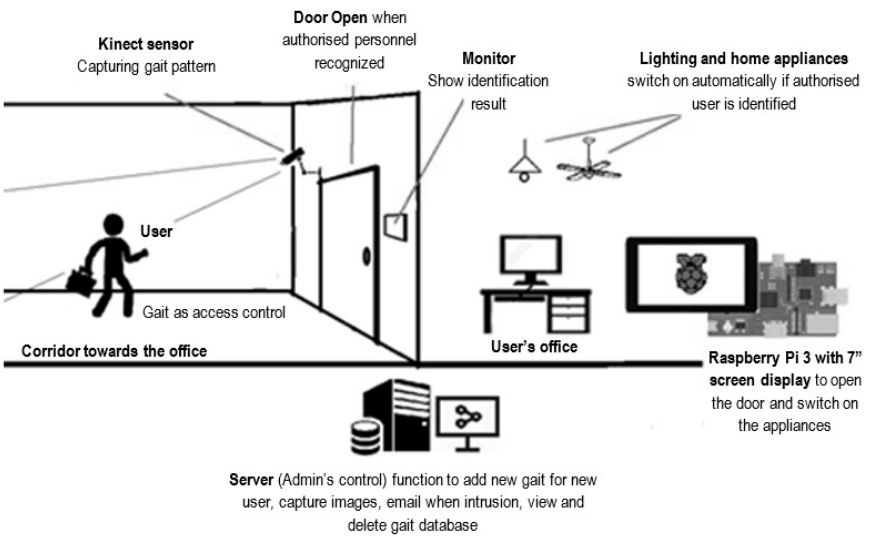


Figure 7. Scenario of Seamless Biometrics for Smart Office.

SYSTEM IMPLEMENTATION

Firstly, a Kinect for Windows sensor is used as the camera to record and track the subject's gait pattern. The Kinect skeleton tracking module is one of the applications in the Kinect for Windows Developer Toolkit. The Kinect skeletal tracking algorithm recognizes and collects the movement of twenty joints of the human body. The figure below illustrates the skeleton position tracked by the program:

During the recognition process, the movement of twenty body joints of the subject will be captured. The output is recorded in a text file as gait pattern's data. The gait pattern's data is sent to Raspberry Pi for pre-processing, features extraction and matching. The method used to save the text file directly to Raspberry Pi is called network drive's mapping in Windows.

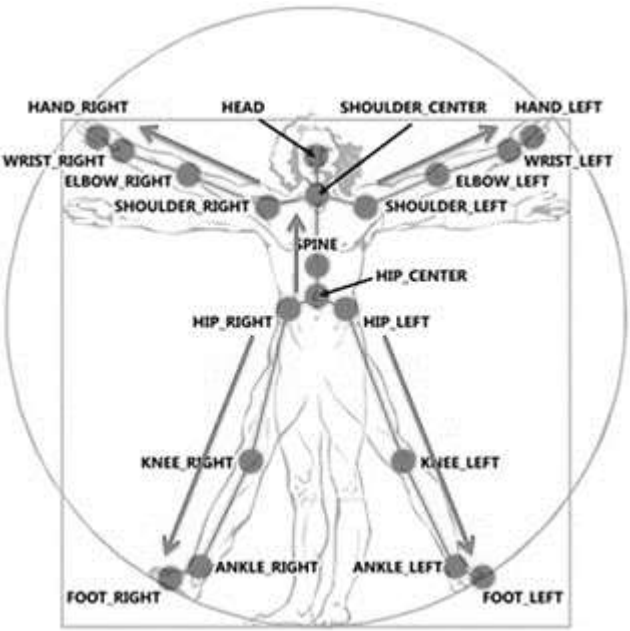


Figure 8. Skeleton position and tracking state of Kinect skeletal tracking.

The Raspberry Pi device calculates and analyses the subject's gait pattern. The subject's gait patterns are pre-processed to make the subsequent matching process more efficient. After that, the gait pattern is compared with the user's templates in the database. Euclidean distance is applied to match two gait patterns. The identification result (Is the subject the correct user and who he/she is?) will be shown after the matching process.

If the subject's identity is acknowledged as a genuine office user in the smart office, Raspberry Pi will send a signal to unlock the door. Otherwise, the office door will remain locked. Meanwhile, the appliances in office such as light and fan will also be switched on. Raspberry Pi is used as the user control interface for door activation and appliances deactivation.

SERVER INTERFACE FOR ADMINISTRATOR'S CONTROL

In this project, C# that runs on the .NET Framework is used to build and design the server interface for administrator's control. The administrator is able to control the smart office system through the server interface. The main menu of the server interface shows a welcome message and a "Go" button. After selecting a specific office function at the main menu, the screen for the specific office will be shown. An enlarged view for normal image-mode and a smaller view for Skeleton stream-mode are presented. The specific office name, current date and current time are also displayed in this window. There are seven different functional buttons provided in the office's window. The "Main Menu" button enables the administrator to go back to the main menu window of the server interface. The figures below illustrate the main menu's window and the smart office's window in the server interfaces:

The administrator can add a new user in a specific office through the "Set new gait pattern" button. A gait setting's window will be shown after the administrator triggers this button. In the gait setting's window, the administrator needs to fill the filename in a textbox. The filename must be saved in the format of "userXY" in which X represents the different user

identities (A to G only) and Y represents the captured gait's order (1 to 5 only).

Figure 9. Main menu's window in server interface of Smart Office.

Figure 10. Smart office's window in server interface of Smart Office.

The administrator can click the "Add new" button to capture the user's gait pattern. Each user will be asked to provide five gait patterns. The gait pattern's capturing process starts when the "Start" button is clicked, and stops when the "Stop & Save" button is clicked. The gait patterns are stored in a specific file (database). Through the "Open file" button, the administrator may open the file that is used to store the user's gait patterns to view or delete the gait data. The "Seated Mode" function is used to track a user who is seated, or when the lower body part is not visible to the

sensor in entirety. The figure below illustrates the gait setting's window in the server interface:

Figure 11. Gait setting's window in server interface of Smart Office.

The administrator may capture a snapshot of the user office's corridor by clicking the "Capture" button. The "Open" button enables the administrator to check the captured images in the database. Apart from this, the administrator can notify the user or the security department if a behavioural is detected at the corridor with an email by triggering the "Emergency" button. An emergency setting window will appear after triggering the button. The administrator may configure the email content before sending the email by clicking the "Send" button. Moreover, the administrator may save a text content as default in the text box by clicking the "Save" button. The administrator will be asked to fill their Gmail address, Gmail password, the receiver's email address, email's subject and content. The figure below illustrates the emergency setting window in the server interface:

Figure 12. Server interface of Smart Office.

USER CONTROL INTERFACE

A user control interface enables the office users to control the appliances remotely. It is executed on Raspberry Pi. Tkinter is a standard user interface to the Tk Graphical User Interface (GUI) toolkit. It is also a Python's de facto standard GUI. Tkinter is used to develop the user control interface on Raspberry Pi. There are two options in the user control interface. The "Just a moment" button is suitable for a user who is just leaving the office for a while. The "Just a moment" button will trigger the door to open automatically. The light and fan in the office will not be turned off. If the user leaves the office for a long period or gets off from work, he/she may choose the "Exit" button. The door will be opened automatically. The light and fan will be turned off after the user leaves. The current time is also shown in the user control interface. The figure below illustrates the smart office's user control interface:

Figure 13. User control interface of Smart Office.

GAIT RECOGNITION IMPLEMENTATION

Raspberry Pi is used as the database to store the user's gait pattern. In Raspberry Pi, there is a file named "Recognition System & Database". Three main file folders named "Guest", "Image" and "User" can be found in "Recognition System & Database" file. The "Recognition System & Database" file plays an important role when the recognition process is running. The figure below shows all folders in the "Recognition System & Database" file in Raspberry Pi:

The recognition process is divided into four main parts: data capture, pre-processing, feature extraction and matching. Three Python files named as "openif.py," "guestGetResult.py" and "userGetCalc.py" were created for the recognition process. The "openif.py" will be executed every seconds.

The recognition process starts with data capture. In this work, a set of gait patterns of the user will be collected through the Kinect sensor. The Kinect sensor detects and collects the movement of twenty joints of the tracked subject's body in the field of view. When a subject is tracked by the Kinect sensor, a "guest.txt" file will be created to store the subject's gait pattern. The "guest.txt" is saved directly to the "Guest" folder in Raspberry Pi. The figure below illustrates a gait pattern data's text file created in the "Guest" folder.

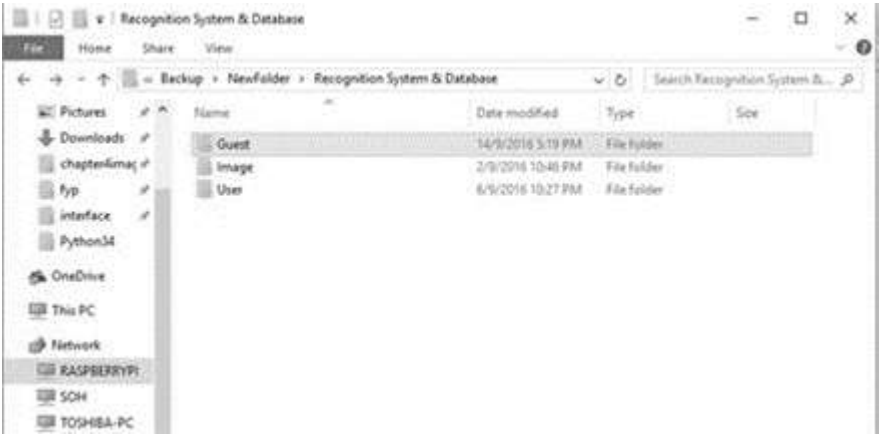


Figure 14. All folders in the "Recognition System & System" file in Raspberry Pi.

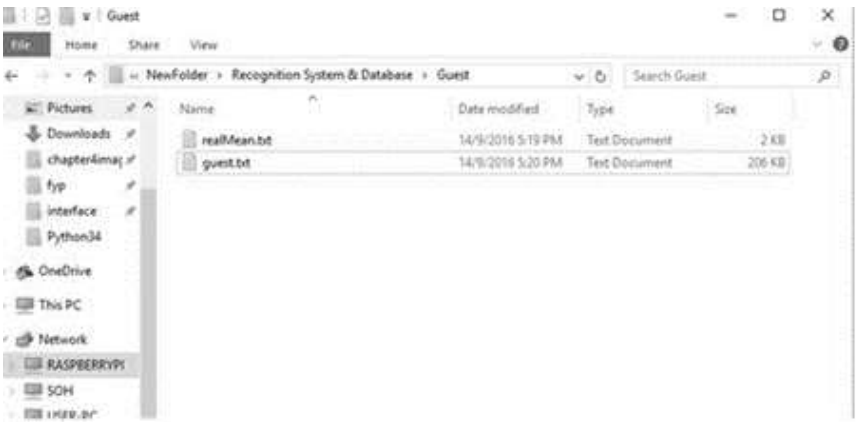


Figure 15. Gait pattern data's text file created in "Guest" folder.

The "openif.py" Python file is used to check whether a user's gait coordinate or a new subject's gait coordinate exist in the "Recognition System & Database" file. Once a subject's gait pattern data ("guest.txt") is found in the folder, "guestGetResult.py" will be executed to analyze and calculate the subject's gait pattern data. If the "userXY.txt" text file is found, the "userGetCalc.py" file will be called.

Since the Kinect Sensor is operating at a rate of 30 frames per seconds, a hundred sets of joint coordinates will be tracked and recorded in a few seconds. The figure below shows a sample file of the gait pattern data:

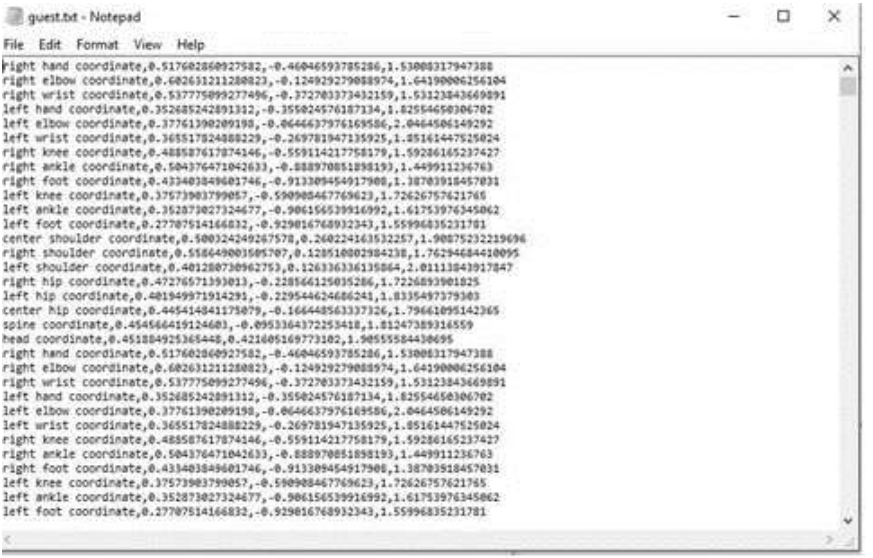


Figure 16. Sample file of gait pattern data.

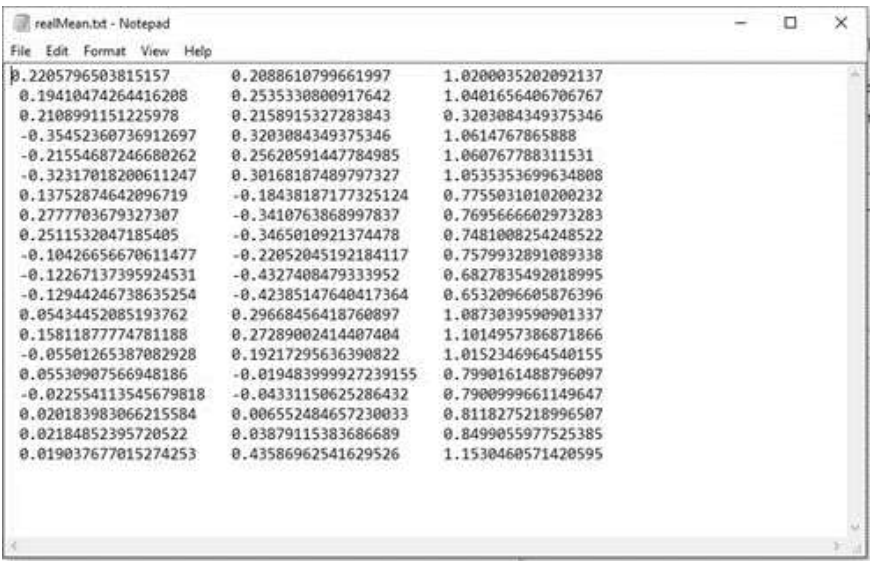


Figure 17. Result of the gait pattern data after pre-processing.

The gait pattern data needs to be pre-processed to average the hundred sets of joint coordinates. The sets of the joint coordinates are condensed into a single set by using a simple averaging method. The result of the averaged gait pattern is stored in "realMean.txt" in the "Guest" folder. The figure above illustrates an example of the compressed gait data after pre-processing:

After that, the feature extraction process is executed on the data. Euclidean distance is applied to calculate the differences between the average set of the subject's data and the average set of each users in database. Equation 1 shows the Euclidean distance in three dimensions used in the feature extraction and matching process:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \quad (1)$$

where d represents the Euclidean distance, and p and q represent the two different joint positions, respectively.

The matching process is the last process in recognition. The system looks for the closest match with the gait templates in the database. The smaller the distance value is, the closer the match becomes. The minimum distance value in the list is chosen as the best matching result. A set of threshold values is used to confirm the recognition result in order to reduce the recognition error rate. If the matching distance is not more than the default threshold value, the subject will be recognize as a genuine user. Otherwise, the subject will be rejected.

The administrator is able to create a new gait data for a new user through the server interface. Each user must provide five sets of gait pattern's data to the system. Three sets of the gait pattern's data are used as training data while the remainder as testing data. All the original user's gait pattern data will be preserved in the "user_OriginalCoordinates" folder. All the user's gait pattern data will be saved in the "user_Mean" folder. The figure below illustrates the "User" folder which stores the user's database:

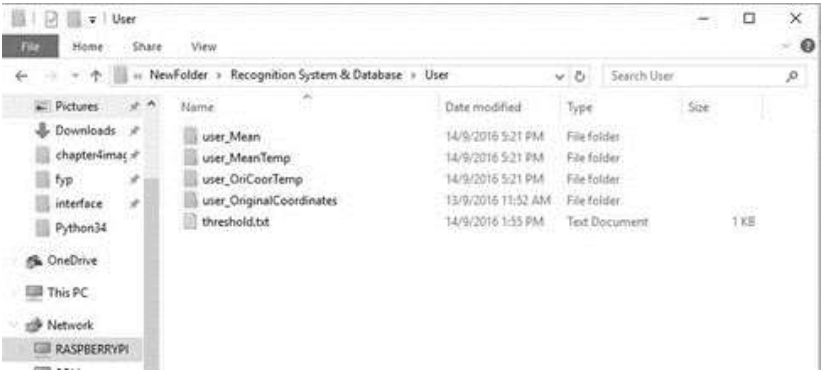


Figure 18. "User" folder which is stored the user's database.

A new gait data file (userXY.txt) will be created in a folder named "user_OriCoorTemp". At the same time, the "userGetCalc.py" Python file will be executed to analyze and calculate the new user's data. The new gait pattern data is pre-processed and saved to the "user_Mean" folder for future use. The threshold value of the new user will be calculated at the same time. The figure below illustrates the result of adding a new user:

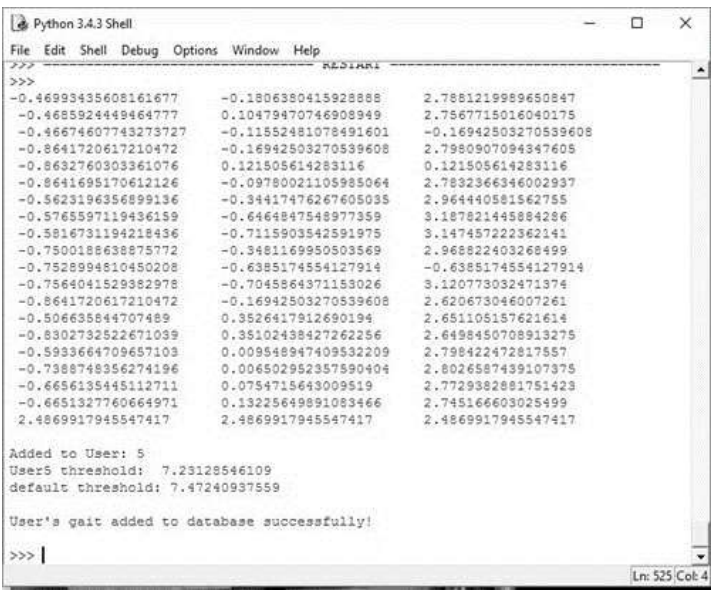


Figure 19. Result of adding a new user.

APPLIANCE AUTOMATION

In the smart office application, the door, lighting and other electronic appliances can be controlled remotely. The automation feature allows the user to have improved convenience and save energy consumption at the same time. In this work, Raspberry Pi is used to implement appliance automation. Raspberry Pi is able to control and monitor many devices such as LEDs and motors by connection to electronic circuits through General Purpose Input Output (GPIO).

A stepper motor is used to represent an office door. Stepper motor is a DC motor that moves in discrete steps. The stepper motor can rotate to specific direction in specific speed by energizing the multiple coils inside the motor. The stepper motor is controlled by the Raspberry Pi (Earl, 2017). In this work, a 28BJY-48 stepper motor with ULN2003 control board is used for door automation. The figure below illustrates a 28BJY-48 stepper motor with ULN2003 control board controlled by a Raspberry Pi:

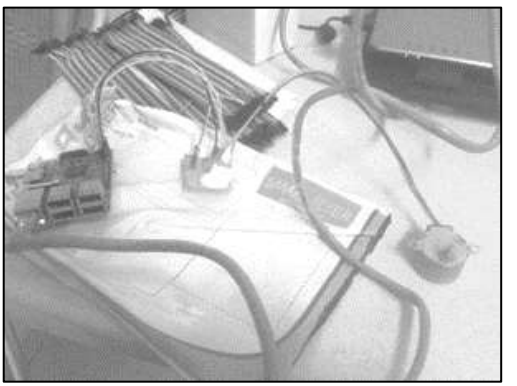


Figure 20. 28BYJ-48 stepper motor with ULN2003 driver controlled by a Raspberry Pi.

Once a user is identified by the system, Raspberry Pi runs the Python script named "stepperMotorDoor.py" to rotate the stepper motor simulating a door opening action. The stepper motor rotates back to the original position after three seconds for a door closing action. The user may also remote control the door through the user control interface to

trigger the "stepperMotorDoor.py" Python file. The figure below depicts a scenario of a small scale model office door and a stepper motor:

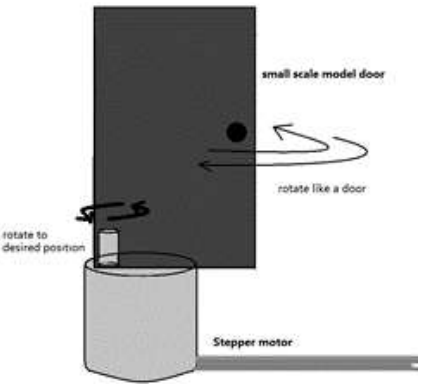


Figure 21. Scenario of a small scale office model door and a stepper motor.

The lighting and electronic appliances in the office will also be switched on automatically when the user enters the office. In this work, two LED bulbs are used to represent the lighting appliances in the office room. A python script named "ledOn.py" for lighting up the LED bulbs is implemented. The LED bulbs can be turned off remotely through the user control interface. A python script named "ledOff.py" is triggered when the user chooses the "Leave" button. The figure below illustrates a simple LED circuit connected to the Raspberry Pi:

Figure 22. Simple LED circuit connected to a Raspberry Pi.

EXPERIMENTAL RESULTS

While a subject is walking towards the smart office, the Kinect sensor will detect and start to capture the subject's gait. A series of recognition processes will take place immediately to identify the subject. The result will be displayed in a monitor screen outside the smart office. The monitor screen is connected with the Raspberry Pi. The interface is implemented using Tkinter in Python which is named as "accessUI.py". The figures below illustrate the two kinds of recognition results displayed in the monitor screen:

Figure 23. Result of recognition displayed in a monitor screen.

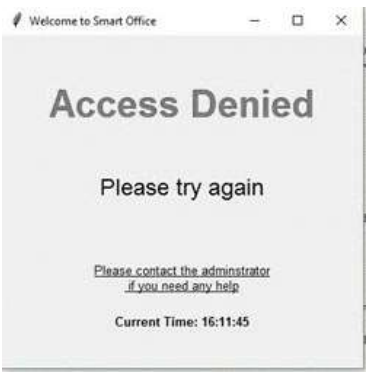


Figure 24. Result of recognition displayed in a monitor screen.

CONCLUSION AND FUTURE WORK

This study develops a solution for seamless biometric for smart office using gait recognition. A complete analysis and solution for the proposed seamless biometric system is presented. The biometric system can recognize users through their natural walking behavior without having to interrupt the users in their daily routine.

There are several improvements that can be made in this work. The system may be enhanced as one to many controls: one admin (server) is able to control multiple office rooms. Besides that, some additional features and functions such as temperature control and light sensor can be added to make the system eco-friendly. An alternative biometric modality like face recognition could also be added to enhance the recognition accuracy.

REFERENCES

Adini Y, Moses Y, & Ulman S. (1997). Face Recognition: The Problem of Compensating for Changes in Illumination Direction. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pp. 721-732.

BIOCAM300. (2017). Retrieved from *ZKAccess*: http://www.zkaccess.com/ProductDetail.aspx/?cat=Standalone+Reader+Controllers&series=Standalone+Reader+Controllers&product=BioCam+300-+New+Product.

Biometrics. (2017). Retrieved from *Wikipedia*: https://en.wikipedia.org/wiki/Biometrics.

Biometrics - Modalities. (2017). Retrieved from *tutorialspoint*: http://www.tutorialspoint.com/biometrics/biometric_modalities.htm.

Bojan Dikovski, Gjorgji Madjarov, Dejan Gjorgjevikj. (2014). Evaluation of different feature sets for gait recognition using skeletal data from Kinect. Information and Communication Technology, Electronics and

Microelectronics (MIPRO), *2014 37th International Convention*. MIPRO.

Dredge, S. (2014, September 17). *10 things you need to know about biometrics technology*. Retrieved from the guardian: http://www.theguardian.com/technology/2014/sep/17/10-things-to-know-about-biometrics.

DriverTrack™. (2017). Retrieved from M2SYS: http://www.m2sys.com/cloud-based-driver-license-identity-management-system-driver-track/.

Earl, B. (2017). *What is a Stepper Motor?* Retrieved from adafruit: https://learn.adafruit.com/all-about-stepper-motors/what-is-a-stepper-motor.

FAQ - Gait Biometrics. (2017). Retrieved from 360Biometrics: http://www.360biometrics.com/faq/Gait-Biometrics.php.

Gait Recognition. (N.D.). Retrieved from Global SECI: http://globalseci.com/?page_id=44.

Gee, M. (2015, October 14). *1-in-4 Internet users have account hacked this year: Kaspersky Lab.* Retrieved from ARN: http://www.arnnet.com.au/article/586682/1-in-4-internet-users-account-hacked-year-kaspersky-lab/.

Goh M., Connie T., & Teoh A. (2010a). An Innovative Contactless Palm Print and Knuckle Print Recognition System. *Pattern Recognition Letter*, 1708-1719.

Goh M., Connie T., & Teoh A. (2010b). Locating Geometrical Descriptors for Hand Biometrics in a Contactless Environment. *International Symposium on Information Technology*. Kuala Lumpur, Malaysia.

Gonser, T. (2016, January 3). *5 Things That Will Disappear In 5 Years.* Retrieved from TechCrunch: http://techcrunch.com/2016/01/03/5-things-that-will-disappear-in-5-years/#.haecvhl:4TD4.

HIKVISION DS-2CD4132FWD-IZ. (2017). Retrieved from Top Ten Reviews: http://facial-analyzer-software-review.toptenreviews.com/hik-vision-review.html.

IRIS ID. (2017). Retrieved from IRIS ID: http://www.irisid.com/.

Kinect. (2017). Retrieved from *Wikipedia*: https://en.wikipedia.org/wiki/Kinect.

Munson, L. (2014, October 17). Average person has 19 passwords – but 1 in 3 don't make them strong enough. Retrieved from *naked security* by SOPHOS: https://nakedsecurity.sophos.com/2014/10/17/average-person-has-19-passwords-but-1-in-3-dont-make-them-strong-enough/

NITGEN (2017). Retrieved from *NITGEN*: http://www.nitgen.com/eng/index.html.

Raspberry Pi. (2017). Retrieved from *Wikipedia*: https://en.wikipedia.org/wiki/Raspberry_Pi.

Reddy, S. (2014, April 14). Why We Keep Losing Our Keys. Retrieved from *The Wall Street Journal*: http://www.wsj.com/articles/SB10001424052702304117904579501410168111866.

Rouse, M. (2009, May). Systems development life cycle (SDLC). Retrieved from *Search Software Quality*: http://searchsoftwarequality.techtarget.com/definition/systems-development-life-cycle.

Trader, J. (2013, September 24). 5 Ways Biometric Technology is Used in Everyday Life. Retrieved from *M2SYS Blog*: http://blog.m2sys.com/guest-blog-posts/5-ways-biometric-technology-is-used-in-everyday-life/.

Trader, J. (2014, November 7). Looking Ahead: What New Biometric Modalities are on the Horizon? Retrieved from *M2SYS:* http://blog.m2sys.com/biometric-hardware/looking-ahead-new-biometric-modalities-horizon/.

Trader, J. (2015, December 7). Biometric Lock Technologies For Smart Homes: Where They Stand Today, And Where They're Going Tomorrow. Retrieved from *M2SYS*: http://blog.m2sys.com/category/future-of-biometrics/.

Waterfall model. (2017). Retrieved from *Wikipedia*: https://en.wikipedia.org/wiki/Waterfall_model.

What are biometrics? (n.d.). Retrieved from *findbiometrics:* http://findbiometrics.com/what-are-biometrics/.

What is Biometrics? (2017). Retrieved from *Biometric Time Clock System*: http://biometrictimeclock.com/what-is-biometrics/.

In: Security and Authentication
Editors: Ong Thian Song et al.
ISBN: 978-1-53612-942-7

Chapter 11

HIDING INFORMATION WITHIN A QR CODE BASED ON A COLOR SUBCELL (SQRC)

Ari Moesriami Barmawi* *and Yudha Viki Alvionata*
Informatics Graduate Program, School of Computing,
Telkom University, Bandung, Indonesia

ABSTRACT

A QR code is usually used for coding identity or other specific data. The information contained in the QR code is a feature of the owner such as a fingerprint or other biometric feature. Nowadays, the use of QR code is gaining popularity, especially in business, banking and government, because of the high storage capacity compared to a conventional barcode, fast code reading and the capability to be read by the majority of smart-phone devices. Since QR code is frequently used for coding identity, it should prevent against ID forgery. To overcome the problem, Wisdarmanto et al. (Sukegawa et al., 2008) proposed a method for hiding biometric information in QR code. However, the capacity of their method was not sufficient for hiding high-quality biometric features. To increase the capacity, implementing color subcell QR code was proposed. The method used RGB color combination for encoding the information which

* Corresponding author, Email: mbarmawi@melsa.net.id

was going to be hidden in the subcell of the QR code. Based on the experiments, it was shown that the capacity of QR code using the proposed method was greater than using Wisdarmanto's method.

Keywords: subcell, QR code, hiding information, capacity

INTRODUCTION

Along with the development of the Internet, our society constantly finds various ways to communicate with each other using electronic media. In certain applications such as in banking, confidential information exchange between authentic parties is necessary. One information type is QR code.

Since QR code is frequently used for coding identity, ID forgery should be prevented. To protect QR code from forgery attack, a unique feature such as a biometric feature should be sent along with the QR code while the QR code can be directly read using its reader. To send the owner's biometric feature along with the QR code, information hiding is introduced. Thus, to secure the QR code against a forgery attack, the owner's biometric feature is embedded in the QR code. Using an information hiding technique, the embedded QR code can be directly read by the reader.

By embedding the biometric feature data in QR code, it becomes a single entity as an inseparable integrated part (Barmawi et al., 2015). Thus, the embedded information (which in this case is biometric feature data) can be used to authenticate the user. However, as there is a limitation in QR code data capacity, the size of information which can be hidden is limited as well. To overcome this problem, some researchers (Furumoto et al., 2012; Sukegawa et al., 2008; Kikuchi et al., 2013; Teraura et al., 2015) have tried to increase the capacity of QR code such that more data can be embedded, or by finding a better method to hide more data into it as in (Barmawi et al., 2015; Teraura et al., 2012; Lin et al., 2013; Bui et al. 2013; Erlangga et al., 2016).

Numerous researches of QR code capacity have been proposed such as (Barmawi et al., 2015; Sukegawa et al., 2008; Kikuchi et al., 2013; Teraura et al., 2015; Teraura et al., 2012; Lin et al., 2013; Bui et al., 2013), even so the QR code capacity using Wisdarmanto's method is the highest compared to the previous works. However, its capacity was still insufficient to hide high-quality biometric features (such as the fingerprint feature extraction proposed by Zhang et al. (2013). To increase the capacity, in this work, color subcell QR code was introduced. The basic idea was based on color palette subcell. At the beginning, the embedded code was divided into two groups, the white group and the black group. Next, the subcell was examined; if the center part was black, then to embed a message in the intended cell, the black group data should be used and vice versa. Each data consisted of two bits. Each subcell can be embedded by 8 two bits data. Using the color subcell method, the capacity of the QR code can be extended to twice that of the extended subcell QR code (Erlangga et al., 2016) while maintaining the error correction capability.

EXTENDING QR CODE CAPACITY USING SUBCELL

This section discusses Subcell QR code and Extended Subcell QR code including its drawbacks.

Subcell QR code

Teraura and Sakurai (Teraura et al., 2012) proposed 3 × 3 subcell QR code that can carry additional confidential secret information using five subcells structures. Each subcell is a square with the same size and ratio. Further, when the subcell was captured by a shaky camera, the border becomes blurred. Therefore, it is difficult to distinguish one pixel from another. Since the border area is considered as a pointer for distinguishing one cell from another, the outer subcell on the cell border are assigned the same color as the cover cell, and the subcell in the center is assigned the opposite color. The layout of 3 × 3 subcell is shown in Figure 1.

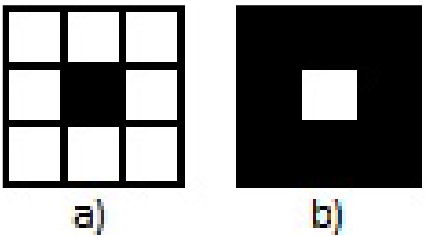


Figure 1. Embedding pattern for 3 × 3 subcells configuration. a) White cell, contained binary "1" secret data, b) Black cell, contained binary "0" secret data.

The center subcell is the peripheral (main) part and the outer subcell is the same color as the cover cell. As the center part contains 1-bit data, it can store a 1-bit secret data of the same size as its cover cell. The data layout for a 3 × 3 subcell QR code consists of two identical sized data in the form of a complete QR code symbol. Each QR code symbol has the same version, error level, format and masking pattern. Thus, if secret data is embedded in a code block $u = (u_0, u_1)$, then the color of the outer subcell level represents the cover data u_0, and the color of the center subcell level represents the embedded data u_1.

Extended Subcell QR code (EQRC)

Conversely to the subcell QR code method, in the Extended Subcell QR code, the outer subcell (as shown in Figure 2) is the embedded data, while the center is the cover data.

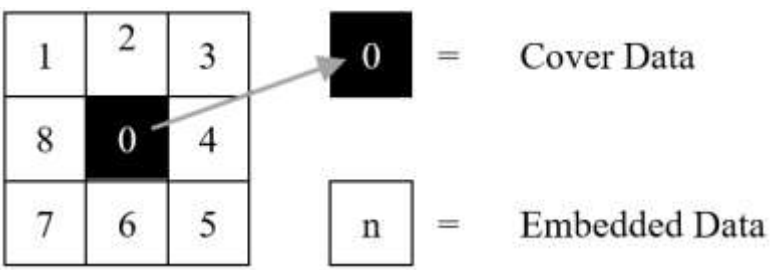


Figure 2. Encoding pattern for Extended Subcell QR code.

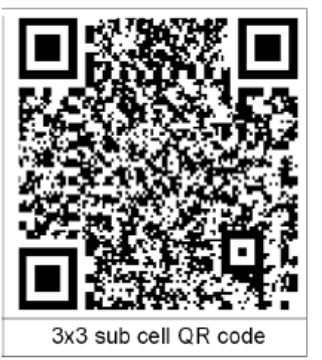
Each bit allocation for these eight additional sets data was randomized. The number of permutations of the data structure is expressed in equation (1),

$$p(n,k) = \frac{n!}{(n-k)!} \quad (1)$$

where n is the number of embedded data and k is the number of secret messages. Suppose the number of embedded data is 8 and the number of subcells that can be embedded by the secret data is 8, then there are 40,320 structure permutations available in each 3 × 3 subcell. Further, randomizing the embedded data location is needed to camouflage the secret data. To randomize the embedded data location in the subcells, a pseudo-random number generator CTR DRBG (Keller & Hall, 2015) is used.

The process of placing the final message data block for each embedded data on a subcell area relied on the CTR DRBG Pseudo Random Generator. For conducting CTR DRBG, key and seed were necessary. Therefore, those parameters should be kept secret by both parties. For securing the seed and the key, secure key distribution such as Diffie-Hellman key exchange could be used. The result of the embedding process using CTR DRBG Pseudo Random Generator is shown in Figure 3.

(a)



(b)

Figure 3. 4L version QR code. (a) Original QR code. (b) Embedded QR code.

EXTENDING QR CODE CAPACITY USING COLOR PALETTE SUBCELL

The basic idea of color subcell QR code (SQRC) is to extend the capacity of QR code using a colored subcell. Similar to EQRC, in SQRC the cell is extended into 9 subcells. The outer subcells are the embedded data, while the center cell (usually called the center part) is the cover data. However, the basic difference between EQRC and SQRC is that SQRC uses color (red, green, blue) for data grouping. There are two data groups used in SQRC, the white group and the black group. The white group consists of a white center part and other colored outer cells, while the black group consists of a black center part and a colored outer cell. The data-colored mapping for both white and black groups is shown in Table 1.

The RGB color grouping was conducted based on the Euclidean distance between RGB values which was used as a rating scale. In this rating scale, identifying the color is easiest in cases with a combination of colors that are mutually separated in a three-dimensional RGB space. Then, the most distant mutual position, which is at the end of the cube in RGB space, was chosen as the color in the concerned position. The concrete value in the RGB space for each selected color is shown in Table 1. Further, the brightness of each color is identified in Table 1. Equation (2) was specified by ITU-R BT.601 (2011) and was used for translating the brightness (Y) and an RGB value. R, G and B were the intensity of red, green, blue respectively.

$$Y = (0.299R + 0.587G + 0.114B)/255 \quad (2)$$

Using this approach, the capacity would be doubled compared to the extended QR code (Sukegawa et al., 2008). Figure 4 shows the conversion from EQRC into SQRC.

Table 1. Data encoding using Color-based data grouping [5]

White Group						Black Group					
Data	R	G	B	Y	Color	Data	R	G	B	Y	Color
00	255	255	255	1		00	255	0	255	0.41	
01	255	255	0	0.89		01	255	0	0	0.3	
10	0	255	255	0.7		10	0	0	255	0.11	
11	0	255	0	0.59		11	0	0	0	0	

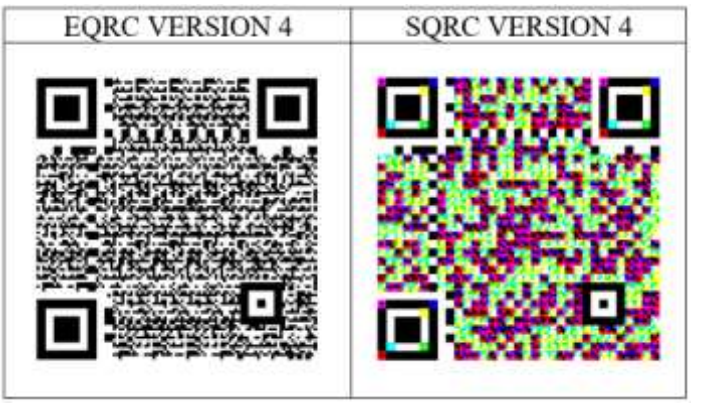


Figure 4. Conversion from EQRC into SQRC.

Encoding Process

In the encoding process, there were three main sub-processes: creating 3 × 3 subcell QR code, subcell randomization, and multicolor encoding. The overview of the encoding process is shown in Figure 5.

There were three inputs in encoding process, the cover data (main data), the secret message and the seed. The encoding process began by generating a standard QR code of the cover data, then continued by creating 3 × 3 subcell QR code. For creating 3 × 3 subcell QR code, Teraura's method (Teraura & Sakurai, 2015) was used. This method was discussed in the Subcell QR code section. Next, the 3 × 3 subcell QR code was used for generating SQRC by implementing the results of subcell randomizing and the multicolor encoding process.

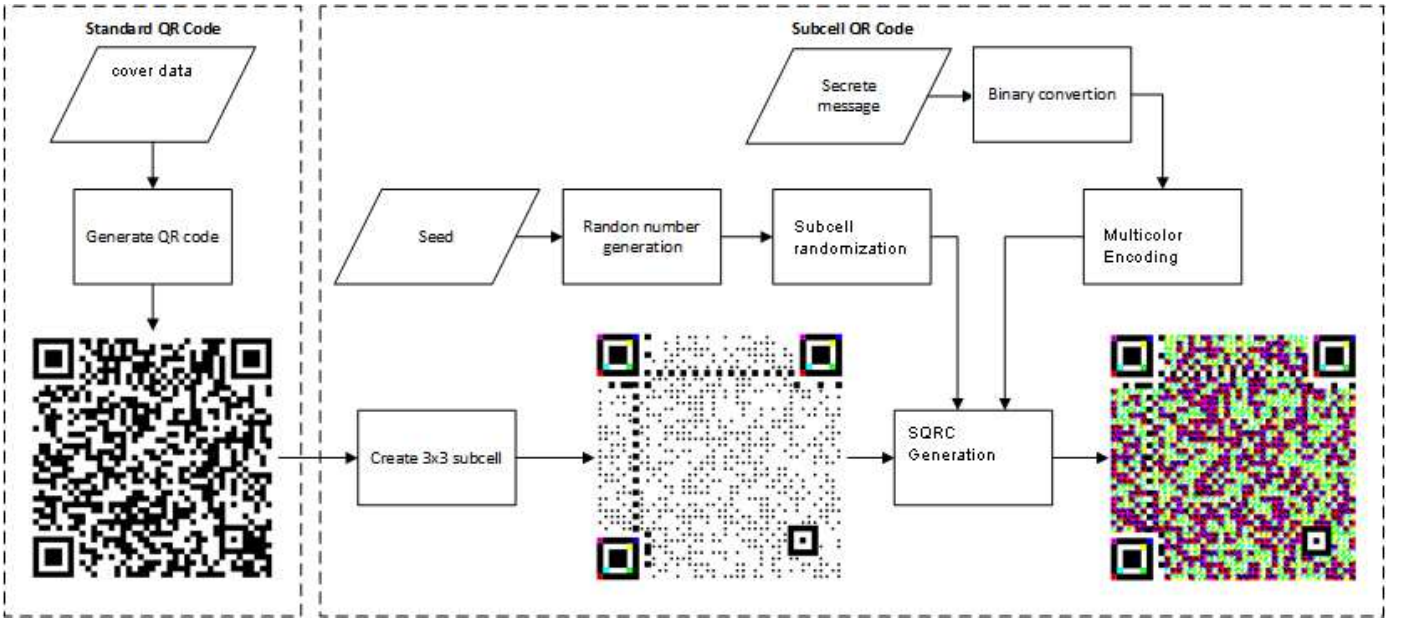


Figure 5. SQRC Encoding Process.

There were two processes conducted at the same time, subcell randomization and multicolor encoding. For randomizing subcell location, random number generation based on the seed was conducted and the output of this process was used to randomize the subcell's location. Meanwhile, in the multicolor encoding process, the binary representation of the secret message was converted into multicolor codes. SQRC was generated from the 3 × 3 subcell QR code using the results of subcell randomization and multicolor encoding processes.

The objective of the randomizing subcell location process was for securing the data in the subcell such that even if an attacker could obtain the content of the subcell, the real data still could not be obtained. This condition occurred because the subcell's location was randomized. As has been discussed, for the randomizing subcell's location, random number generation was used, and then the Blum-Blum Shub (BBS) (Zhang et el., 2010) method was introduced for random number generation. BBS is a random number generation process based on quadratic residue. Suppose there is an integer number n which is the multiplication of two prime numbers p and q (p and q have to be congruent to 3 mod 4), and another integer s which is a member of quadratic residue modulo n ($QR(n)$) or in other words $s \in QR(n)$ (Zhang et el., 2010). Further, s is used as the seed. The random number is generated using equation (3),

$$x_i \equiv x_{i-1}^2 \bmod n \tag{3}$$

where i is the index of the number and x_i is the i^{th} random number generated by the process. For generating the first random number, a seed s would be used as x_0 such that $x_1 \equiv x_0^2 \bmod n \equiv s^2 \bmod n$. Since x_i could be greater than 8, while the number of subcells was only 8, then for randomizing the subcells, a number y_i was necessary where y_i was calculated using equation (4). The process of random number generation for $s=2$ is shown in Table 2.

$$y_i \equiv x_i \bmod 9 \tag{4}$$

Table 2. Random number generation using Blum-Blum Shub method

i-th rotation	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_i	4	16	256	423	196	397	289	54	294	347	234	131	118	377	104
x_i (mod 9)	4	7	4	0	7	1	1	0	6	5	0	5	1	8	5
i-th rotation	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
x_i	328	82	169	156	301	142	62	348	55	403	282	427	100	386	416
x_i (mod 9)	4	1	7	3	4	7	8	6	1	7	3	4	1	8	2

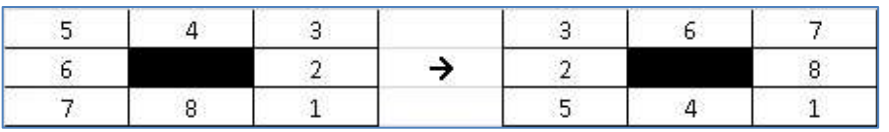


Figure 6. Randomizing subcell's location.

Further, the subcell's location is randomized based on y_i. An example of the randomizing process result is shown in Figure 6.

The multicolor encoding process was conducted by implementing the mapping as shown in Table 1. Suppose the secret message is *0011 1100 0100 1010*, and the subcell location is not randomized, then the result of the multicolor encoding process is as shown in Figure 7.

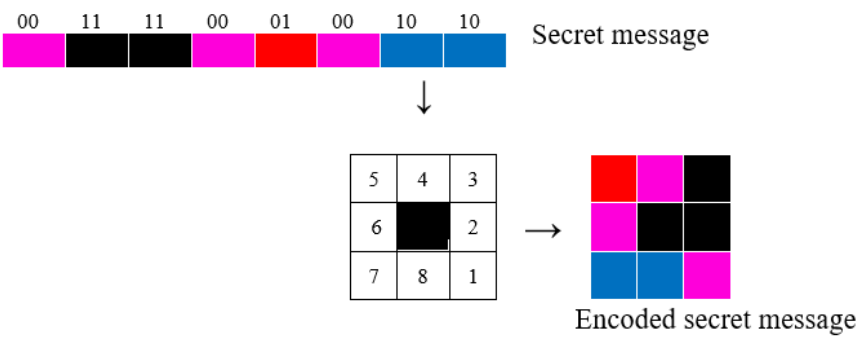


Figure 7. Multicolor encoding result for Black Group.

Figure 8. Multicolor encoded subcell after randomizing.

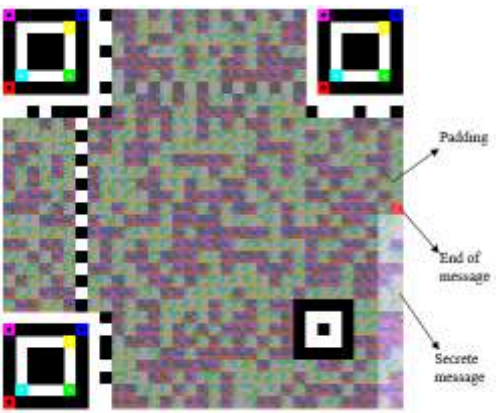


Figure 9. Detailed structure of SQRC.

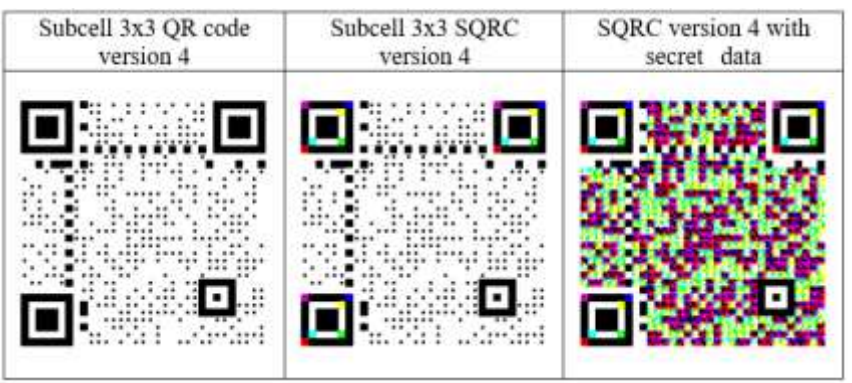


Figure 10. Generating SQRC from 3 × 3 subcell QR code.

The result of multicolor encoding is randomized based on the number obtained from the BBS random number generator. Suppose the results of BBS random number generation is 1, 6, 5, 8, 7, 4, 3, 2, furthermore the result after randomizing process is shown in Figure 8.

Suppose the message size was less than the number of the codewords, then the secret message should end using a double space, and the remaining codewords were filled by random padding data. The padded data was introduced for camouflaging the embedded data. Figure 9 shows the padding data (the dark shadowed area), the secret message (the bright area) and the end symbol which is orange. Since the end symbol was a symbol which was only agreed upon by both encoder and decoder, then it should not be easily recognized by adversaries.

Finally, the SQRC was generated by implementing the subcell location randomizing process and multicolor encoding. The resulting SQRC is shown in Figure 10, and the detailed SQRC with the location of the secret message and padding data is shown in Figure 9.

The SQRC could be used by assuming that the seed could be agreed upon by both the encoder and the decoder, and the modulus number n was public.

Decoding Process

The decoding process consisted of four sub-processes: converting the luminance of the subcells into binary codes (data), generating a random number, sorting the binary data based on the random number and finally converting the binary data into a hexadecimal number for obtaining the secret message. The overview of the decoding process is shown in Figure 11.

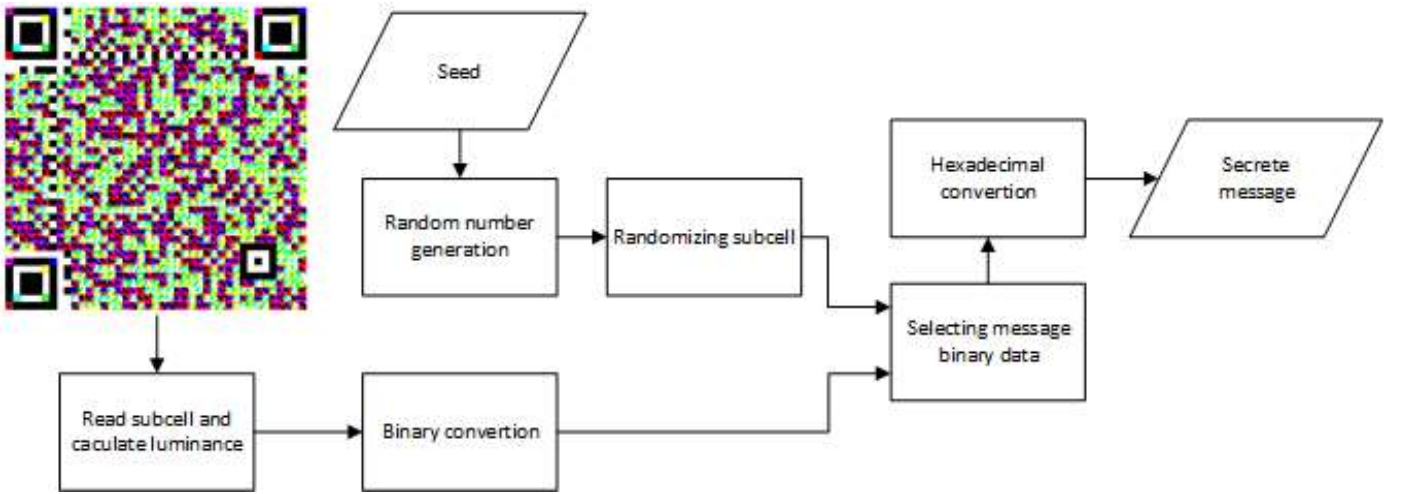


Figure 11. Decoding process of SQRC.

The decoding process began with scanning/reading the QR code, then converting the luminance of the color QR code into binary data using Table 1. At the same time, the decoder should generate the random number using BBS (Zhang et al., 2010) method similar to the one that the encoder did. Further, the decoder selected the binary data based on the random number. Finally, the binary data was converted into hexadecimal and translated into ASCII Code for obtaining the characters of the message. The example of the decoding process is shown in Figure 12.

Suppose the SQRC cell that has to be decoded is as shown in Figure 12, then the color should be aligned in one row based on the subcell's number. Further, the content of each subcell should be converted into binary number based on Table 1. At the same time the decoder generates the random number using BBS and obtains *1, 8, 7, 6, 3, 2, 5, 4*. The random number is used to rearrange the sequence of the subcells and continues by converting the binary number of each subcell's content into a hexadecimal number. Finally, the secret message is obtained by converting the hexadecimal number into ASCII code.

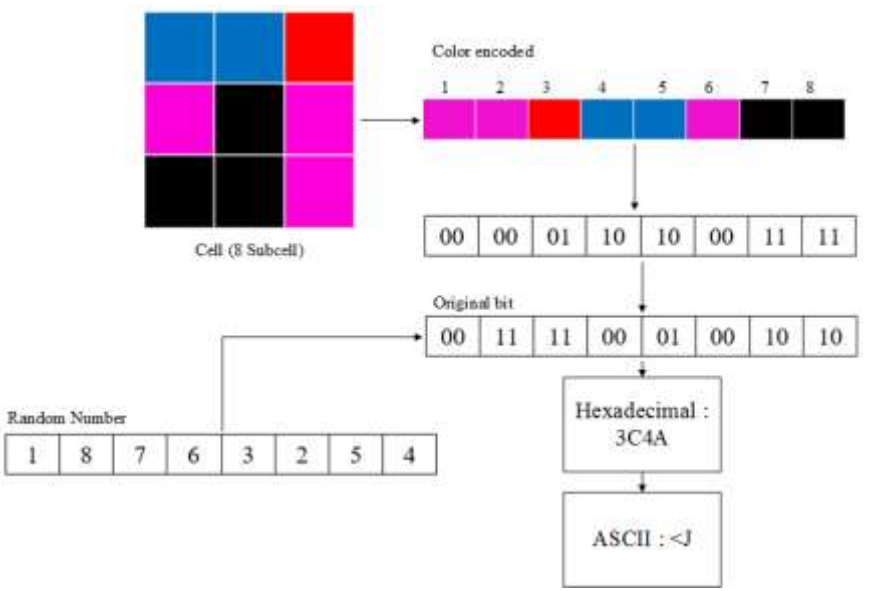


Figure 12. Example of SQRC decoding process.

CAPACITY AND PERFORMANCE EVALUATION OF COLOR SUBCELL METHOD

For evaluating the SQRC performance, three aspects were observed: the capacity, the processing time and the error correction capability. For evaluating the capacity of SQRC, several calculations and experiments were conducted such as encoding specific data using type 1-6 version of SQRC and EQRC. The data capacity using both EQRC and SQRC is shown in Figure 3.

Since the EQRC has 8 subcells, where each subcell could be filled by 1 bit, while in the case of the SQRC each subcell could be filled by two bits then the capacity of the SQRC was twice that of the EQRC capacity as shown in Table 3. However, due to camera limitations, the data that could be read by the camera was less than the capacity of the SQRC. This condition occurred because of using color subcells, the border between the colors became blurred when the number of the sub pixel was large enough. Table 3 shows that the readable data could be less than its capacity.

As the impact of capacity was extended, the encoding and decoding processing time was increased. This condition occurred because besides the main data (the center of the subcells), the system should encode other 8 subcell data. However, the encoding processing time was not dependent on the amount of data that was embedded in the subcells, since even if the message was short, the subcell in the remaining codeword's should be filled by padded data. As the consequence, even if the message was short, all codewords should be encoded.

Suppose type-1 QR code consists of main data "UNDANG-UN" and is embedded by "Loremipsum dolor sit amet, consecteturadipiscingelit. Aenean.," which is further encoded as SQRC1. Further, the QR code was also embedded with "1234567890abcdefghijklmnopqrtuvwxyz ABCDEFGHIJKLMNOPQRTUVWXYZ" and called SQRC2 as shown in Table 4. Based on Table 4, it is shown that even if the secret message embedded in SQRC1 is less than the secret message embedded in SQRC2,

the encoding processing time is equal. Thus, it is proven that the encoding time is not dependent on the size of the secret message. This condition occurred in all versions of the QR code. This phenomenon is shown in Figure 13 concerning the embedding data processing time.

In contrast to the encoding process, the decoding process was dependent on the size of the secret message because the decoding process was stopped after the secret message had been decoded. The padding data should not be decoded at all because it was unused. Based on Table 4, it is shown that the decoding processing time of SQRC1 is always less than SQRC2. Finally, it is proven that the decoding processing time depended on the size of the secret message. This phenomenon is shown in Figure 14.

Table 3. Readable Capacity between EQRC and SQRC

NO	QR code Version	QR code type	data Capacity	Readable Data
1	1H	EQRC	448	448
		QR code (BW)	56	56
		SQRC	896	896
2	2H	EQRC	896	896
		QR code (BW)	112	112
		SQRC	1.792	1,664
3	3H	EQRC	1.536	1.536
		QR code (BW)	192	192
		SQRC	3.072	2176
4	4H	EQRC	2.176	2.176
		QR code (BW)	272	272
		SQRC	4.352	2688
5	5H	EQRC	2.816	2.816
		QR code (BW)	352	352
		SQRC	5.632	3200
6	6H	EQRC	3.712	3.712
		QR code (BW)	464	464
		SQRC	7.424	3812

Table 4. Encoding and decoding processing time for embedded and original QR code for type 1 to type 6

QR code version	QR code type	Secret Message	Processing time (s)	
			Encoding	Decoding
1	SQRC 1	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	1.010	0.031
	EQRC	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	1.020	0.041
	SQRC 2	1234567890abcdefghijklmnopqrtuvwxyz ABCDEFGHIJKLMNOPQRTUVWXYZ	1.010	0.033
	Standard	-	0.820	0.007
2	SQRC 1	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	1.100	0.037
	EQRC	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	1.110	0.066
	SQRC 2	1234567890abcdefghijklmnopqrtuvwxyz ABCDEFGHIJKLMNOPQRTUVWXYZ	1.100	0.040
	Standard	-	0.910	0.007
3	SQRC 1	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	1.120	0.032
	EQRC	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	1.140	0.087
	SQRC 2	1234567890abcdefghijklmnopqrtuvwxyz ABCDEFGHIJKLMNOPQRTUVWXYZ	1.120	0.036
	Standard	-	0.950	0.008
4	SQRC 1	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	2.000	0.029
	EQRC	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	2.010	0.109
	SQRC 2	1234567890abcdefghijklmnopqrtuvwxyz ABCDEFGHIJKLMNOPQRTUVWXYZ	2.000	0.042
	Standard	-	1.000	0.009
5	SQRC 1	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	2.140	0.030
	EQRC	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	2.150	0.141
	SQRC 2	1234567890abcdefghijklmnopqrtuvwxyz ABCDEFGHIJKLMNOPQRTUVWXYZ	2.140	0.033
	Standard	-	1.210	0.010

Table 4. (Continued)

QR code version	QR code type	Secret Message	Processing time (s)	
			Encoding	Decoding
6	SQRC 1	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	2.980	0.031
	EQRC	Loremipsumdolorsitamet, consecteturadipiscingelit. Aenean.	3.000	1.156
	SQRC2	1234567890abcdefghijklmnopqrtuvwxyz ABCDEFGHIJKLMNOPQRTUVWXYZ	2.980	0.034
	Standard	-	1.300	0.013

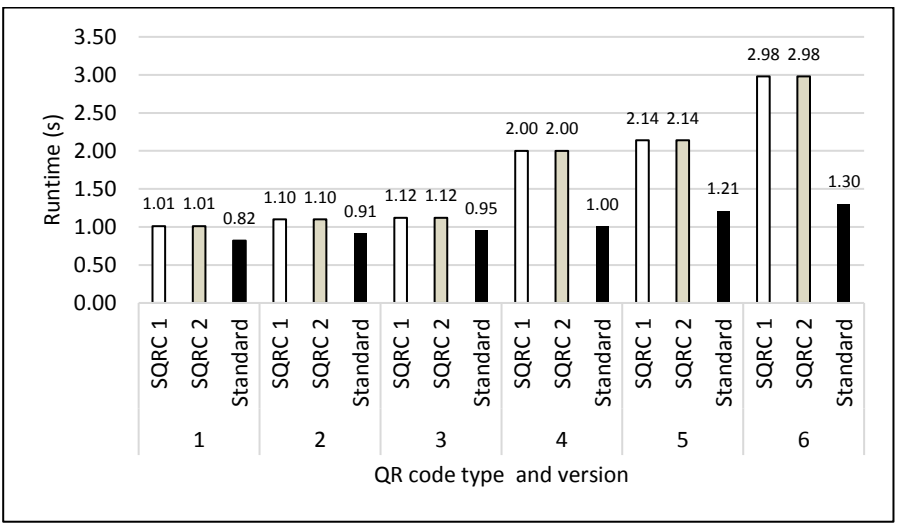


Figure 13. Processing time of QR code encoding.

Since one of the important aspects was error correction, error correction capability should be evaluated. Basically, since in SQRC there were no changes in the error correction area, then the error correction capability was equal to that in a standard QR code. For proving this claim, experiments were conducted by destructing the SQRC and then decoding it using a cellphone camera with ZXing software. Based on the experiment results shown in Table 5, it can be concluded that all noisy QR code could be recovered such that the readings result were similar to the original main data.

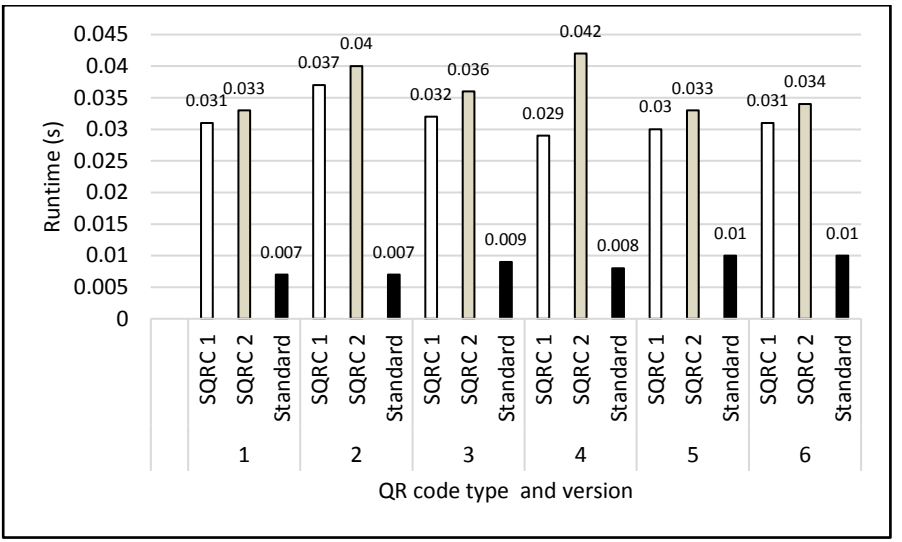


Figure14. QR code Decoding Runtime.

Table 5. Recovering capability of SQRC for type 1 to 6

QR code type	Main data	Secrete data	SQRC	Dirty SQRC	Reading result
1	0001-YUDH	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean			0001-YUDH
2	0001-YUDHA VIKI ALV	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean			0001-YUDHA VIKI ALV
3	0001-YUDHA VIKI ALVIONATA-COMPUTER	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean			0001-YUDHA VIKI ALVIONATA-COMPUTER

CONCLUSION

The conclusion was arrived at by observing three aspects: capacity, runtime, and error correction. According to the experiments' results and the discussion, it can be concluded that the capacity of the SQRC can be extended to twice of the EQRC capacity. However, since the use of the 3 × 3 color subcell may blur the border between subcells, the subcell reading might fail especially if a low resolution camera was used. In this research, using 8MP camera, the maximum subcell reading for type-6 QR code was about 50% from the extended capacity. This shows that the larger the type number, the smaller the percentage of readable data. However, from type-1 to type-6 QR code the capacity of the SQRC was still higher than the capacity of the EQRC.

The processing time of the SQRC encoding was about twice of the standard QR code one, especially for the higher type. Meanwhile, for the lower type, the encoding processing time was about 1.2 times that of the standard QR code, but still similar to the EQRC encoding time. This condition occurred due to the impact of the extension process. In the case of the SQRC decoding, the processing time was greater than the standard one but similar to the EQRC one. Although the encoding and decoding processing time using the proposed method were greater than using the standard QR code, but the encoding time was about 2 seconds, and the decoding time was less than 1 second. Thus, it is still implementable.

Concerning the message recovery capability based on the error correction, it can be concluded that the main data was still readable even when the subcell extension was implemented into the QR code. Finally, this indicates that SQRC performance is better than EQRC in capacity and message recovery, but not in encoding and decoding processing time.

REFERENCES

Barmawi, A. Moesriami, and Yulianto, F. Arif. (2015). "Watermarking QR Code." Paper presented at *IEEE 2nd International Conference on Information Science and Security (ICISS).* Seoul, Korea, December 14-16.

Boneh, D. (2011) "Blum-Blum Shub Pseudorandom Bit Generator", In *Encyclopedia of Cryptography and Security*. Edited by Henk C.A. Tilborg and Sushil Jajodia. 160-161. Springer.

Erlangga, W., and Barmawi, A. M., (2016). *"Increasing Secret Data Hiding Capacity in QR Code Using 3x3 Subcells."* Master thesis. Telkom University.

Furumoto, K., Watanabe, Y., and Morii, M. (2012). "Grey scale two-dimensional code and its applications", *IEICE technical report. Information and communication system security*, vol. 112, 315:7-12.

Keller, S., Hall, T. A. (2015). "The NIST SP 800-90A Deterministic Random Bit Generator Validation System (DRBGVS)." *National Institute of Standards and Technology, Information Technology Laboratory*, Computer Security Division. NIST.

Kikuchi, M., Fujiyoshi, M. and Kiya, H. (2013). "A New Color QR Code Forward Compatible with the Standard QR Code Decoder." Paper presented *at IEEE International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS).* Okinawa, Japan, November 12-15.

Lin, Pei-Yu., Chen, Yi-Hui., Lu, E., Jui-Lin, and Chen, Ping-Jung. (2013). "Secret Hiding Mechanism Using QR Barcode." Paper presented at *IEEE International Conference on Signal-Image Technology & amp; Internet-Based Systems (SITIS)*, Kyoto, Japan, December 2-5.

Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, *Recommendation ITU-R BT.601-7*, 2011, ITU-R Radio Communication Sector of ITU, International Telecommunication Union.

Sukegawa, S., Ito, M., Kondo, K., Ozono, T., and Shintani, T. (2008). "A High Capacity Multicolored 2d Barcode Based on QR Code". *National Convention Papers*, vol. 70:845-846.

T. V. Bui, N. K. Vu, T. T. Nguyen, I. Echizen, and T. D. Nguyen, 2013, "Robust message hiding for QR code", *The Tenth IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP).* Beijing, China, December 2-5.

Teraura, N., and Sakurai, K. (2012), "Information Hiding in Subcells of a Two-Dimensional Code." Paper presented at the *1st IEEE Global Conference on Consumer Electronics.* Tokyo, Japan, October 2-5.

Teraura, N., and Sakurai, K. (2015). "Proposal of Multi-Value Cell Structure for High-Density Two-Dimensional Codes and Evaluation of Readability Using Smartphones." Paper presented at *7th International Conference on New Technologies, Mobility and Security (NTMS).* Paris, France, July 27-29.

Zhang, D., Liu, F., Zao, Q., Lu, G., and Luo, N. (2010) "Selecting a Reference High Resolution for Fingerprint Recognition Using Minutiae and Pores." *IEEE Transaction on Instrumentation and Measurement*, 60:863-871.

INDEX

#

6LoWPAN, 118, 124

A

abnormal behavior detection, 87, 88, 89, 92, 97, 110, 114
access, 3, 19, 31, 48, 53, 57, 59, 60, 122, 124, 127, 131, 132, 133, 140, 166, 217, 218, 220, 221
advancement(s), 42, 119, 150
algorithm, 10, 11, 15, 22, 24, 27, 28, 29, 30, 37, 38, 43, 44, 45, 82, 93, 94, 95, 113, 129, 222, 224
Android, vi, 10, 12, 139, 140, 141, 149, 150, 151, 152, 153, 154, 155, 158, 159, 160, 161, 169, 170, 177, 178, 179, 180
anomaly detection, 18, 182
artificial intelligence, 14, 87, 88, 89, 113
assessment, 197, 199
association rules, 22
atmosphere, 118
attack, 1, 9, 10, 17, 19, 20, 21, 53, 54, 59, 62, 63, 118, 122, 124, 125, 126, 134, 145, 182, 183, 213, 214, 242
attacker, 17, 18, 19, 24, 44, 45, 53, 59, 124, 143, 249
audit(s), 198, 200, 202
authentication, v, vi, vii, 1, 2, 4, 5, 6, 8, 11, 42, 46, 50, 53, 57, 59, 60, 62, 63, 65, 66, 67, 79, 81, 83, 84, 117, 121, 122, 126, 127, 128, 129, 130, 131, 132, 133, 136, 137, 139, 143, 145, 146, 147, 197, 218, 219, 220, 222
authenticity, 50, 68, 69, 121, 133
authority(ies), 46, 49
automation, 234
automobiles, 154

B

bandwidth, 28, 126, 205
banking, 132, 139, 145, 218, 241, 242
base, 44, 47
behaviors, 92, 99, 101, 103
bioinformatics, 103
biometric(s), vi, 5, 12, 132, 133, 134, 135, 136, 137, 138, 139, 142, 143, 144, 145,

146, 217, 218, 219, 222, 223, 237, 238, 239, 241, 242
Bluetooth, vi, 149, 150, 152, 160, 161, 164, 165, 166, 167, 169, 170, 171, 172, 173, 176, 177, 178, 179
border security, 218
branching, 97

C

C4.5, 14, 26, 28, 29, 32, 36, 37, 38, 212
capacity, 122, 241, 242, 246, 255, 260
certificate, 46, 47, 48, 50, 51, 55, 56, 60, 61
challenges, vii, viii, 41, 56, 62, 214
China, 2, 218, 261
classes, 61, 62, 110
classification, 14, 25, 26, 27, 28, 29, 30, 31, 32, 33, 35, 38, 39, 92, 101, 109, 110, 182, 214
clients, 48, 54, 58, 59
clone, 12, 136, 145
cloud key exchange, 42
clustering, 14, 22, 34
CNN, 87, 109, 110, 114
coding, 67, 73, 241, 242
color, 67, 68, 79, 104, 241, 243, 244, 246, 254, 255, 260
communication, vii, 30, 31, 37, 52, 59, 61, 85, 118, 119, 126, 127, 129, 131, 238, 261, 262
  overhead, 127
compatibility, 127, 128, 129, 157, 178
complexity, 6, 28, 31, 56
computation, 20, 22, 28, 29, 30, 31, 43, 120
computer, 5, 6, 10, 11, 12, 35, 36, 38, 42, 63, 66, 84, 87, 88, 89, 90, 91, 92, 93, 103, 104, 108, 114, 115, 129, 147, 150, 171, 182, 191, 202, 218, 223, 262
  vision, 5, 6, 10, 11, 12, 87, 88, 89, 90, 91, 92, 103, 114, 218
computing, vii, 29, 57, 117
conference, 5, 34, 36, 142, 147
confidentiality, 46, 48, 50, 121
configuration, 59, 82, 202, 244
construction, 22, 23, 29, 183, 184, 218
Convolutional Neural Network, 109
correlation(s), 72, 82, 129, 214
  analysis, 214
cost, vii, 22, 30, 31, 56, 100, 118, 140, 145, 150
counterfeiting, 145
credentials, 122
crimes, 88
criminals, 88
cross-validation, 212, 213
cryptography, 28, 41, 42, 47, 49, 57, 60, 62, 63, 64, 121
cyber-attack, 9

D

data analysis, 21
data collection, 13, 15, 21, 27, 197, 198, 200, 202
data distribution, 27
data generation, 121
data mining, 22, 23, 32, 33, 34, 36, 213
data processing, 256
data set, 14, 18, 19, 21, 24, 26, 27, 28, 29, 30, 31, 33, 215
data structure, 25, 245
data transfer, 122, 129
database, 4, 16, 20, 21, 48, 54, 56, 58, 76, 92, 222, 223, 225, 226, 227, 229, 232, 233
dataset collection, 182
DDoS, 183, 205, 213, 214
decision trees, 22, 37, 38
decoding, 252, 254, 255, 256, 257, 258, 260
deep learning, vii, 23, 37, 87, 88, 93, 103, 104, 105, 108, 109, 110, 113
denial, 20, 122, 214

denial of service attack, 122
detection, 15, 17, 19, 66, 69, 70, 72, 81, 82, 83, 99, 101, 102, 109, 114, 115, 118, 140, 181, 182, 202, 212, 214, 215, 221
  system, 81, 82, 140, 181, 182
  techniques, 101
detection system, vii, 18, 81, 82, 140, 181, 182
detection techniques, 101
digital television, 262
digital watermarking, 66
disaster, 119
disclosure, 21
discretization, 34
diseases, 16
distribution, 27, 41, 42, 50, 57, 62, 93, 94, 95, 99, 245
diversity, 21, 35
DNA, 135, 218
DOS, 118

E

electronic circuits, 234
emergency, 122, 227
encoding, 241, 247, 249, 250, 252, 255, 256, 258, 260, 262
encryption, 9, 30, 38, 39, 42, 43, 44, 45, 49, 58, 61, 63, 118, 121, 126, 127, 129
energy, 234
  consumption, 234
energy consumption, 234
engineering, 19, 23
English language, 88, 116
entropy, 28, 29
Environment(s), 15, 31, 43, 53, 81, 82, 84, 190, 212, 223
environmental factors, 118, 119
Europe, 138
European Commission, 33
European Parliament, 33
European Union, 32, 52, 63
evidence, 66, 137
exponential functions, 43
extraction, 16, 19, 93, 100, 109, 110, 224, 229, 232, 243
eye specular highlight, 65, 74

F

face, 1, 2, 4, 5, 6, 12, 53, 100, 118, 132, 134, 135, 136, 139, 140, 141, 143, 145, 146, 147, 217, 218, 219, 220, 221, 237
  authentication, 4
  recognition, 1, 5, 6, 12, 100, 132, 134, 139, 140, 141, 146, 217, 237
Facebook, 5, 132
facial expression, 5
facial recognition, 2, 5, 222
false positive, 6, 19, 76, 81
family members, 158, 164
FBI, 202, 204
features extraction, 224
financial, vii, 13, 15, 150
financial records, 13, 15
fingeprint recognition, 132
Finger Vein, vi, 131, 132, 133, 137, 138, 143, 144, 145, 146, 147
  Biometric(s), vi, 131, 143, 145
  Recognition, 132, 143
Fingerprint(s), 1, 2, 3, 4, 8, 11, 132, 133, 134, 135, 136, 139, 141, 142, 143, 144, 145, 146, 147, 217, 218, 220, 241, 242, 262
  Recognition, 1, 2, 132, 134, 139, 141, 142, 217
fingerprints, 2, 3, 4, 133, 136, 142, 143, 145, 218
force, 9, 93, 96, 97, 99, 115
forest fire, 118
formation, 74, 75
fruits, 104, 105

G

gait, 133, 218, 222, 223, 224, 225, 226, 229, 230, 231, 232, 233, 236, 237
gait recognition, 218, 222, 223, 237
Galaxy, 7, 12, 139, 140, 142, 143, 146
geometry, 133, 218
Google, 5, 10, 12, 139, 140, 151, 154, 179, 180
GPS, 150, 152, 154, 155, 158, 159, 160, 161, 162, 166, 170, 178, 179, 180
grouping, 246, 247
growth, 13, 118
guardian, 238
guidelines, 48

H

hacking, 9, 191
hand/palm vein, 137
health, 16, 136, 218
hiding information, 242
histogram, 93, 95, 98, 102
human, vii, 14, 23, 68, 88, 89, 92, 93, 101, 114, 132, 133, 134, 136, 143, 145, 154, 218, 224
  body, 132, 134, 136, 143, 145, 224
  values, 23
human body, 132, 134, 136, 143, 145, 224
human values, 23
humidity, 118
hybrid, 98

I

ID3, 14, 26, 27, 28, 29, 30, 38
IDA, 60, 61
identification, 15, 66, 134, 218, 220, 225
identity, 6, 18, 42, 47, 49, 53, 54, 55, 56, 61, 64, 218, 222, 225, 238, 241, 242
identity-based cryptography, 42, 64
Image Interpolation, 72
Image(s), 2, 3, 5, 6, 7, 8, 12, 65, 66, 67, 68, 69, 71, 72, 73, 74, 75, 76, 77, 79, 81, 82, 83, 84, 90, 91, 92, 96, 99, 101, 102, 104, 108, 109, 111, 112, 113, 115, 136, 137, 140, 142, 143, 145, 219, 225, 227
immigration, 2, 219
improvements, 58, 62, 237
individuals, vii, 5, 15, 16, 22
Indonesia, 65, 154, 241
industry, 42, 139, 218
information exchange, 125, 242
information technology, viii
infrastructure, 17, 42, 47
integration, 117, 118, 120, 129, 176
integrity, 46, 47, 59, 66, 74, 75, 77, 79, 121, 125, 128
intellectual property, 150
intelligence, 14, 87, 88, 89, 113
intensity values, 72
interface, 119, 170, 185, 186, 197, 205, 223, 225, 226, 227, 228, 229, 232, 234, 235, 236
interference, 123, 124, 221
intervention, 55
Intrusion Detection Systems, 215
inventors, 45
IoT, v, 117, 118, 119, 123, 124, 126, 127, 128, 129, 130
IP address, 187, 205
iris, 1, 2, 7, 8, 12, 132, 133, 134, 135, 136, 139, 142, 143, 145, 218, 219, 220
Iris Authentication, 7
iris recognition, 134, 136, 139, 142, 143, 145, 219
irises, 143, 145
IRS, 69
issues, vii, 54, 55, 88, 117, 123, 129, 157, 221

J

Japan, 261
Java, 167

K

key distribution, 41, 42, 62, 245
kidnapping, 156
Korea, 13, 32, 261

L

laptop, 2
latency, 58
law enforcement, 218
laws and regulations, 16
leakage, 15, 18, 23, 24, 175
Learning, vii, 5, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 31, 32, 33, 35, 37, 38, 87, 88, 93, 103, 104, 105, 107, 108, 109, 110, 112, 113, 115, 181, 182
  task, 20, 25
learning task, 20
least-squares estimation method, 66
LED, 7, 137, 235
lens, 142, 221
life cycle, 239
lifetime, 120, 125, 187
light, 7, 68, 74, 140, 143, 222, 225, 228, 237
Linux machines, 183
location information, 170
logging, 183, 190

M

machine learning, vii, 5, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 31, 32, 35, 38, 87, 93, 103, 104, 105, 107, 114, 181, 182
magnitude, 99, 102
majority, 28, 154, 241
Malaysia, 1, 13, 41, 87, 117, 131, 149, 154, 180, 181, 217, 238
malware, vii, 9
management, 1, iii, v, viii, 32, 36, 41, 42, 49, 54, 55, 56, 57, 58, 62, 63, 119, 124, 139, 197, 238
manipulation, 65, 66, 67
mapping, 224, 246, 250
matrix, 68
measurement, 77
media, 1, 5, 132, 134, 242
medical, 13, 15, 16, 17, 120, 121, 139
medical history, 13, 15
messages, 50, 51, 52, 53, 55, 121, 169, 245
Microsoft, 58, 63, 139, 140, 141, 142
military, 118, 119, 120, 121, 218
misplacement, vi, 149, 150, 164
mobile authentication, vii, 132
Mobile Device(s), vii, 1, 10, 131, 132, 134, 139, 140, 141, 142, 143, 144, 145, 146, 147, 149, 150, 169, 179, 200
mobile phone, 2, 11, 131, 150
models, 5, 7, 17, 18, 19, 21, 31, 47, 132, 134
modules, 175
modulus, 45, 252
mold, 2
motivation, 24
multidimensional, 129
multimedia, 66
multiplication, 29, 45, 249

N

navigation system, 150
negotiation, 50
network congestion, 125

network-based intrusion detection system, vii, 182
Neural Network, 34, 87, 88, 105, 106, 107, 114
neural networks, 34
neurons, 107
New Zealand, 139
nodes, 29, 105, 118, 120, 124, 125
non-repudiation, 46, 48, 133
NSL, 183
null, 191

O

operating system, 149, 154, 160, 178, 191, 197, 200, 212
operations, 50, 51, 57, 58, 59, 67, 77, 109
opportunities, 145

P

pairing, 63, 149, 150, 161, 164, 170, 171, 172, 176
parameter estimation, 67
participants, 177, 179
password, 1, 8, 9, 42, 53, 54, 59, 61, 63, 131, 132, 152, 153, 155, 160, 162, 164, 166, 171, 177, 178, 197, 227
password-based authenticated key exchange, 42
passwords, 1, 2, 9, 10, 53, 132, 145, 218, 239
pattern lock, 1, 2, 10, 12, 132
pattern recognition, 218
photographs, 3, 145
physical characteristics, 133, 218
pixel intensities reconstruction, 66
platform, 154, 156, 160, 182, 201
predictive coding, 67, 73
prevention, 16, 151, 166, 177, 178
principles, 24, 123
prior knowledge, 43, 61
privacy-preserving, 13, 14, 16, 19, 23, 31, 33, 34, 36, 39
private information, 166
private sector, 219
probability, 4, 104
project, 6, 118, 152, 217, 222, 225
proposed method, 29, 242, 260
protection, vii, 1, 10, 16, 17, 18, 20, 21, 23, 25, 32, 139
prototype, 161
public key infrastructure, 42, 64
public safety, 87, 88
publishing, 21, 34

Q

QR code, vi, 241, 242, 243, 244, 245, 246, 247, 249, 251, 254, 255, 256, 257, 258, 259, 260, 261

R

radio, 119, 124, 220
rating scale, 246
reading, 241, 254, 260
reality, 5, 212
reasoning, 88
recognition, 1, 2, 5, 6, 12, 14, 16, 38, 90, 92, 93, 100, 103, 114, 115, 116, 132, 134, 136, 139, 140, 141, 142, 143, 144, 145, 146, 147, 217, 218, 219, 222, 223, 224, 229, 230, 232, 236, 237, 238, 262
reconstruction, 5, 27, 66, 72, 73, 77, 100, 136, 145
recreation, 136, 145
redundancy, 120
reference image utilization, 66
regression, 22, 32, 33, 39
relevance, 49, 126
reliability, 122, 145, 176

requirement(s), 31, 50, 54, 121, 122, 129, 133, 134, 217
researchers, viii, 2, 3, 4, 5, 6, 29, 83, 142, 242
resilience, 127
resolution, 2, 8, 134, 136, 219, 260
resource availability, 125
resources, 28, 57, 58, 120, 125, 126, 129, 146, 205, 213
response, 33, 121
retina, 133, 218
rings, 176, 177
risk(s), 15, 16, 18, 53, 143, 149
ROC, 80, 81
ROI, 91

S

SaaS, 57, 58
safety, 87, 88, 149
sample design, 158
Samsung, 7, 8, 12, 139, 140, 142, 143
SANS, 198, 200, 202, 204
scope, 50, 118, 153
Seamless, vi, 217, 223
seamless biometrics, 218
secure communication, 118
secure routing protocol, 118
security, v, vii, viii, 1, 2, 5, 6, 8, 9, 11, 12, 18, 19, 24, 29, 32, 33, 35, 36, 37, 42, 48, 52, 57, 58, 59, 60, 61, 62, 63, 64, 84, 88, 117, 120, 121, 122, 123, 127, 128, 129, 130, 131, 139, 141, 142, 143, 145, 146, 149, 150, 151, 155, 160, 161, 179, 180, 183, 191, 196, 200, 201, 202, 213, 215, 217, 219, 221, 227, 239, 261, 262
seed, 245, 247, 249, 252
sensing, 118, 120, 125
sensor, 3, 117, 118, 119, 120, 121, 122, 123, 127, 129, 130, 136, 137, 141, 144, 217, 222, 223, 224, 227, 229, 236, 237
sensor network, 117, 118, 119, 120, 121, 122, 123, 127, 129, 130
sensor nodes, 118, 120
Sensor(s), 2, 3, 91, 117, 118, 119, 120, 121, 122, 123, 129, 136, 137, 141, 142, 144, 145, 217, 222, 223, 224, 227, 229, 236, 237
servers, 53, 54, 57, 58, 63
service provider, 15, 123
services, 57, 62, 141, 191
shape, 68, 69, 74, 91, 104
showing, 15, 132, 157
single authentication, 65, 66, 67, 79, 81, 83
skeleton, 224
skin, 104, 136, 145
Smart Office, vi, 217, 219, 223, 226, 227, 228, 229
SMS, 149, 152, 155, 160, 161, 162, 164, 165, 166, 167, 168, 169, 170, 172, 173, 176, 178, 179
snippets, 167
society, 88, 242
software, vii, 2, 4, 5, 6, 9, 17, 48, 142, 167, 175, 197, 238, 258
solution, 14, 20, 21, 22, 31, 55, 132, 137, 138, 143, 145, 149, 151, 153, 160, 178, 217, 222, 237
space-time, 93, 95, 96
specifications, 48, 55, 62
speech, 14, 22, 103, 218
  processing, 22
speech processing, 22
splining operation, 76, 83
spoofed, 1, 136, 142
spoofing, 132, 136, 146, 147
  attack, 140, 143
spoofing, 134
stability, 177
standard deviation, 72, 77, 78
standard QR code, 247, 258, 260
state, 2, 5, 12, 14, 17, 87, 92, 107, 108, 224
storage, 57, 58, 131, 221, 241

structure, 25, 66, 245, 251
style, 121
subcell, 241, 242, 243, 244, 245, 246, 247, 249, 250, 251, 252, 254, 255, 260
subscribers, 15
success rate, 10
supplier, 221
surveillance, vii, 6, 91, 92, 93, 218, 221

T

target, viii, 18, 19, 82, 83, 205
targeted image forensics, vii, 66
taxonomy, 57
techniques, 1, 2, 5, 28, 29, 31, 33, 34, 36, 41, 89, 91, 93, 114, 121, 132, 136, 139, 143
technological advancement, 119
Technology(ies), vii, viii, 2, 5, 14, 15, 16, 21, 24, 25, 42, 62, 89, 117, 132, 149, 150, 153, 162, 178, 179, 217, 218, 219, 238, 239
temperature, 118, 237
terrorism, 88, 114
test data, 19, 214
testing, 20, 38, 69, 87, 105, 111, 153, 175, 191, 212, 232
texture, 5, 82, 91, 104
Theft, 151, 160, 161, 162, 164
threats, 16, 53, 182
training, 14, 15, 17, 18, 19, 20, 22, 23, 24, 30, 31, 87, 100, 105, 109, 110, 111, 112, 113, 114, 212, 232
traits, 133, 134
transactions, 9, 132, 145
transformations, 54
translation, 103
transmission, 46, 120, 121, 126
transparency, 24
transport, 42
  layer security, 42
transportation, 218
triggers, 221, 225

U

United States (USA), 9, 32, 34, 36, 84, 154, 218
universality, 145
user, 4, 8, 10, 21, 23, 24, 26, 47, 48, 49, 53, 59, 60, 69, 122, 123, 127, 131, 132, 133, 134, 136, 140, 143, 144, 152, 155, 156, 158, 161, 164, 165, 166, 167, 169, 170, 172, 174, 177, 178, 217, 219, 221, 223, 225, 226, 227, 228, 229, 230, 232, 233, 234, 235, 242

V

validation, 30, 212, 213
vector, 30, 95, 101, 102, 110, 118
vehicles, 118
Vein, 132, 133, 135, 136, 137, 138, 143, 144, 145, 146
  pattern, 133, 136, 137, 145
  recognition, 144
vein-based, 136, 137, 138, 145
  biometrics, 136, 137, 138, 145
velocity, 97, 118
videos, 93, 96, 97, 110, 111, 132, 145
vision, 5, 6, 10, 11, 12, 87, 88, 89, 90, 91, 92, 103, 114, 115, 218, 238
visual area, 100
vulnerability, 142, 197, 199, 200, 201, 202

W

walking, 92, 217, 222, 236, 237
Web, 5, 50, 62, 124, 151, 190, 191, 196, 205
  browser, 50

websites, 134
weighted-average splining, 66, 73
Wi-Fi, 124
windows, 102, 141
wireless sensor, vii, 117, 118, 119, 120, 121, 123, 127, 129, 130
  networks, vii, 117, 130
wireless sensor networks, vii, 117, 130
workplace, 221

X

XML, 124, 184