A Thesis for the Degree of Ph.D. in Engineering

A Study on Security and Privacy for Ad-hoc Network, VoIP Service and RFID-enabled Supply Chains System

February 2016

Graduate School of Science and Technology
Keio University

Kentaroh Toyoda

Acknowledgments

I would like to express my special appreciation and thanks to my supervisor, Prof. Iwao Sasase. You gave me a lot of chances to be a full-fledged scientist. The first and most important thing you taught me is that doing research is finding a root cause of the problem and giving a theoretical solution for it. This instruction gave me a confidence that I can do on any research field. Prof. Sasase also gave me a lot of chances to enhance my carrier. You entrusted me application for a competitive research fund (MEXT Kakenhi) and research instruction for junior students in the laboratory. These experiences will help my carrier as a researcher.

I would also like to thank my committee members, Prof. Panagiotis Takis Mathiopoulos, Prof. Tomoaki Ohtsuki, Prof. Naoaki Yamanaka, Prof. Yukitoshi Sanada for serving as my committee members even at hardship. I also want to thank Prof. Satoru Okamoto.

My family and friends were a great support from both mental and financial aspects. Especially, I would like to express my gratitude for my parents, Noriko Toyoda and Shinkichi Toyoda, and my grandparents, Haruko Toyoda and Toshiro Toyoda, and Misako Tanaka and Masao Tanaka. You all forgave me to proceed to Ph.D course without any complaint. Without their support, I could not finish my Ph.D thesis.

Contents

List of Figures

List of Tables

Acronyms

AAA

Authentication Authorization and Accounting . . . . . .

ACD

Average Call Duration . . . . . .

AN

Airborne Network . . . . . .

CPD

Calls Per Day . . . . . .

DHT

Distributed Hash Table . . . . . .

EPC

Electronic Product Code . . . . . .

EPCDS

EPC Discovery Service . . . . . .

EPCIS

EPC Information Service . . . . . .

FFS

Feige-Fiat-Shamir protocol . . . . . .

GF

Galois Feild . . . . . .

GQ

Guillou-Quisquater protocol . . . . . .

IOR

Incoming/Outgoing Ratio . . . . . .

MIPv6

Mobile IPv6 . . . . . .

MITM

Man-in-The-Middle . . . . . .

P2P

Peer-to-peer . . . . . .

PAM

Partitioning Around Medoids . . . . . .

RF

Random Forests . . . . . .

RND

Random sequence . . . . . .

SGTIN

Serialized Global Trade Item Number . . . . . .

SIP

Session Initiation Protocol . . . . . .

SPIT

SPam over Internet Telephony . . . . . .

SPITter

A SPIT caller . . . . . .

ST

Strong Ties property . . . . . .

TID

Transaction Identifier . . . . . .

VoIP

Voice over Internet Protocol . . . . . .

WSN

Wireless Sensor Networks . . . . . .

WT

Weak Ties property . . . . . .

ZKP

Zero Knowledge Proof . . . . . .

Symbols

$\mathcal{A}_{privacy}$

A privacy attacker . . . . . .

$\mathcal{A}_{robustness}$

A robustness attacker . . . . . .

b

A binary sequence for FFS protocol . . . . . .

e

A challenge in FFS protocol . . . . . .

$\mathbb{C}$

An access code space . . . . . .

Collect

Algorithm to collect shares for an attacker . . . . . .

x

A commitment in FFS protocol . . . . . .

d_{prover}

A device of a prover in FFS protocol . . . . . .

$d_{verifier}$

A device of a verifier in FFS protocol . . . . . .

d_{target}

The target average call duration for the SPITters with colluding accounts .

E_{conv}

The expected number of multiplication in the conventional scheme . . . . .

ϵ_p

The negligible function of a privacy attacker . . . . . .

ϵ_r

The negligible function of a robustness attacker . . . . . .

E_{prop}

The expected number of multiplication in the proposed scheme . . . . . .

f_{ACD}

A value of ACD . . . . . .

f_{CPD}

A value of CPD . . . . . .

f_{IOR}

A value of IOR . . . . . .

f_{ST}

A value of ST . . . . . .

f_{WT}

A value of WT . . . . . .

$H()$

A hash function used in FFS protocol . . . . . .

M_{conv}

Required memory amount in the conventional scheme . . . . . .

M_{prop}

Required memory amount in the proposed scheme . . . . . .

M_{try}

The number of features used in tree construction in RF . . . . . .

μ_{comp}

The mean value of compensation calls . . . . . .

μ_{SPIT}

The mean value of SPIT calls . . . . . .

$N_{callers}$

The total number of callers to be inspected . . . . . .

n_D

The number of dummy tags . . . . . .

N_{days}

The number of days for inspection . . . . . .

$N_{features}$

The number of features used for RF+PAM . . . . . .

$n_{\mathbf{IP}}$

The number of IP addresses that an attacker possesses . . . . . .

n_L

The number of legitimate products . . . . . .

p_{detect}

The probability that an authentication server can detect the attacker . . .

$\mathbf{v}$

A public key of a prover . . . . . .

r_D

The ratio of dummy tags to the entire product . . . . . .

Recover

Algorithm to recover an access code from shares . . . . . .

$\mathbf{y}$

A response in FFS protocol . . . . . .

$R_{SPITters}$

The ratio of SPITters to the entire caller . . . . . .

r_τ

The ratio of required tags to the entire product . . . . . .

Share

Algorithm to split an access code into shares . . . . . .

$\mathbf{s}$

A secret key of a prover . . . . . .

$\mathbb{S}$

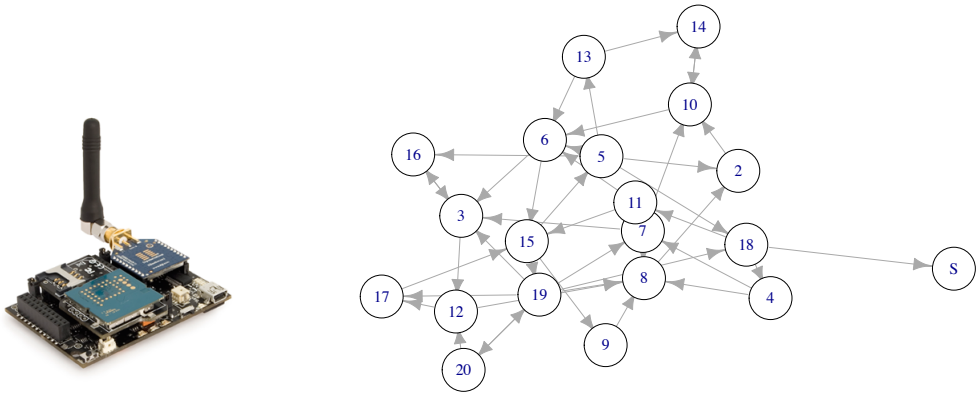
Share space of an access code . . . . . .

Chapter 1

Introduction

1.1 Systems and Services Requiring Wireless Devices

In the late 20th century, the Internet was widely spread thanks to the technological advances in the personal computer and backbone network. The Internet enables us to communicate with people in all over the world and broadcast information via web sites. However, recently, not only people but also things communicate each other due to technology improvement, i.e., miniaturization of devices and improvement of wireless technology in the near future. There are three such devices, which are wireless sensor devices, smartphones, and RFID (Radio Frequency IDentification), other than the personal computer. By using these devices, new systems and services have been emerged. For example, wireless sensor devices are used to monitor structural health for buildings or infrastructure in real-time [1]. Sensor devices are deployed in a building and a sink device periodically collects structural health data from sensor devices and monitors the building. In addition, the wide spread of smartphones and tablet devices yields many services e.g., online social communications and location services. The RFID technology eases the complex operations in supply chines, e.g., traceability, quality management, and recall problem. A manufacturer creates, composes, and ships products with an EPC (Electronic Product Code) to distributors. An EPC is written into a tag and an RFID tag is attached to a product. By interrogating RFID tags, each party knows when, where, and which party deals with products. In this



(a) A wireless sensor device (cited from `http://www.libelium.com/`)

(b) An example of wireless sensor network

Figure 1-1: Example of wireless sensor network.

section, we summarize the systems and services with (i) wireless sensor devices, (ii) smartphones, and (iii) RFID.

1.1.1 Wireless Sensor Devices

We first introduce systems and services that require wireless sensor device. Figure 1-1(a) shows an example of it. Wireless sensor devices equip several sensors, e.g., pressure, humidity and temperature sensors, and wireless interfaces, e.g., 3G, ZigBee, WiFi, and Bluetooth. By deploying wireless sensors in a building, home, industry, and even the public area in the city, we can continuously collect sensory data from them. Figure 1-1(b) shows an example of WSN (Wireless Sensor Network). In this figure, a wireless sensor device is indicated by a circle with a number and is connected with other devices by an ad-hoc network. Although the topology depends on what systems and services are deployed, in most cases, a sink node, which is denoted as 'S' in Figure 1-1(b), collects sensory data and sends commands to underlying nodes. In the following, we introduce some representative systems and services that use wireless sensor devices.

Sensor-enabled Homes and Buildings

One of the benefits provided by sensor devices is to visualize unnoticed cost and waste [2], [3]. Sensor devices are deployed everywhere in the home and building and monitor temperature and light [3]. It enables to automatically control light bulbs and air-conditioning units. This process is automatically optimized by the system and thus much energy can be saved compared with self-control by human.

Structure Health Monitoring

The beneficial system is environmental monitoring, e.g., volcanic areas, oceanic abysses, roads, tunnels and buildings where human cannot easily enter [4], [5]. One of the examples is the bridge monitoring system, e.g., [6]–[8]. In this system, sensor devices are deployed on foot bridges and measure structural health, e.g., crack and tension detection. Since sensory data are periodically sent to a sink node, any abnormal events can be quickly detected. Another merit to use sensor devices in the bridge monitoring system is that sensor devices can harvest energy from solar and wind power to operate.

Forest Monitoring

Another example is fire detection in the forest with WSN [9], [10]. It is undesirable to monitor abnormal events from airplanes due to the cost. In sensor-enabled fire forest systems, sensor devices are deployed on trees and/or ground and temperature and humidity are periodically measured. If any anomaly, e.g., sudden increase in temperature, is detected at a sensor device, it informs the sink node of anomaly in order for persons to take a quick action for it.

ITS (Intelligent Transportation System)

Sensor devices make infrastructures smarter. ITS is an example of smart infrastructures [11]–[13]. If cars are equipped with wireless communication modules, e.g., ZigBee and/or 3G communication module, a congestion-free traffic route may be offered.

In addition, by using sensor technology, a smart parking system can be realized that monitors available parking spaces and suggests the best candidates to park his/her car [14], [15].

Smart Metering System

Smart metering is another example. A smart meter is equipped with a wireless communication module and a wireless network is created with other smart meters and utility. By using this network, the utility can real-timely collect each household's electric consumption and realize demand-response [16]. In addition, consumers can also know when, which and how much each device is used [3].

Healthcare

Wireless sensor devices are also used in the medical services [17], [18]. For example, a patient's body temperature, blood pressure, and breathing activity are measured and are sent to a sink device. By doing this, patient's health can be remotely and quickly monitored. Wearable devices equipped with a step counter and a heart rate tracker can track the personnel activity which visualizes total walking/running distance and consumed calories and may enhance their lifestyle.

1.1.2 Smartphones

The smartphone is the more powerful device compared with wireless sensors and RFID. The smartphone is typically equipped with 3G/4G, WiFi, and Bluetooth connectivities and several sensors, e.g., GPS, accelerometer, and light sensor. By leveraging them, new systems and services have been emerging. For the convenience of presentation, only two major services, which are the voice and video communications services and location services, are shown in the following.

Voice and Video Communications Services

VoIP (Voice over IP) service enables us to communicate each other at very low charge rate or even free of charge [19]. Since recent voice communication services use the IP network instead of PSTN, the cost and charge for calling gets much decreased. In the IP network, voice communication service, such as VoIP, uses SIP (Session Initiation Protocol) for session management e.g., session establishment, forking, and termination [20]. Since Android and Apple's iOS devices can be VoIP/SIP clients, the services would be widespread for smartphones' users. Not only VoIP/SIP service providers but also online social networks, e.g., Facebook and LINE, have started voice communication services [21]. They are also offered free of charge or at very low calling rate.

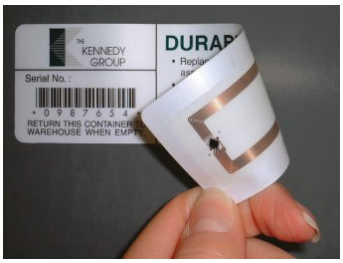
Location Information Services

As smartphones are typically equipped with GPS and WiFi subsystems, a service provider can offer point-of-interests, e.g., cafes, museums nearby a user, and best traffic route to a desired destination [22]. Another example of location information services is mobile cloud sensing which is a new sensing service with smartphones [23], [24]. In this service, sensory data are measured by users' smartphones and are sent to a service provider. Sensory data include temperature, air pollution data, noise level, and even photos of point-of-interests. For instance, a user who wants to know air pollution information in the specific area, he/she queries to the server and retrieves the result and then a querier pays incentive to the provider.

1.1.3 RFID

RFID is a promising technology to identify objects. RFID consists of readers and tags. In general, tags send its information to readers. Tags are classified into active and passive ones. Active tags operate with battery while passive ones are not equipped with any battery and operate with continuous wave fed by a reader [25]. Although operating frequency bands range from 120 kHz (low frequency) up to 10

(a) HF RFID tag (cited from www.boston.com)



(b) UHF RFID tag (cited from www.harlandsimon.co.uk)

(c) UHF RFID reader (cited from www.barcodesinc.com)

Figure 1-2: RFID devices.

GHz (ultra wide band), 13.56 MHz (high frequency) and 865-920 MHz (ultra high frequency) bands are widely used. The communication range of HF (High Frequency) RFID is up to 1 meter and thus it is suitable for smart cards for transportation systems or wireless payment systems. On the other hand, UHF-band RFID is used for object management e.g., supply chains since tags can be interrogated within 10 meters. RFID-enabled supply chains ease the complex operations in supply chines, e.g., traceability, quality management, and recall problem [26]. Typically, three parties, which are manufacturers, distributors, and retailers are involved in the supply chains. A manufacturer produces, composes, and ships products toward distributors. It also generates an EPC (Electronic Product Code) to each product and attaches it to a product. The EPC typically involves the information of product, e.g., item type, company identifier, and a serial number. After distributors and retailers receive products, EPCs are interrogated for each product by using RFID readers. Each party can retrieve more detailed information about EPCs from EPCIS (EPC Information Service) server managed by a manufacturer [27]. Similarly to DNS (Domain Name Service) in the Internet, a party queries EPCDS (EPC Discovery Service) servers with interrogated EPCs and find the location of EPCIS on the Internet.

1.2 Security and Privacy Issues for Emerging Systems and Services

Although wireless technology enables us to produce new useful systems and services, unprecedented security and privacy issues will occur. For example, wireless sensor devices connect each other by creating ad-hoc network. Since it might be difficult to place administrative devices, an attacker can easily intercept, eavesdrop, and disrupt the network. Another example is that systems and services requiring smartphones are vulnerable to service specific issues. For example, let us consider the location service in the mobile environment. In this case, a service provider may be able to know where a user wants to go and even track the user, which causes a privacy issue. Not only sensor devices and smartphones but also RFID suffers from a privacy issue because tags can be interrogated by any RFID readers without being noticed. In other words, an attacker can also obtain the information of products without being detected by a tags' owner.

Therefore, it is quite important to grasp the threats of emerging systems and services and propose solutions to avoid them. In the following, typical security and privacy issues against them, namely the attacks and defences for (i) ad-hoc network, (ii) VoIP service, and (iii) RFID-enabled supply chains. In addition, although many works will be cited, only key papers are introduced.

1.2.1 Ad-hoc Network

The attacks against the systems and services with wireless sensor devices are mainly related with ad-hoc network. Since sensor nodes may be deployed in the area where anyone can access, e.g., roads, forest, and buildings, attackers can easily replace and/or add sensor devices [76]. Attacks and defences against ad-hoc or wireless sensor network are summarized in detail in [76]. Figure 1-3 shows a classification of three major attacks and defences in ad-hoc network. The attacks involve (i) routing-related attack, (ii) eavesdropping, and (iii) node compromise. We describe the three attacks

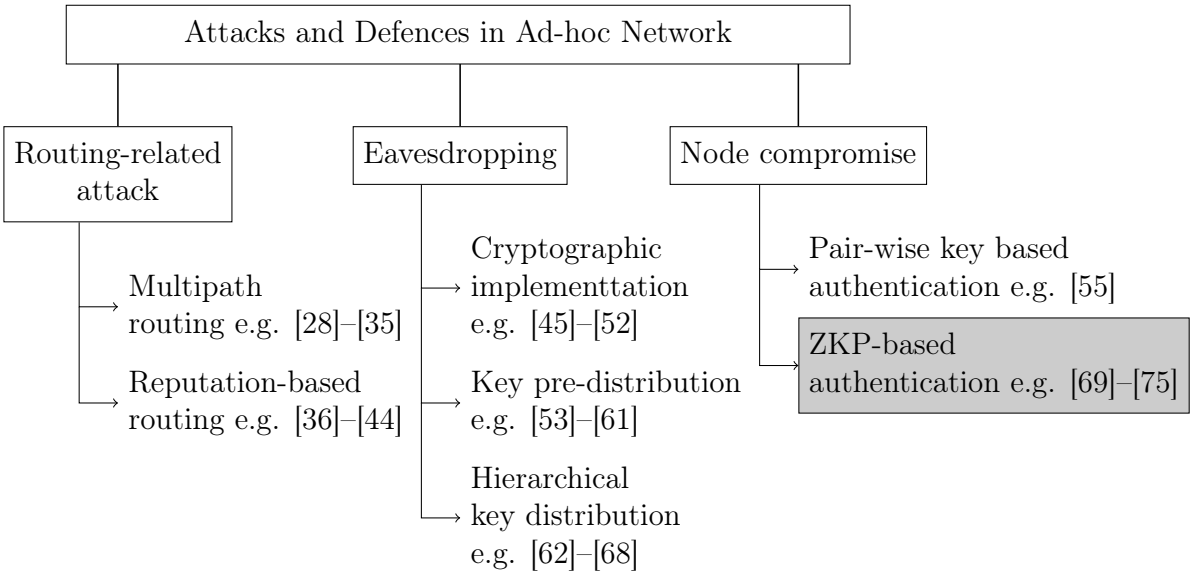


Figure 1-3: Classification of major attacks and defences for ad-hoc network.

one by one. First, the routing-related attack is that attackers typically inject malicious nodes into the network and let them do illegal executions. For example, in order to disrupt network functionality, a malicious node drops, forges, and forwards packets to a wrong destination. In order to detect or avoid these attackers, secure routing protocols have been widely proposed. These include multipath routing schemes e.g., [28]–[35] and reputation based ones e.g., [53]–[61]. The basic idea of multipath routing based schemes is that a sensor device sends data to a sink node via multiple paths to reliably transfer data. In reputation based routing schemes, each device rates their neighbours with performance parameters, e.g., the successful packet transmission rate and decides a proper route with reputations.

We then describe the eavesdropping attack that attackers eavesdrop transmitted messages and try to decode them. If a device sends a very sensitive information, e.g., consumed electricity in smart metering system, it will be a privacy issue. In order to avoid eavesdropping, data must be encrypted before transmission. Due to the limited computational power of sensor devices, it is important to design a lightweight cryptographic scheme. For example, Karlof *et al.* proposed TinySec, which is the first symmetric encryption implementable to an off-the-shelf sensor device [45]. In

general, the PKC (Public Key Cryptography) is much heavier than the symmetric key cryptography and thus it is important to design lightweight PKC. Liu and Ning proposed TinyECC, which is an implementation of PKC [47]. They use ECC (Elliptic Curve Cryptography) which is one of the most efficient PKC. Szczechowiak *et al.* also proposed but much faster implementation by introducing pairing over binary [49]. In a sensor network, it is quite useful if a node identifier can be used as a public key, namely IBE (Identifier Based Encryption). Oliveira *et al.* proposed an identity-based paring based cryptography called TinyPBC [51].

In addition, key management schemes are necessary for encryption and decryption because devices may easily leave and attend a network. There exist (i) key pre-distribution schemes, e.g., [55], [57], [61] and (ii) pair-wise key establishment scheme with a trusted intermediate node e.g., [77], and (iii) key management schemes in hierarchal network [62], [63]. In the key pre-distribution schemes, a set of keys are pre-distributed to each node and when two nodes want to communicate, they share a pair-wise key from a key set. For instance, in [55], a key is chosen from a set of keys called key pool and is used for the symmetric encryption. The receiver node tries to detect which key is used for decryption. However, Chan and Perrig argue that key pre-distribution schemes do not scale to large network and proposed a pair-wise key establishment scheme with intermediate nodes [77]. In this work, one or multiple trusted intermediate nodes are used to establish a pair-wise key between two nodes that want to share a key. Another research includes a hierarchal key distribution. In this research, not a pair-wise key but a key shared with entire network is distributed to devices. Such key is used for securing the message broadcast by a sink node. In order to avoid the leakage of a key, it must be updated whenever nodes leave and attend the network. As mentioned by many researchers, by taking into account that a network is large, the number of messages for key update must be smaller, e.g., [62]–[68]. Of these works, the work by Pietro *et al.* is most fundamental one. They proposed LKHW (Logical Key Hierarchy for Wireless sensor networks) that divides the nodes into several hierarchal groups and iteratively update keys by groups [62].

Finally, we introduce the node compromise attack and its countermeasures. In this

attack, attackers replace deployed sensor nodes with compromised malicious nodes. The compromised nodes may possess a valid identifier, e.g., IP address, and thus they can impersonate existent legitimate nodes. In order to avoid impersonation, node-to-node (or device-to-device) authentication is necessary for the network. In this case, the challenge is that any trusted third party cannot be used since each node is not connected with the Internet. There are typically two authentication approaches. The first one is using a pair-wise key for authentication e.g., [55]. Nodes authenticate each other by checking the possession of a correct pair-wise key. The other one is to use ZKP (Zero Knowledge Proof) as authentication scheme [69]–[75]. ZKP is that a prover, who possesses a pair of public and secret keys, tries to convince a verifier the possession of keys without revealing the keys itself. By using ZKP, a receiving node can judge whether a sending node is impersonated or not without requiring any trusted third party. A major implementation of ZKP is FFS (Feige-Fiat-Shamir) protocol [78] and it is used in many works, e.g., [71], [73], [74]. Udgata *et al.* and Hashim *et al.* proposed an FFS-based authentication scheme for WSN to avoid a malicious node from impersonating a legitimate node [73], [74]. Kizza *et al.* proposed to use FFS protocol to authenticate devices in AN (Airborne Network), where flying devices construct ad-hoc network and perform tasks, e.g., environment sensing and monitoring ground in the sky [71].

1.2.2 VoIP Service

The one of the most fundamental functionalities of smartphones is the communication between users. Therefore, it is required for a service provider to offer a secure and privacy-preserving service. Figure 1-4 shows a typical classification of major attacks and defences for VoIP service. We refer the work by Keromytis [117] to summarize this figure. The first threat in the VoIP service is SPIT (Spam over Internet Telephony), which an attacker spreads advertisement, malicious phishing, and persistent survey. One of the major challenges in SPIT detection is that he legitimacy of a call cannot be judged before a callee takes it. Nevertheless, there are many countermeasures against SPIT and they are mainly divided into three research categories: (i)

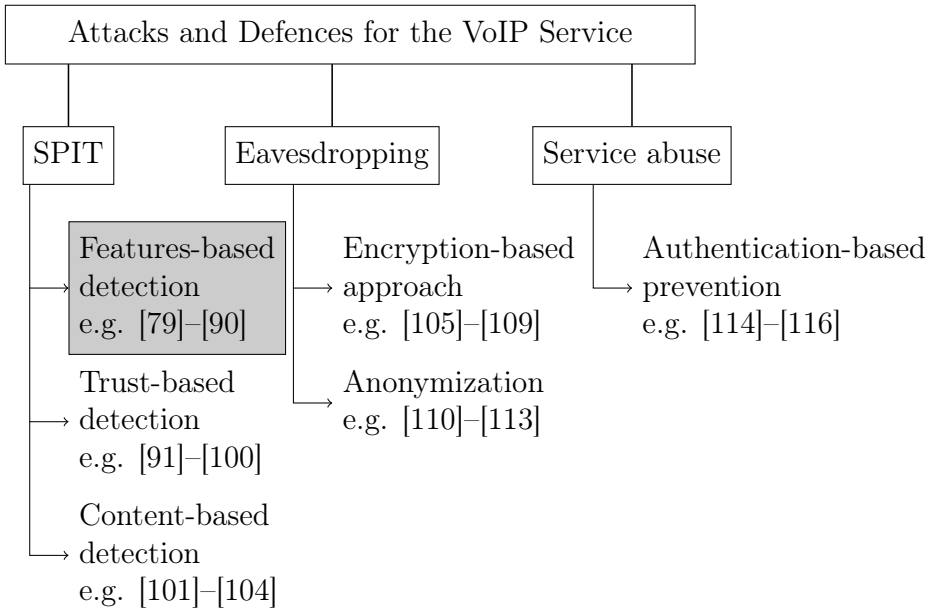


Figure 1-4: Classification of major attacks and defences for the VoIP service.

features-based SPITters (SPIT callers) detection scheme, e.g., [79], [82], [87], [88], (ii) SPITters detection based on social network trustworthiness, e.g., [91], [93], [96], [98] and (iii) content-based SPIT call detection, e.g., [101]–[104]. The feature-based SPITters detection scheme is to detect callers of SPIT by analyzing users' call logs. For example, since it can be considered that some of SPITters may make a large of calls, one of the calling features, call frequency, is effective to detect SPITters [79]. Another example is to use the difference of distribution of call duration between SPITters and legitimate callers [87]. The approach of detection schemes based on social network trustworthiness is to detect SPITters by measuring the reputation of callers. Balasubramaniyan *et al.* proposed CallRank that uses call duration as trustworthiness between a caller and callee [91]. The idea of this scheme is based on the fact that if one calls to a friend for a certain time duration, say 10 min, it must be trustful and a reputation is given to a caller according to the call duration. Azad *et al.* proposed Caller-REP which is a reputation-based SPITters detection scheme [98]. In this scheme, three calling features, which are call duration, call frequency, and the node degrees, are used for calculating the trustworthiness of a caller. The approach of the last category is to detect SPITters by analysing the content of calls.

Lentzen *et al.* and Strobl *et al.* proposed a content-based SPIT detection scheme by fingerprinting the audio data [103], [104]. In these schemes, an audio fingerprint of spectral feature vectors is computed for incoming call. Using a database of feature vectors, new calls are compared with previous ones and replays with identical or similar audio data are detected. Future calls from the same source can then be blocked during call setup.

VoIP services may be also vulnerable to eavesdropping attack. The eavesdropping attack is that an attacker listens in on the signal or the content of a VoIP call session. Wright *et al.* showed that even if audio data is encrypted, it can be inferred what language is spoken between two [118] and even some phrases can be identified with probability of greater than 90% [119]. They leverage the length of encrypted VoIP packets to train phrases with HMM (Hidden Markov Model). In order to avoid the information leakage of VoIP/SIP services, two approaches exist, namely (i) encryption-based, e.g., [105]–[109] and (ii) anonymization, e.g., [110]–[113]. One of the representative work in the former approaches is [107]. The authors of [107] proposed that multiple keys are shared and are switched during the call without being detected by an eavesdropper. Since audio data is stream and VoIP is used on mobile phones, a lightweight encryption is necessary for securing contents [106]. The latter approach is to anonymize the routes that audio data pass through. For example, Zhang and Fischer-Hübner proposed to use Tor (The onion router) which is a well known proxy-based anonymised network. Since routes are periodically changed in Tor, it is robust against eavesdropping.

The last one is called a service abuse attack. In this attack, attackers make victimized accounts use some premium services and charge them its fee [120]. This is a specific issue for VoIP/SIP services because session is established via the Internet, and thus an attacker can execute MITM (Man-in-The-Middle) attack by illegally hijacking some SIP messages, e.g., `BUSY` and `BYE`. In order to avoid this attack, many researchers proposed source authentication schemes, e.g., [114]–[116]. For example, Mazurczyk and Zbigniew proposed a digital watermarking technique for audio data and SIP messages to authenticate both the caller and voice data [114]. Geneiatakis

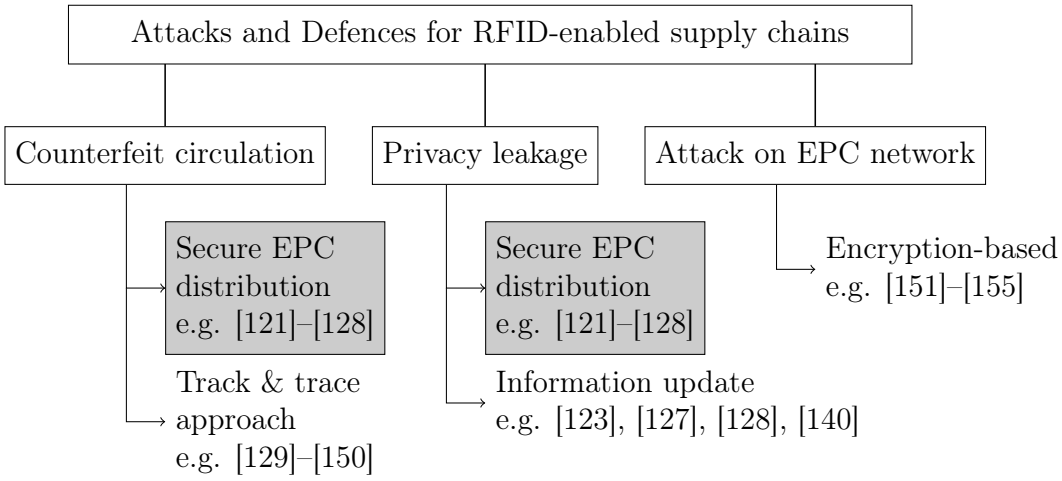


Figure 1-5: Classification of attacks and defences for RFID-enabled supply chains.

et al. proposed an AAA (Authentication Authorization, and Accounting) assisted billing scheme in VoIP [115]. In this scheme, a user authentication is executed before SIP establishment with an AAA server.

1.2.3 RFID

Figure 1-5 shows a classification of attacks and defences of RFID-enabled supply chains. Although other classifications may exist, we classify them as (i) counterfeit circulation, (ii) privacy leakage, and (iii) attack on EPC network for the convenience of presentation.

We first explain the counterfeit circulation attack. In this attack, an attacker obtains genuine EPCs by any means and injects counterfeits with copied tags in the supply chains, e.g., [136], [142], [146]. Since no party can detect counterfeits at the time of arrival, each party must check the EPCs of tags. If counterfeits pass this check, they are circulated in the supply chains. Therefore, we must avoid the leakage of genuine EPCs. For this purpose, an attacker who wants to obtain EPCs must be modeled in the real-world setting and many security researchers have modeled the "hit and run" attacker, e.g., [121], [140], [156]. They assume that RFID tags can be only interrogated by the attacker during transportation, e.g., truck carrying products with tags. This is rational since an attacker may not be able to interrogate tags inside

parties and public area is the only channel to interrogate tags on products. Based on this attacker model, many researchers proposed schemes to securely distribute EPC in the supply chains, e.g., [121]–[128], [156]. One of the most important work is a secret sharing based key distribution proposed by Juels [156]. In this scheme, EPCs are encrypted with a symmetric encryption scheme and an encryption key is split into multiple shares by (τ, n) secret sharing scheme [156]. The secret sharing scheme realizes that one can extract the key if he/she can obtain more than τ unique shares out of n shares [157]. An encrypted EPC and share of a key is written into a tag on product. After receiving products, an authorized partner interrogates tags, recovers key from sufficient number of shares, and finally decrypt EPCs with the extracted key.

Many researchers consider the case when genuine EPCs are leaked to attackers and counterfeits are injected into the supply chains. In order to detect counterfeits products in the supply chains, the track and trace approach is studied by many researchers e.g., [129]–[150]. In the track and trace approach, each product is checked whether it goes through legitimate paths from a manufacturer to retailers. This approach assures that genuine products are certainly passed through legitimate parties and detects any products that do not pass correct parties as counterfeits. The common idea of these works is to update the extra information of tags whenever products are arrived at parties in supply chains. For example, Zanetti *et al.* proposed a tailing scheme [146]. This scheme requires a detector who can obtain information of a supply chain. When a party interrogates a tag, it then checks this extra information and query a detector. If no inconsistency is found, a party writes random data to the next available position in the tag memory. By doing this, a detector can detect counterfeits if any inconsistency is found on this extra information. Another example is to use the ordered multi-signature scheme, which is a signature scheme that preserves the order of signatures, and write or update signature directly to tags on products, e.g., [139], [142]. This way also assures that products pass through correct parties in order.

The second attack we explain is privacy leakage. Since an EPC involves very sensitive information e.g., a product code, a company code, and serial number, an

attacker may obtain such information. It could be that an attacker queries obtained EPCs to EPCIS servers and retrieves more information about products. Furthermore, if an attacker tracks from a manufacturer to the end retailer, he/she can infer the business relationships. In order to avoid the information leakage, secure EPC distribution schemes are also effective. However, they cannot solve the leakage by tracking. Therefore, many schemes try to effectively update the tags' contents, e.g., [123], [127], [128], [140]. For example, Cai *et al.* proposed a flexible secret update for secret sharing scheme [123].

The last one is attack on EPC network. A party of supply chains may retrieve more detailed information of products and/or update the information of tags in EPCIS. In this case, it queries EPCs to EPCDS server and obtains the location of EPCIS server where tags' detailed information exist. Although EPC network is useful for visualising the products information, many researchers argue that the access to EPC network, especially EPCDS, must be restricted to authorized parties unless attackers can retrieve sensitive information of products, e.g., [152], [153], [155]. Therefore, the securely designed EPCDS is necessary and many solutions have been proposed. For example, Shi *et al.* proposed a policy based access control EPCDS system [152]. In this scheme, the authorization language is used to specify who is allowed to perform what operations on what data. Fabian *et al.* proposed a privacy-enhanced EPCDS which is designed with a DHT (Distributed Hash Table) structure and uses a secret sharing scheme for securely preserving the information [153]. The information of EPC, e.g., the location of EPCIS, is split into shares by a secret sharing scheme and is put on nodes of DHT. Only the person who knows true EPC can retrieve sufficient shares of tags' information.

Table 1.1. Contribution of this thesis.

(a) Chapter 2

Purpose	Lightweight Verification Scheme in Feige-Fiat-Shamir Protocol
Issue	The verification cost is high.
Proposal	Restrict the maximum number of elements set as 1 in each challenge to decrease the number of multiplication, which is the heaviest part of verification, without lowering the security.
Result	Process time is shortened by 37-73% on an Android device.

(b) Chapter 3

Purpose	Unsupervised Clustering-based SPITters Detection Scheme in VoIP Service
Issue	Training data cannot be used for SPITter detection due to the privacy issue.
Proposal	Find the (dis)similarity between callers from calling features and cluster them.
Result	When the SPITters account for more than 20%, better classification accuracy is achieved compared with the conventional schemes.

(c) Chapter 4

Purpose	Secure Products Distribution Scheme in RFID-enabled Supply Chains
Issue	EPCs may be leaked during transportation without being noticed.
Proposal	EPCs are masked with random numbers. Introduce the authentication server to detect attackers and to securely distribute the random numbers. An authentication code is split into shares with dummy tags.
Result	By adding dummy tags and an authentication server, an attacker can be detected with high probability even if he/she possesses 10,000 IP addresses.

1.3 Research Motivations and Contributions of Dissertation

Although many researchers have solved important security and privacy issues, they have limitations and require further improvements. Table 1.1 shows the summary of

contribution of our research. For example, although FFS (Feige-Fiat-Shamir) protocol is suited to pervasive environment where trusted third party cannot be used, it is necessary to consider the cost on verifiers because any device must verify an authentication request whenever he/she receives it. Hence, the first contribution is mainly related with systems for wireless sensor devices and we propose a lightweight verification scheme in device-to-device authentication scheme in ad-hoc network environment. We first point out that the heaviest calculation in the verification is to multiply 1,024-bit variables many times and when multiplication occurs. Based on these observations, we propose a provably secure lightweight verification scheme in FFS protocol. The basic idea is to divide the protocol into multiple phases and restrict the upper bound of elements set as 1 in generating challenges so as to decrease the number of multiplication. We give security analysis and it will be proven that our scheme is satisfied with the requirements for ZKP. Some performance metrics, e.g., the expected number of multiplication, required memory amount, and the process time measurements on an Android device, are evaluated and it will be shown that our scheme effectively decreases the verification cost without lowering the required security.

Previous SPITters detection schemes in the VoIP service have also limitations. Although there are many calling patterns that are effective to distinguish SPITters and legitimate callers, it is difficult to obtain training data labeled as "SPITter" or "legitimate caller" for threshold-based and supervised machine learning detection schemes. Hence, the second contribution is to propose an unsupervised SPITters detection scheme in the VoIP service. We propose an unsupervised and threshold-free SPITters detection scheme by using a clustering algorithm. The proposed scheme tries to separate the inspected callers into two clusters, one is the legitimate cluster and the other is the SPITters one by using multiple features. Since the scheme leverages the features to find dissimilarity among the callers, any complex threshold settings and training phases can be avoided. Although clustering itself does not give us the SPITter cluster, the "SPITters cluster" can be identified by comparing the average of a feature, e.g., calls per day. Since if the callers are clustered well, the callers in one

of the cluster call more frequently than the others and it can be done without the training phase. By the computer simulation with real and artificial datasets, it will be shown that the proposed scheme achieves the better detection accuracy when the SPITters take account of more than 20% of entire caller.

The third contribution is to propose a secure product distribution scheme in RFID-enabled supply chains. In RFID-enabled supply chains, it is necessary to protect the contents of EPCs (Electronic Product Code) since an EPC contains sensitive information such as the product code and serial number. Although many protecting schemes have been proposed, no scheme can limit the number of illegal attempts for revealing EPCs or notice whether an attacker exists. Moreover, the conventional schemes assume a weak adversary and, in reality, EPCs may be still revealed to an attacker. Hence we propose a secure illegal interrogation detectable products distribution scheme for RFID-enabled supply chains. The idea is to make an attacker access an authentication server and detect him/her. EPCs are masked with random sequences and the masked ones are written into genuine tags on products while random sequences are placed on an authentication server with an access code. An access code is divided into shares with a secret sharing scheme and they are written into genuine tags. The second proposal is to prepare dummy tags, which are extra off-the-shelf tags and possess bogus shares but not attached to any products. Since an attacker who wants to know genuine EPCs may obtain a large number of access code candidates and must try each on the authentication server, the server can detect and limit such illegal attempts. We prove that our construction is secure against 'strong attacker' who can interrogate all tags. We also implement the proposed scheme with off-the-shelf RFID devices and a computer to clarify the latency.

1.4 Limitations of Our Scheme

In order to clarify the contribution, the validity of our schemes must be clarified in each research. Table 1.2 shows a classification of attacks summarized in the Section 1.2 and the validity of our scheme. Here, the validity means whether our research

Table 1.2. Major attacks and the validity of our scheme.

(a) Ad-hoc network

Attack	Overview of attack	Validity of our scheme
Eavesdropping	Passively eavesdrop data in the network	-
Route disruption	Cause DoS by dropping or selectively forwarding packets	-
Node compromise	Impersonate legitimate nodes and collect packets and/or disrupt the network	✓

(b) VoIP

Attack	Overview of attack	Validity of our scheme
SPIT	Make a large number of malicious calls, e.g., advertisement, phishing, and survey	✓
Eavesdropping	Eavesdrop the content of a call	-
Service abuse	Abuse VoIP service, e.g., toll fraud and free rider	-

(c) RFID-enabled supply chains

Attack	Overview of attack	Validity of our scheme
Counterfeit circulation	Make counterfeit products with genuine EPCs	✓
Illegal interrogation	Illegally interrogate genuine EPCs	✓
Privacy leakage	Leak the business relationships in the supply chains and useful information	✓
Attack on EPC network	Typical attacks, e.g., DoS and information leakage of tags on EPCDS and/or EPCIS	-

is related with such attacks. In the first work regarding the device-to-device authentication, only node compromise can be avoided by our scheme. In the second work, which is related with the security of VoIP, SPIT detection is covered by our research. Finally, our third work solves counterfeit circulation, illegal interrogation, and privacy leakage attacks will be solved in the RFID-enabled supply chains.

1.5 Thesis Type

Each chapter except for Chapters 1 and 5 is based on manuscripts published in conferences or journals. Therefore, Chapters 2, 3, and 4 are self-contained manuscripts of our research, in which some similar description exists.

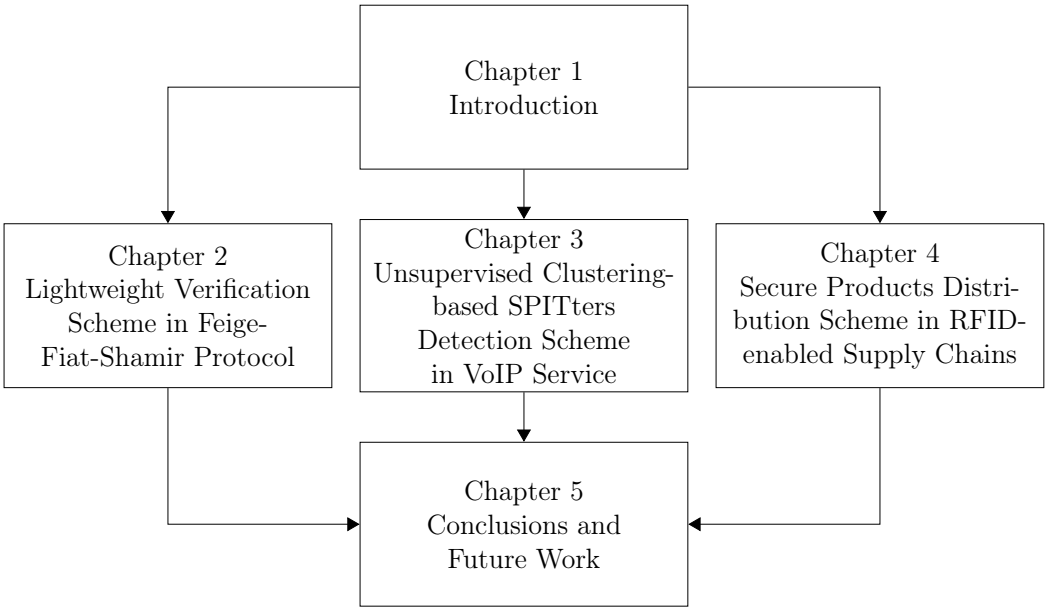


Figure 1-6: Outline of Dissertation.

1.6 Outline of Dissertation

Figure 1-6 shows the outline of this dissertation.

Chapter 2 deals with a lightweight verification scheme in FFS protocol. Related work is summarized in Section 2.2. Section 2.3 deals with the original FFS protocol. The proposed scheme is shown in Section 2.4. The security analysis is given in Section 2.5. The performance evaluation is shown in Section 2.6. The conclusions of this chapter is in Section 2.7.

Chapter 3 deals with an unsupervised SPITters detection scheme. The attacker model is described in Section 3.2. Related work is summarized in Section 3.3. The proposed scheme is described in Section 3.4. The performance evaluation is shown in Section 3.5. Finally we conclude this chapter in Section 3.6.

Chapter 4 deals with a secure product distribution scheme in RFID-enabled supply chains. The system and attacker models are described in Section 4.2. Section 4.3 deals with related work. The proposed scheme is described in Section 4.5. Security analysis is shown in Section 4.6. The performance evaluation is shown in Section 4.7. We conclude the discussion of this chapter in Section 4.8.

Chapter 5 deals with the conclusions of this dissertation and future works.

Chapter 2

Lightweight Verification Scheme in Feige-Fiat-Shamir Protocol

2.1 Introduction

As more and more devices are connected in the ad-hoc network, it is quite important to authenticate a device to avoid impersonation. However, as pointed out in [158], authentication with a third party CA (Certificate Authority) is not suited for such a pervasive environment because each node cannot assure the connectivity with a CA. Hence it is necessary to authenticate nodes only between source and destination nodes. In order to realize it, ZKP has been proposed [159]. ZKP realises that a prover who possesses a pair of public and secret keys convinces a verifier that he/she certainly owns the secret key that corresponds with the claimed public key without revealing the secret key itself. Most of ZKP implementations are interactively executed with commitment, challenge, response, and verification phase. A prover first sends a commitment value to avoid cheating. Then, a verifier replies with a challenge and a prover calculates a response with the received challenge and the commitment already sent by itself. Finally a verifier checks the correspondence of commitment, challenge, and response. There exist many implementations of ZKP, e.g., FFS (Feige-Fiat-Shamir) protocol [78], GQ (Guillou-Quisquater) protocol [160], and Schnorr protocol [161]. Among them, FFS protocol offers a flexible key size and number of rounds and is

suited for many ad-hoc network systems, e.g., [69]–[75].

The FFS protocol offers device-to-device authentication without a trusted third party, e.g., certificate authority. However, it is necessary to consider the cost on verifier when the large number of nodes exist and low computational devices are used, e.g., sensor devices and smartphones. That is, each verifier device must verify every authentication request from both legitimate and malicious devices and it may consume precious energy on devices. Let us consider the case of hierarchical WSN that consists of few cluster heads and many ordinal sensor nodes. In this case, cluster heads must authenticate sensor nodes by itself before collecting data from them. Therefore we should decrease the verification cost in FFS protocol without lowering the required security.

In this thesis, we propose a provably secure lightweight verification scheme in FFS protocol to reduce the computation cost on the verifier. We point out that the heaviest calculation of the verification is multiplication. Therefore, the aim of our approach is to reduce the number of multiplication. In fact, the expected number of multiplication can be controlled when generating a challenge. However, as will be shown later, if the expected number of multiplication is reduced, the attacker's success probability gets higher. In order to maintain the same probability with the original FFS protocol, we divide the protocol into several rounds and a prover device is authenticated if and only if all verifications are passed. The efficiency of our scheme will be shown by means of security analysis, theoretical calculation, and the experiment on an Android device. We first prove that our scheme is satisfied with the requirements of ZKP. Then, we show that our scheme can theoretically reduce the number of multiplication. Finally, we show that the calculation time on an Android device is shortened compared with the original FFS scheme.

The rest of this chapter is as follows. The related work is summarized in Section 2.2. The original FFS scheme is described in Section 2.3. The proposed scheme is shown in Section 2.4. The security analysis is given in Section 2.5. Section 2.6 deals with the performance evaluation. We conclude this chapter in Section 2.7.

2.2 Related Work

Udgata *et al.* and Hashim *et al.* proposed an FFS-based authentication scheme for WSN to avoid a malicious node from impersonating a legitimate node [73], [74]. Similarly, Lu *et al.* proposed a FFS-based authentication for P2P network [69]. By using a one-way hash function and FFS protocol, they realize an anonymous and reliable P2P network.

Le *et al.* proposed to use FFS protocol as the authentication for mobile nodes in a MIPv6 address [70]. In this scheme, a mobile node generates the latter 64-bit of MIPv6 address by a public key and also owns the correspondent secret key. A recipient can check the legitimacy of a source node by authenticating the claimed MIPv6 through FFS protocol.

Kizza *et al.* proposed to use FFS protocol to authenticate devices in AN [71]. In AN, flying devices construct ad-hoc network and perform tasks, e.g., environment sensing and monitoring ground in the sky. The authors assume that devices frequently enter and leave the network and some tasks has time-constraint. Hence, they proposed a modification to the original FFS protocol to quickly finish authentication.

Sandhya and Rangaswamy proposed a FFS-based authentication scheme for the mobile RFID reader [75]. The FFS protocol enables each mobile RFID reader to anonymously being authenticated with a backend server for querying.

2.3 FFS Protocol

We describe the device-to-device authentication protocol based on FFS protocol in ad-hoc network environment. At first, the system and attacker models are described. We then describe the procedures of authentication.

2.3.1 System Model

We first describe the system model. More than two wireless devices, e.g., wireless sensor devices and smartphones, are deployed and connect each other by ad-hoc

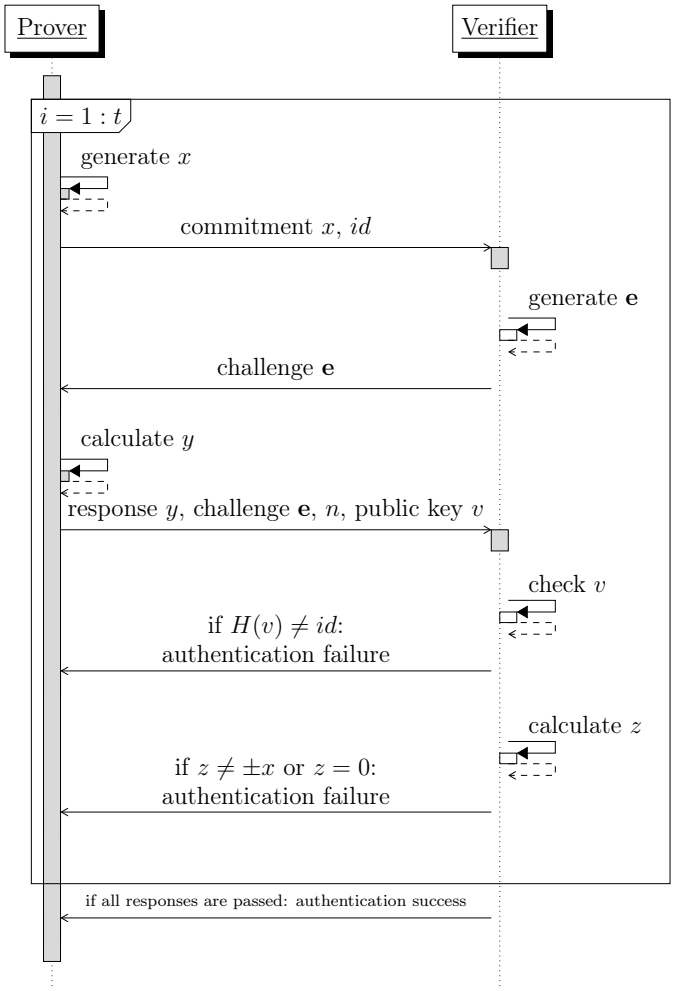


Figure 2-1: Flowchart of FFS protocol.

network. Each device has its own identifier id. In order to calculate id, each device calculates a pair of its own public and private keys $\mathbf{v}$ and $\mathbf{s}$ with a Blum integer $n = pq$ before deployment, where p and q are distinct prime numbers congruent to 3 mod 4 and are kept secret [162]. For achieving sufficient security as described in [70], we assume $|n| = 1,024$ bits. A device generates its secret key $\mathbf{s}$ as follows:

$$\mathbf{s} = (s_1, s_2, \cdots, s_k), \tag{2.1}$$

where k is the number of elements in $\mathbf{s}$ and $s_i, i \in [1, k]$ is randomly chosen in the range of 1 to $n - 1$. The order of k is set to $\mathcal{O}(\log |n|)$ by referring to [78]. Each

device randomly chooses a binary sequence $\mathbf{b}$ as follows:

$$\mathbf{b} = (b_1, b_2, \cdots, b_k), \tag{2.2}$$

where $b_i = \{0, 1\}, i \in [1, k]$. The device also calculates its public key $\mathbf{v}$ as follows:

$$\mathbf{v} = (v_1, v_2, \cdots, v_k), \tag{2.3}$$

where

$$v_i = (-1)^{b_i} \left(s_i^2\right)^{-1} (\text{mod } n). \tag{2.4}$$

By using $\mathbf{v}$ and a cryptographic one-way hash function $H(\cdot)$, an identifier id is calculated as follows.

$$id = H(\mathbf{v}). \tag{2.5}$$

2.3.2 Attacker Model

The attacker model assumed in this thesis is a malicious device that tries to impersonate another legitimate device. The attacker knows a series of n, $\mathbf{v}$ and id of a device to be impersonated, whereas he/she does not know the correspondent $\mathbf{s}$. An attacker tries to authenticate itself to a receiver device, $d_{verifier}$, by challenging the following procedures.

2.3.3 Procedures of Authentication

Let us consider the situation where a device d_{prover} proves the legitimacy to another device $d_{verifier}$ before starting communication with $d_{verifier}$. More specifically, d_{prover} proves that it actually possesses the secret $\mathbf{s}$ that generates its identifier id without revealing the secret itself. It is assumed that the case of two devices, which is a mobile device d_{prover} who wants to prove not to be impersonated and a verifier $d_{verifier}$. This also works in the existence of multiple devices because any device can verify the prover's legitimacy.

Figure 2-1 shows the flowchart of FFS protocol. The FFS protocol consists of four phases, which are (i) commitment, (ii) challenge, (iii) response, and (iv) verification phase.

2.3.4 Commitment Phase

A device d_{prover} proves its legitimacy to a receiver $d_{verifier}$. d_{prover} first sends its identifier id and a commitment x to $d_{verifier}$. x is calculated as follows:

$$x = (-1)^b r^2 (\text{mod } n), \tag{2.6}$$

where $r \in [0, n)$ and $b = \{0, 1\}$ are randomly chosen and kept secret, respectively.

2.3.5 Challenge Phase

After receiving d_{prover}'s id and x, a verifier $d_{verifier}$ generates a challenge $\mathbf{e}$ as follows and sends $\mathbf{e}$ to d_{prover}.

$$\mathbf{e} = (e_1, e_2, \cdots, e_k), \tag{2.7}$$

where $e_i = \{0, 1\}, i \in [1, k]$.

2.3.6 Response Phase

A prover d_{prover} calculates a response y against a received challenge $\mathbf{e}$. y can be calculated as follows:

$$y = r \prod_{i=1}^{k} s_i^{e_i} (\text{mod } n). \tag{2.8}$$

A prover d_{prover} sends a challenge $\mathbf{e}$, public key $\mathbf{v}$, n, and a response y to the verifier $d_{verifier}$.

2.3.7 Verification Phase

When a device $d_{verifier}$ receives the above variables, it checks whether the id is correspondent with the hash of $\mathbf{v}$. If it does not match, authentication fails. Otherwise,

it verifies whether the received y and the sent $\mathbf{e}$ are certainly corespondent with the following equation.

$$z = y^2 \prod_{i=1}^{k} v_i^{e_i} (\text{mod } n). \tag{2.9}$$

If $z \neq \pm x$ or $z = 0$, a verifier $d_{verifier}$ judges that d_{prover} does not possess the secret key $\mathbf{s}$ that corresponds with the public key $\mathbf{v}$. Otherwise, a verifier repeats the above procedures, i.e., commitment, challenge, response, and verification phases until t rounds.

2.3.8 Shortcomings in FFS Protocol

The FFS protocol offers device-to-device authentication without a trusted third party, e.g., certificate authority. However, when we consider that the FFS protocol is used in the network where the large number of nodes exist and mainly low computational devices are used, e.g., sensor devices and smartphones, it is necessary to consider the cost on verifier. That is, each verifier device must verify every authentication request from both legitimate and malicious devices and it may consume precious energy on devices. Let us consider the case of hierarchical WSN that consists of few cluster heads and many ordinal sensor nodes. In this case, cluster heads must authenticate sensor nodes by itself before collecting data from them. Therefore we should decrease the verification cost in FFS protocol without lowering the required security.

2.4 Proposed Scheme

In order to reduce the computation cost on the verifier, we propose a provably secure lightweight verification scheme in FFS protocol by dividing a challenge into multiple challenges and varying the difficulty of each challenge. We point out that the heaviest calculation in the verification is to multiply v_i in Eq. (2.9) since the length of v_i is as long as $|n| = 1,024$ bits. We also argue that v_i is only multiplied when $e_i = 1$ in $\mathbf{e}$. Therefore, the most naïve approach is to make most of e_i to 0. However, it offers significantly lower security because an impersonator, who only knows a someone's

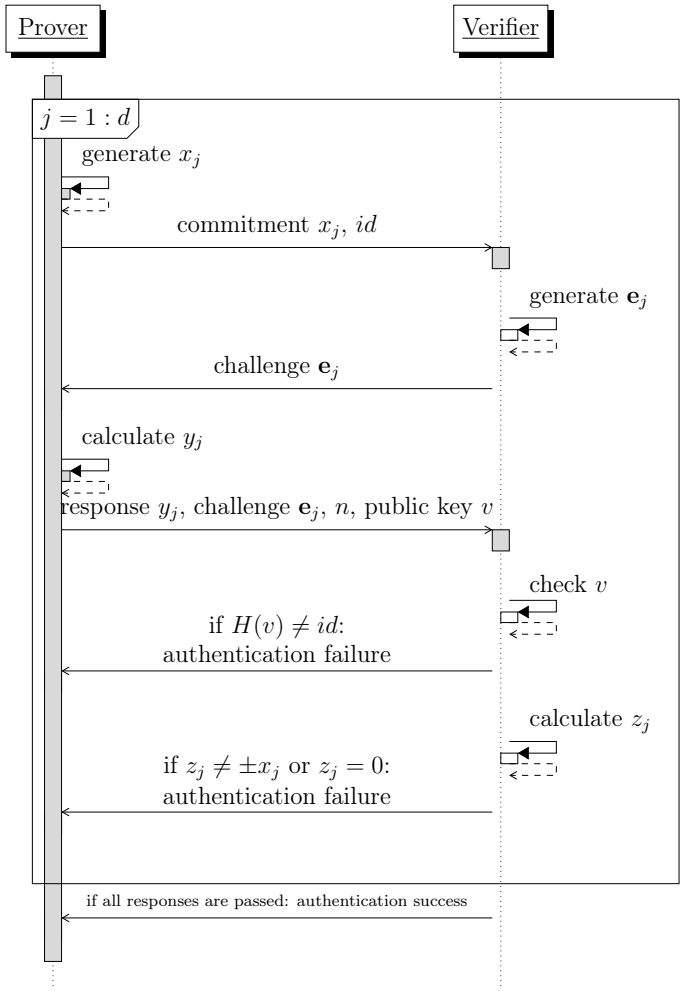


Figure 2-2: Flowchart of the proposed scheme.

public key **v** but does not know its secret key **s** and tries to pass authentication, could impersonate with high probability. In order to decrease the calculation amount without sacrificing security, a challenge is divided into multiple ones and the number of elements that set as 1 in each divided challenge is controlled. In the conventional FFS protocol, a verifier must calculate Eq. (2.9) at once regardless of whether a prover is legitimate or not. On the contrary, the proposed scheme quickly terminates verification for requesting from an impersonator in earlier round. Since a prover must pass every challenge, the proposed scheme can achieve the sufficient security by setting appropriate upper bounds and the number of divided challenges.

Figure 2-2 show a flowchart of the proposed scheme. The proposed scheme also consists of commitment, challenge, response, and verification phases and assumes the same model as the FFS protocol. $j \in [1, d]$ and d indicate an identifier of round and the number of total rounds, respectively. In the following, each phase is described in detail.

Commitment: A device d_{prover} proves its legitimacy to a receiver $d_{verifier}$ when it starts communication with $d_{verifier}$. d_{prover} first sends its identifier id and a commitment x_j to $d_{verifier}$. x_j is calculated as follows.

$$x_j = (-1)^{b_j} {r_j}^2 (\text{mod } n), \tag{2.10}$$

where $r_j \in [0, n)$ and $b_j = \{0, 1\}$ are randomly chosen and kept secret, respectively.

Challenge: After receiving d_{prover}'s id and x_j, a verifier $d_{verifier}$ generates a challenge $\mathbf{e}_j$ as follows and sends $\mathbf{e}_j$ to d_{prover}.

$$\mathbf{e}_j = (e_{1j}, \cdots, e_{kj}), \tag{2.11}$$

where $e_{ij} = \{0, 1\}, i \in [1, k]$ and let $u_j \in [1, k)$ be the maximum number of bits set as 1 in the challenge $\mathbf{e}_j$.

Response: A prover d_{prover} calculates a response y_j against a received challenge $\mathbf{e}_j$. y_j can be calculated as follows:

$$y_j = r_j \prod_{i=1}^{k} s_i^{e_{ij}} (\text{mod } n). \tag{2.12}$$

A prover d_{prover} sends a challenge $\mathbf{e}_j$, public key $\mathbf{v}$, n, and a response y_j to the verifier $d_{verifier}$.

Verification: When a device $d_{verifier}$ receives the above variables, it checks whether the id is correspondent with the hash of $\mathbf{v}$. If it does not match, authentication fails. Otherwise, it verifies whether the received y and the sent $\mathbf{e}_j$ are certainly

corespondent with the following equation.

$$z_j = y_j^2 \prod_{i=1}^{k} v_i^{e_{ij}} (\text{mod } n). \tag{2.13}$$

If $z_j \neq \pm x_j$ or $z_j = 0$, a verifier $d_{verifier}$ judges that d_{prover} does not possess the secret key $\mathbf{s}$ that corresponds with the public key $\mathbf{v}$. Otherwise, a verifier iterates the above procedures, i.e., commitment, challenge, response, and verification phases until d rounds.

2.5 Security Analysis

In this section, we prove that the proposal possesses all of the required properties for ZKP: completeness, soundness, and zero-knowledge properties by following the security analysis given in [78], [163]. Here, completeness property ensures that a legitimate prover can convince a legitimate prover that the prover certainly possesses the secret without fail. Soundness is a property that a prover that does not possess the legitimate secret key against a public key, i.e., an impersonator, cannot prove the possession of the secret with overwhelming probability. Zero knowledge property ensures that a verifier cannot obtain any knowledge about the secret from the series of ZKP protocol. In the following, the notations 'overwhelming' and 'negligible' probabilities are used by following the notation used in [78]. That is, 'overwhelming' and 'negligible' probabilities indicate the probability that exceeds $1 - 1/|n|^b$ for any integer b and the one belows $1 - 1/|n|^a$ for any integer a, respectively.

2.5.1 Completeness

Lemma 2.5.1. *A legitimate prover can convince a verifier without fail.*

Proof. Without loss of generality, always $s_i^2 v_i = \pm 1 (\text{mod } n)$ holds since a prover is legitimate and he/she has s_i that corresponds with v_i for $i \in [1, k]$. Hence Eq. (2.13) that a verifier computes can be transformed into Eq. (2.14). Lemma 2.5.1 has been

proven.

$$
\begin{aligned}
z_j &= y_j^2 \prod_{i=1}^{k} v_i^{e_{ij}} (\text{mod } n) \\
&= \left(r_j \prod_{i=1}^{k} s_i^{e_{ij}} \right)^2 \cdot \prod_{i=1}^{k} v_i^{e_{ij}} \\
&= r_j^2 \prod_{i=1}^{k} (s_i^2 v_i)^{e_{ij}} \\
&= \pm r_j^2 = \pm x_j (\text{mod } n).
\end{aligned} \tag{2.14}
$$

□

2.5.2 Soundness

Lemma 2.5.2. *Against a malicious prover that does not know any secret s_i and cannot compute any square root of $\prod_{i=1}^{k} v_i^{c_{ij}} (\text{mod } n)$ where $c_{ij} = \{-1, 0, 1\}$ but $\boldsymbol{c}_j = \{c_{ij}\} \neq \boldsymbol{0}$ within polynomial time, the probability p that a verifier accepts such a malicious prover satisfies $p < 2^{-d\bar{u}}$, where $d = \Theta(|n|)$ and $\bar{u}$ denotes the mean value of $u_j, j \in [1, d]$ and $\bar{u} = \mathcal{O}(\log |n|)$.*

Proof. The plan for the proof of soundness is as follows. We first show a malicious prover's strategy for convincing a verifier without knowing a secret key $\mathbf{s}$ and calculate its success probability. Then, it is shown that such a prover cannot improve the probability. In order for a malicious prover to convince a verifier that he/she possesses a secret key (without possessing it), one can consider that a prover must forge a commitment x_j and y_j that can satisfy $z_j = \pm x_j$. For this purpose, a malicious prover first guesses challenge $\mathbf{e}_j$ for $j \in [1, d]$ and calculates $\prod_{i=1}^{k} v_i^{e_{ij}} (\text{mod } n)$. Note that anyone can compute this value since $\mathbf{v} = \{v_i\}$ is a public key. Assuming that the prover's guess is correct, he/she randomly choose r_j and forge x_j and y_j as follows.

$$
x_j = \pm r_j^2 \prod_{i=1}^{k} v_i^{e_{ij}} (\text{mod } n), \tag{2.15}
$$

$$y_j = r_j. \tag{2.16}$$

A malicious prover can convince a verifier if and only if every guess of $\mathbf{e}_j$ is correct. However, a commitment x_j must be submitted to a verifier before receiving a challenge $\mathbf{e}_j$, the success probability p can be represented as 1/(the total number of combination of challenges $\mathbf{e}_j$ for $j \in [1, d]$). Here, let p_j denote the probability that a prover can guess a correct $\mathbf{e}_j$. Since there exist at least one and at most u_j bits out of k bits are set as 1 for a $\mathbf{e}_j$, the total number of combination of challenge $\mathbf{e}_j$ is represented as the summation of each combination.

$$\begin{aligned} p_j &= \left\{ \binom{k}{1} + \binom{k}{2} + \cdots + \binom{k}{u_j} \right\}^{-1} \\ &= \left\{ \sum_{i=1}^{u_j} \binom{k}{i} \right\}^{-1}. \end{aligned} \tag{2.17}$$

The probability p can be calculated by the product of each p_j for $j \in [1, d]$.

$$\begin{aligned} p &= p_1 p_2 \cdots p_d \\ &= \left\{ \sum_{i=1}^{u_1} \binom{k}{i} \right\}^{-1} \left\{ \sum_{i=1}^{u_2} \binom{k}{i} \right\}^{-1} \cdots \left\{ \sum_{i=1}^{u_d} \binom{k}{i} \right\}^{-1} \\ &= \left\{ \prod_{j=1}^{d} \sum_{i=1}^{u_j} \binom{k}{i} \right\}^{-1}. \end{aligned} \tag{2.18}$$

Here, without loss of generality, the following inequality holds because $u_j < k$ and $2^{u_j} = \sum_{i=0}^{u_j} \binom{u_j}{i}$ for $j \in [1, d]$.

$$\begin{aligned} \sum_{i=1}^{u_j} \binom{k}{i} - 2^{u_j} = \sum_{i=1}^{u_j} \binom{k}{i} - \sum_{i=0}^{u_j} \binom{u_j}{i} > 0, \\ \Leftrightarrow \left\{ \sum_{i=1}^{u_j} \binom{k}{i} \right\}^{-1} < 2^{-u_j}. \end{aligned} \tag{2.19}$$

By leveraging Eq. (2.19), the probability p is bounded by the $2^{-d\bar{u}}$ by denoting $\bar{u} =$

$\frac{1}{d}\sum_{j=1}^{d} u_j$.

$$p = \left\{ \prod_{j=1}^{d} \sum_{i=1}^{u_j} \binom{k}{i} \right\}^{-1} < 2^{-(u_1+\cdots+u_d)} = 2^{-d\bar{u}}. \tag{2.20}$$

Therefore the success probability of the above strategy is less than $2^{-d\bar{u}}$. Henceforth we prove that such a malicious prover cannot increase the success probability p by considering the case that a prover pre-calculates multiple responses y_j for multiple challenges $\mathbf{e}_j$ for a commitment x_j. Let $(\mathbf{e}'_j, \mathbf{e}''_j)$ and (y'_j, y''_j) denote arbitrary two challenges and responses out of all pairs. Leveraging the fact that $y_j \neq 0$ and a commitment x_j must be submitted before receiving a challenge, the following equation holds:

$$x_j = y'^2_j \prod_{i=1}^{k} v_i^{e'_{ij}} (\text{mod } n) = y''^2 \prod_{i=1}^{k} v_i^{e''_{ij}} (\text{mod } n). \tag{2.21}$$

If a prover was legitimate, he/she could calculate the square root of following equation:

$$\left(\frac{y'_j}{y''_j} \right)^2 = \frac{\prod_{i=1}^{k} v_i^{e'_{ij}}}{\prod_{i=1}^{k} v_i^{e''_{ij}}} (\text{mod } n) = \prod_{i=1}^{k} v_i^{c_{ij}} (\text{mod } n), \tag{2.22}$$

where $c_{ij} \in \{-1, 0, 1\}$. However, a polynomial time prover who does not know the secret key $\mathbf{s}$ cannot compute it since it involves the computation of the square root of $\prod_{i=1}^{k} v_{ij}^{c_{ij}} (\text{mod} n)$. Therefore a malicious prover cannot prepare multiple responses that pass multiple challenges and improve the success probability than p. Thus Lemma 2.5.2 has been proven. □

2.5.3 Zero Knowledge Property

Lemma 2.5.3. *When $d = \Theta(|n|), u_j = \mathcal{O}(\log |n|)$, the proposed scheme has the zero knowledge property.*

Proof. In order to prove that the proposed scheme has the zero knowledge property, it must be shown that a malicious verifier can obtain any information by itself even though he/she does execute the proposed protocol with a (legitimate) prover. This can be usually done with a simulator [78], [163]. The simulator is a series of procedures that an ideal attacker can do. Before defining the simulator for the zero knowledge

property, let $\bar{A}$, $\bar{B}$, and $\widetilde{B}$ denote a legitimate prover, a legitimate verifier, and a malicious verifier that wants to know $\bar{A}$'s secret from it, respectively. Then we consider the following simulator $M_{\widetilde{B}}$:

1. Sequentially execute the following steps from 2 to 6 against $j \in [1, d]$.
2. Guess a challenge $\mathbf{e}'_j$.
3. Calculate a commitment $x_j = \pm r_j^2 \prod_{i=1}^{k} v_i^{e'_{ij}} (\mathrm{mod}\ n)$ and a response $y_j = r_j$.
4. Send a commitment x_j and receive a challenge $\mathbf{e}_j$.
5. If the guessed challenge $\mathbf{e}'_j$ and $\mathbf{e}_j$ differ, go back to the step 2.
6. Otherwise, send a response y_j to the simulator $M_{\widetilde{B}}$.

The information obtained through the above simulator, i.e., $\{\mathbf{x} = \{x_j\}, \mathbf{y} = \{y_j\}, E = \{\mathbf{e}_j\}\}$, is a sequence that a malicious prover can pass the protocol without knowing a secret key $\mathbf{s}$ and is same as the sequence obtained from a legitimate prover.

The simulator $M_{\widetilde{B}}$ can be executed within $2^{u_j} = \mathcal{O}(|n|)$ trials on average for each $j \in [1, d]$. Since $d = \Theta(|n|)$, the entire calculation $2^{u_1} + 2^{u_2} + \cdots + 2^{u_d}$ can be executed with the polynomial time of $|n|$. Therefore a verifier can fully execute the simulator $M_{\widetilde{B}}$ and any information is simulatable by itself. □

2.6 Performance Evaluation

In order to show the effectiveness of the proposed scheme, the appropriate $\mathbf{u} = \{u_j\}$ is calculated for each d that satisfies the required security and most minimizes the expected number of multiplication in the verification. Then, the computation time is measured on an off-the-shelf Android device. Finally we compare the memory requirement and communication cost between the conventional FFS protocol and the proposal.

2.6.1 Appropriate u to Minimize the Number of Multiplication

It is necessary to set minimum u_j while maintaining the required security. We first clarify the meaning of 'the required security'. In the conventional FFS protocol, a malicious prover convinces a verifier with the probability of 2^{-kt} since each element in challenge $\mathbf{e} = \{e_1, \cdots, e_k\}$ can take 0 or 1 and the protocol consists of t rounds. Therefore, to fulfil the required security, the probability that a malicious prover passes the verification, which is p in Eq. (2.18), must be smaller than 2^{-kt}.

$$\left\{ \prod_{j=1}^{d} \sum_{i=1}^{u_j} \binom{k}{i} \right\}^{-1} \leq 2^{-kt}. \tag{2.23}$$

Then, we formulate the expected number of multiplication in the verification. It is noted that the number of multiplication differs when verifying a legitimate prover and malicious prover. This is because a malicious prover cannot proceed the subsequent round unless his/her guess against a challenge is correct. On the contrary, an honest prover should pass all challenges in the verification. Therefore, we calculate four expectation values when verifying an honest and malicious provers for both schemes.

In the conventional FFS protocol, as seen from Eq. (2.9), the number of multiplication is represented as $\frac{k+1}{2} + 1$ because the multiplication occurs every $e_i = 1$ in totally and at least one element must be set as 1, and y is also multiplied. Since a verifier must calculate t rounds' Eq. (2.9) for a legitimate prover, the number of expectation $E_{conv,leg}$ is represented as:

$$E_{conv,leg} = t\left(\frac{k+1}{2} + 1\right). \tag{2.24}$$

A malicious prover cannot proceed the subsequent round unless his/her guess against a challenge is correct. Since the success probability of guessing a challenge is 2^{-k}, the probability that a verifier calculates j-th round's response is represented as

$2^{-k}2^{-k}\cdots = 2^{-k(j-1)}$. Therefore the number of expectation $E_{conv,mul}$ is as follows:

$$\begin{aligned} E_{conv,mul} &= \left(1 + 2^{-k} + \cdots + 2^{-k(t-1)}\right)\left(\frac{k+1}{2} + 1\right), \\ &\approx \frac{1}{1-2^{-k}}\left(\frac{k+1}{2} + 1\right), \end{aligned} \tag{2.25}$$

By letting $\alpha \in [0,1]$ denote a ratio of malicious devices to all devices in a network, the expected value of multiplication in the conventional FFS protocol E_{conv} is represented as

$$\begin{aligned} E_{conv} &= \alpha E_{conv,mul} + (1-\alpha)E_{conv,leg}, \\ &\approx \{t + \alpha(1-t)\}\left(\frac{k+1}{2} + 1\right). \end{aligned} \tag{2.26}$$

The expected value of multiplication in the proposed scheme can be calculated in the same way as the above. In the proposed scheme, as seen from Eq. (2.13), the number of multiplication in challenge $\mathbf{e}_j$ is represented as $\frac{u_j+1}{2} + 1$. Since a verifier must calculate Eq. (2.13) for all d rounds, the number of expectation $E_{prop,leg}$ is represented as

$$E_{prop,leg} = \sum_{j=1}^{d}\left(\frac{u_j+1}{2} + 1\right). \tag{2.27}$$

In contrast, a malicious prover cannot proceed the subsequent round unless his/her guess against a challenge is correct. Since the success probability of correctly guessing a j-th challenge is p_j, the probability that a verifier calculates j-th round's response is represented as $p_1 p_2 \cdots p_{j-1}$. Therefore the number of expectation $E_{conv,mul}$ is as follows:

$$\begin{aligned} E_{prop,mul} &= \left(\frac{u_1+1}{2} + 1\right) + p_1\left(\frac{u_2+1}{2} + 1\right) + \cdots + \left(\prod_{j=1}^{d-1} p_j\right)\left(\frac{u_d+1}{2} + 1\right) \\ &= \frac{u_1+1}{2} + 1 + \sum_{l=2}^{d}\prod_{j=1}^{l-1} p_j\left(\frac{u_l+1}{2} + 1\right). \end{aligned} \tag{2.28}$$

Table 2.1. The relationships between $\mathbf{u}$, p, and $\bar{E}_{prop}$.

(a) $d = 2$

$\mathbf{u}$	p	$\bar{E}_{prop}$
$(1, 6)$	8.3×10^{-7}	4.36
$(2, 4)$	7.7×10^{-7}	4.26
$(3, 3)$	5.5×10^{-7}	4.50
$(4, 2)$	7.7×10^{-7}	4.75
$(5, 2)$	2.2×10^{-7}	5.25
$(6, 1)$	8.3×10^{-7}	5.50

(b) $3 \leq d \leq 5$

d	$\mathbf{u}$	p	$\bar{E}_{prop}$
3	$(1, 1, 4)$	4.0×10^{-7}	4.80
4	$(1, 1, 1, 2)$	6.0×10^{-7}	5.30
5	$(1, 1, 1, 1, 1)$	3.1×10^{-7}	6.05

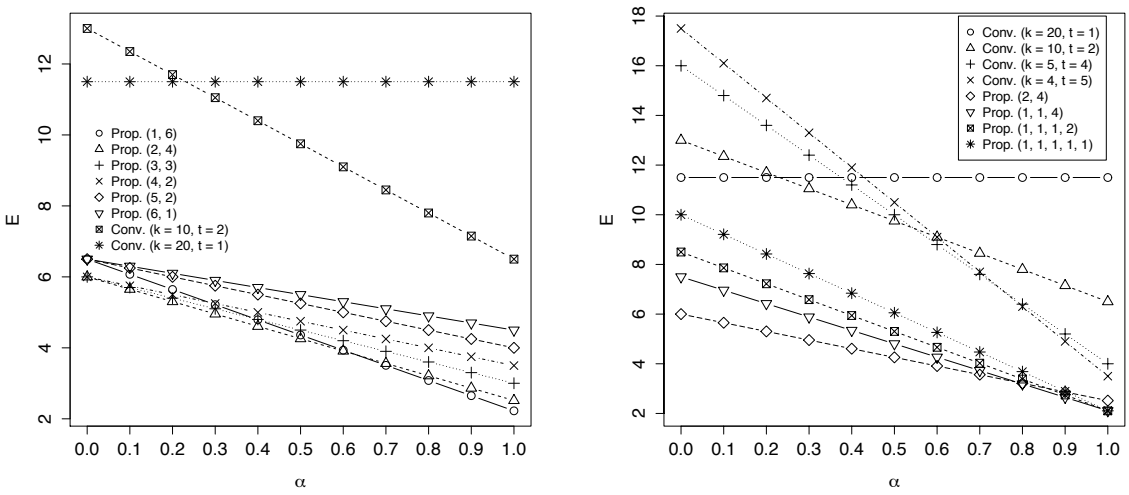
From the above result, the expected value of the multiplication in the proposed scheme E_{prop} is represented as

$$\begin{aligned} E_{prop} &= \alpha E_{prop,mul} + (1-\alpha) E_{prop,leg} \\ &= \frac{u_1+1}{2} + 1 + \sum_{l=2}^{d} \left\{ 1 - \alpha \left(1 - \prod_{j=1}^{l-1} p_j \right) \right\} \left(\frac{u_l+1}{2} + 1 \right). \end{aligned} \tag{2.29}$$

We finally calculate $\mathbf{u}$ that minimizes E_{prop} for each d. For this purpose, k and t in the conventional FFS protocol must be fixed to decide the security requirement in Eq. (2.23). By referring the original FFS protocol in [78], we set $kt = 20$ that achieves $2^{-20} \approx 9.5 \times 10^{-7}$. As seen from Eq. (2.29), α affects the expected value of multiplication. We marginalize α from E_{prop} as represented as Eq. (2.30) and find $\mathbf{u}$ for $d = 2, \cdots, 5$.

$$\bar{E}_{prop} = \int_0^1 E_{prop} d\alpha. \tag{2.30}$$

We first consider $d = 2$. Table 2.1(a) shows the relationships of the candidate $\mathbf{u} = (u_1, u_2)$, the correspondent p, and the marginalized expected value of the multiplication in the proposed scheme. Figure 2-3(a) shows the expected value of multiplication versus α when $d = 2$. In this figure, 'E' represents both E_{prop} and E_{conv} whereas "Prop. (u_1, u_2)" and "Conv. (k, t)" indicate the proposed scheme with $\mathbf{u} = (u_1, u_2)$ and the conventional FFS protocol with k and t, respectively. Although other candidates exist for $\mathbf{u}$, the aim is to reduce the expected value of multiplication and only six candidates that most minimize E_{prop} are shown. From Table 2.1(a), $\mathbf{u} = (2, 4)$ achieves



(a) only $d = 2$ (b) $2 \leq d \leq 5$

Figure 2-3: Expected value of multiplication E versus α.

minimum E_{prop} while maintaining the required security, i.e., $p = 7.7 \times 10^{-7} < 2^{-20}$. As can be seen from Figure 2-3(a), the combination $(2, 4)$ minimizes E_{prop} when $0 \leq \alpha \leq 0.6$. We can also observe that $\mathbf{u} = (1, 6)$ minimizes E_{prop} for $\alpha > 0.6$, i.e., when most of the provers are malicious. Moreover the proposed scheme with $\mathbf{u} = (2, 4)$ reduces the number of multiplication against the conventional schemes $(k, t) = (20, 1)$ and $(k, t) = (10, 2)$ by 48% and 54% when $\alpha = 0$ and 78% and 61% when $\alpha = 1$, respectively.

Then we consider the cases where $d > 2$. Table 2.1(b) shows $\mathbf{u}$ and p that achieve minimum $\bar{E}_{prop}$ for $d = 3, 4,$ and5. In Figure 2-3(b), "Conv. (k, t)" and "Prop. $(u_1, u_2, \cdots)$" indicate the conventional FFS protocol with k and t and the proposed scheme with $\mathbf{u}$, respectively. As can be seen from Table 2.1(b) and Figure 2-3(b), although the proposed scheme with $d \geq 3$ much decreases the number of multiplication when $0.8 \leq \alpha \leq 1.0$, the proposed scheme with $d = 2$ most minimizes the number of multiplication on average. From these results, it can be argued that the proposed scheme effectively reduces the number of multiplication in the verification and the combination of $d = 2$ and $\mathbf{u} = (2, 4)$ is the best choice for the security parameter $kt = 20$.

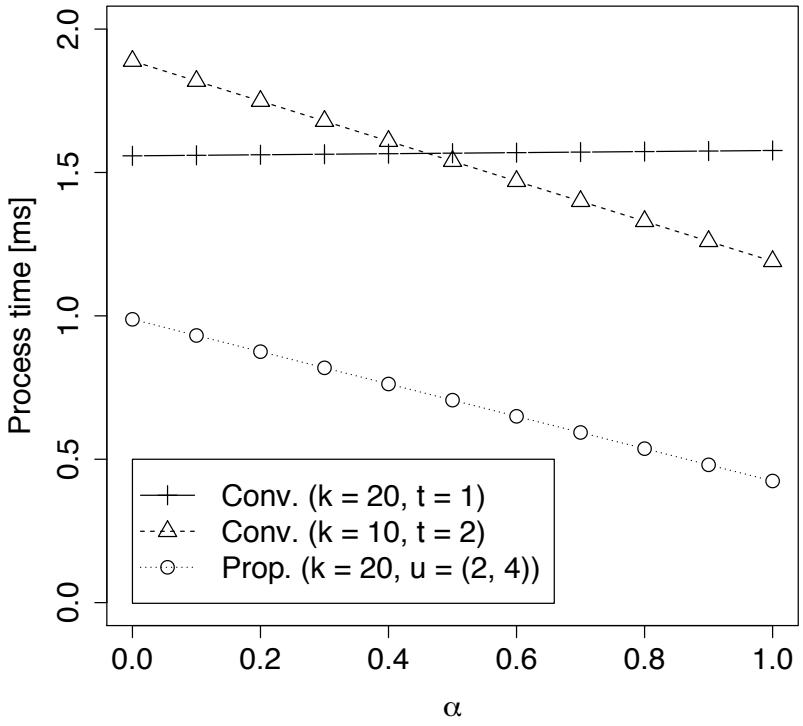


Figure 2-4: Process time for the verification versus α measured on an Android device.

2.6.2 Comparison of Computation Time for Verification on Android device

The effect of the scheme represented in the above is theoretical and one cannot say that the scheme actually reduces the computation complexity on a real device. In order to clarify the effectiveness on a real device, we measure the computation time for the verification on an Android device. Nexus S, which is an Android device with a 1 GHz CPU and 512 MB RAM memory, is used as a verifier device $d_{verifier}$. We simulate both legitimate and malicious provers by varying their probabilities of occurrence, i.e., α, and average measurements observed from 100 trials. From the result of Section 2.6.1, $d = 2$ and $\mathbf{u} = (2, 4)$ are used for the proposed scheme. Since the aim of the proposed scheme is to reduce the verification cost, only the computation time for the verification is measured and it does not include the time for communication between a prover and

a verifier. Figure 2-4 shows the computation time for the verification versus α on a Nexus S. From this figure, the proposed scheme effectively reduces the computation time against the conventional FFS protocol regardless of α. When $\alpha = 0$, the proposed scheme reduces the process time by 37% and 48% against the conventional one with $(k, t) = (20, 1)$ and $(k, t) = (10, 2)$, respectively. When $\alpha = 1$, the proposal also shortens it by 73% and 64%, respectively. However we can see that the reduction rate is not as good as the theoretical one. This is because the proposed scheme requires a trivial cost for generating two challenges and verifying responses one by one.

In Figure 2-4, even if no attacker exists, the number of multiplication and the verification time can be decreased and shorten in our scheme. This is because the required number of combination can be achieved by multiplying that of each round. Therefore even if the number of elements set as 1 is restricted to a small value in each challenge, the required number is quickly reached. Let us consider the following condition: the required number of combination $kt = 20$ and the number of rounds $d = 2$. In this case, $k = 20$ and the upper bounds for each challenge is $(u_1, u_2) = (2, 4)$ in our scheme whereas the previous scheme is $(k, t) = (10, 2)$. On the one hand, the expected number of multiplication for a legitimate device in the previous scheme $E_{conv,leg}$ can be calculated as follows.

$$E_{conv,leg} = \left(\frac{10+1}{2} + 1\right) + \left(\frac{10+1}{2} + 1\right) = 13. \tag{2.31}$$

On the other hand, that of our scheme $E_{prop,leg}$ is calculated as follows.

$$E_{prop,leg} = \left(\frac{2+1}{2} + 1\right) + \left(\frac{4+1}{2} + 1\right) = 6. \tag{2.32}$$

Therefore, even if a prover is legitimate, the process time can be reduced as well.

2.6.3 Memory Requirement

We finally compare the required memory amount for the conventional FFS protocol and the proposed scheme. Let M_{conv} and M_{prop} denote the required memory amount for the conventional FFS protocol and the proposed scheme, respectively. Since the conventional scheme requires t commitments, challenges, responses and a prover's public key $\mathbf{v}$ and n, M_{conv} can be represented as

$$\begin{aligned} M_{conv} &= t|n| + tk + t|n| + k|n| + |n| \\ &= (k + 2t + 1)|n| + tk. \end{aligned} \tag{2.33}$$

When $|n| = 1,024$ bits and $(k, t) = (20, 1)$, $M_{conv} = 2,964$ Bytes. M_{prop} can be also calculated in the same way as M_{conv}.

$$\begin{aligned} M_{prop} &= d|n| + dk + d|n| + k|n| + |n| \\ &= (k + 2d + 1)|n| + dk. \end{aligned} \tag{2.34}$$

When $|n| = 1,024$ bits and $(k, d) = (20, 2)$, $M_{prop} = 3,240$ Bytes. From this result, the proposed scheme increases memory amount by 9% ($\approx \frac{3,240-2,964}{2,964}$).

2.7 Conclusions

We have proposed a provably secure lightweight verification scheme in FFS protocol. The basic idea to reduce computation cost in the verification is to make most of elements in a challenge $\mathbf{e}$ to 0 when generating a challenge. To avoid lowering security, a challenge is divided into multiple ones and the number of elements set as 1 in each challenge is restricted. Since a prover must pass every challenge, the proposed scheme can achieve the sufficient security by setting appropriate upper bounds for each challenge. The proposed scheme is proved as ZKP by referring the security analysis in [78]. By the theoretical computation, we have shown that the number of division $d = 2$ and upper bounds for each challenge $\mathbf{u} = (2, 4)$ is the lowest computation on the verification for the security parameter $kt = 20$. The proposed scheme reduces the

number of multiplication by 48-54% and 61-78% when no malicious provers exist and when only malicious provers exist, respectively. We have also represented that the scheme is effective on an Android device.

Chapter 3

Unsupervised Clustering-based SPITters Detection Scheme in VoIP Service

3.1 Introduction

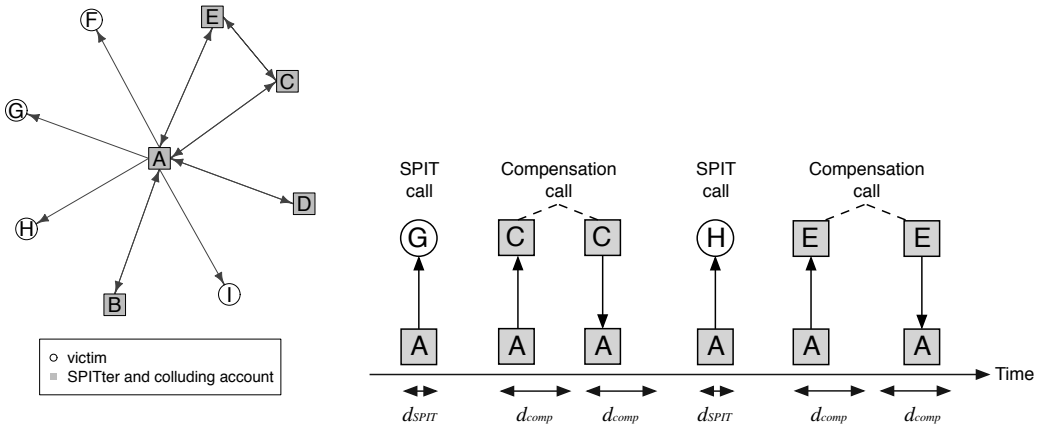
Recently, VoIP (Voice over IP) is becoming a major telephony protocol thanks to inexpensive call charge. Unfortunately, the merit that we can call at very low calling rate is also beneficial for a caller who spreads ads, malicious phishing, or persistent survey. This unsolicited call is referred to as SPIT (SPam over Internet Telephony) and a SPIT call or SPITters (SPIT callers) detection system must be implemented in SIP servers on service providers. One of the major challenges in SPIT detection is that we cannot judge the legitimacy of a call before a callee takes it and thus spam detection scheme in e-mail or text-based chat services cannot be applied. Although the content of calls cannot be used for SPIT detection, the service providers are able to collect CDR (Call Detail Records) which are call logs of each caller. By leveraging CDR, several calling features which distinguish SPITters from legitimate callers, e.g., call frequency, average call duration, out-degree, and in-degree, can be calculated and they are useful for the SPITters detection. Although many feature-based detection

scheme exist, e.g., [79], [82], [87], [88], [164], none of them provide reasonable solutions to set the threshold and reference models in order to differentiate legitimate callers and SPITters appropriately. In other words, if a new feature is found, it is difficult to integrate it into their schemes. In addition, it may be infeasible to obtain training data labeled as "SPITter" or "legitimate caller" in privacy concerns since the content of calls must be checked to confirm whether the caller is legitimate to obtain label. These two shortcomings motivate us to research a flexible SPITters detection approach without threshold-based detection as an unsupervised detection method.

In this thesis, we propose an unsupervised and threshold-free SPITters detection scheme by using a clustering algorithm. The proposed scheme turns complex threshold setting and training into clustering the callers and identifying the SPITters cluster. The aim is to separate the inspected callers into two clusters, one is the legitimate cluster and the other is the SPITters one by using multiple features. Since the scheme leverages the features to find the dissimilarity among the callers, any complex threshold settings and training phases can be avoided. Although clustering itself does not give us the SPITter cluster, we can identify which cluster is the "SPITters cluster" by comparing the average of a feature, e.g., calls per day.

The effectiveness of the proposed scheme depends on a dissimilarity measure and a clustering algorithm. We compare three combination of dissimilarity measures and clustering algorithms, namely (i) k-means clustering, (ii) the ED (Euclidean Distance) + PAM (Partitioning Around Medoids) clustering, and (iii) RF (Random Forests) dissimilarity + PAM clustering. By computer simulation, we show that the combination of RF + PAM is the best choice for SPITter detection since RF effectively finds the importance of features while the ED does not. We also clarify that the scheme works well when the SPITters account for more than 20% of entire callers. We compare the proposed scheme with the well-studied conventional schemes and the proposed RF + PAM scheme outperforms them in terms of true positive rate while maintaining low false positive rate.

The rest of this chapter is structured as follows. Section 3.2 describes the model of SPITters. Related work is summarized in Section 3.3. The proposed scheme is



(a) Relationship of SPITter with colluding accounts.

(b) Time series of a SPITter with colluding accounts.

Figure 3-1: Graphical model of SPITter with colluding accounts.

described in Section 3.4. Simulation results and evaluation are discussed in Section 3.5. Finally the conclusions are shown in Section 3.6.

3.2 SPITters Model

SPIT is voice-based spam for advertising merchandise or to commit fraud, such as credit card fraud, to deceive someone to deposit his/her money into their bank accounts. Many researchers define "SPITter" as automatic computer-based SPIT calls generator, e.g., [165]–[168]. The content of SPIT is pre-recorded and automatically played when the call is successfully connected. Most works deal with SPITters who disperse many SPIT calls and calls only victims. However, to our knowledge, there is no statistics of SPITters. Hence, it is necessary to consider other models of SPITters who own multiple accounts. By using multiple accounts, the SPITters can lower the call frequency, compensate for short average call duration, and make more human-like relationships. Figures 3-1(a) and 3-1(b) show an example of the relationship and call behavior of SPITters with colluding accounts, respectively. In Figure 3-1(a), node A is a SPITter and A has four colluding accounts, B, C, D, and E. On the other hand, F, G, H, and I are victim callees. Arrows indicate the direction of calls, i.e., SPITter A and the colluding accounts have bi-directional links but the victim callees have

Table 3.1. Parameters for SPITter model.

(a) without colluding accounts

PARAMETER	VALUE
# of SPIT calls per day	10, 50, 100, 500, and 1,000
SPIT call duration	$d_{\text{SPIT}} \sim Exponential(\mu_{\text{SPIT}} = 15sec)$
callee	uniformly chosen
call back rate	0.01

(b) with colluding accounts

PARAMETER	VALUE
# of SPIT calls per day	10, 50, 100, 500, and 1,000
SPIT call duration	$d_{\text{SPIT}} \sim Exponential(\mu_{\text{SPIT}} = 15sec)$
callee	uniformly chosen
call back rate	0.01
# of colluding accounts	5
compensation call duration	$d_{comp} \sim Exponential(\mu_{comp})$

uni-directional link from A. From Figures 3-1(a) and 3-1(b), it can be seen that a SPITter with colluding accounts can compensate for short average call duration by occasionally calling back and forth with colluding accounts for a certain duration. Let d_{comp} denote average call duration to compensate for short call duration. We discuss how to set d_{comp} in the following section. By preparing some colluding accounts, a SPITter can imitate the call behavior of a legitimate caller. Tables 3.1(a) and 3.1(b) show the call model of SPITters with and without colluding accounts, respectively. In the following section, we discuss how to set the parameters in Tables 3.1(a) and 3.1(b).

- Call frequency (calls per day)

  In order to determine the call frequency model of SPITters, we estimate the maximum amount of SPIT calls per account. In the assumption, the upper bound of calls per day is estimated under the following situation. SPITters are assumed to make calls between 9 - 5. This assumption comes from the following two reasons. The first reason is that SPITters expect the callees to catch as many calls as possible and thus SPIT calls are generated while the

most callees are active. The second reason is that most SPIT calls may be generated by a corporation and traditional business hour of corporation is 9 - 5. In reality, human-based SPITter can call as many as 1 or 2 calls per minute: therefore, estimated upper bound per account is 500 or 1,000 calls per day. However, it is too frequent compared with legitimate callers: thus, SPITters are easily detected by a threshold-based SPITter detection [79]. By using multiple accounts, SPITters may avoid a threshold-based SPITter detection. Hence, we also consider a low frequency SPITter model that disperses SPIT calls to as many as 10, 50, and 100 calls per day.

- Call duration for SPIT call

  For most callees, a received SPIT call must be unsolicited and this could result in a shorter call duration than a legitimate one. Hence the mean duration of SPIT call μ_{SPIT} is set as 15 sec and SPIT call duration d_{SPIT} obeys exponential distribution. This is the same as other related works [169] and [87].

- Callee selection

  Since SPITters aim to broadcast their merchandises or deceive as many victims as possible, SPITters are assumed to seldom call to the same callee again and thus the callees are randomly chosen.

- Call back rate

  It could be possible that callees intentionally or unintentionally call back to a SPIT caller. Some callees may be interested in the content of SPIT calls. As an unintentional case, some callees cannot catch the call and may call back later without knowing that the caller is a SPITter. We consider the above cases and set the call back rate as 0.01.

- Number of colluding accounts

  We assume that a SPITter can prepare as many as five colluding accounts. This is because a SPITter can forge ST (Strong Ties) property, which is a ratio of total call duration of the top 5 callees to the total call time.

- Call duration for compensation call

  In order to compensate for the short average call duration and the ST property, a sophisticated SPITter can easily forge them by calling with five colluding accounts back and forth. A SPITter with colluding accounts makes not only SPIT calls but also an outgoing call and an incoming call with each colluding account once a day. Thus, if a SPITter prepares five colluding accounts, he/she makes and receives as many as five compensation calls toward/from colluding accounts daily. This can be the case since the calling rate is very low in a VoIP/SIP environment. We set the mean duration for the compensation call as follows. Let d_{target} be the target average call duration by compensation calls and d_{target} is calculated as follows.

$$d_{target} = \frac{\mu_{\text{SPIT}} \times \#(\text{SPIT calls}) + \mu_{comp} \times \#(\text{compensation calls})}{\#(\text{SPIT calls}) + \#(\text{compensation calls})}, \qquad (3.1)$$

  where # (X) denotes the number of X. For instance, when the $d_{target} = 60$ sec and the number of SPIT calls is 100 calls per day, μ_{comp} can be calculated as follows.

$$\begin{aligned} 60 &= \frac{15 \times 100 + \mu_{comp} \times 5}{100 + 5}, \\ \mu_{comp} &= 960 \text{ sec.} \end{aligned} \qquad (3.2)$$

  Note that when the number of SPIT calls is very large, e.g., 1,000 calls per day, μ_{comp} is the upper bound since the most active time (9-5) is spent for SPIT calls. Hence the compensation calls can be exchanged within 16 hour (5-9) and the upper bound of $\mu_{comp} = 96$ min (16 hour / (2 $\times$ 5)). In this case, d_{target} is calculated from (3.1) and $d_{target} \approx 44$ sec. Therefore, each μ_{comp} is calculated subject to an upper bound of d_{target} for a variable number of SPIT calls.

3.3 Related Work

Many researches try to detect SPITters in VoIP services. There exist three research categories: (1) features-based SPITters detection scheme, e.g., [79], [82], [87], [88], (2) SPITters detection based on social network trustworthiness, e.g., [91], [93], [96], [98] and (3) content-based SPIT detection, e.g., [101]–[104]. However, social network-based SPITter detection and content-based detection have some limitations. In order to calculate the trustworthiness of callers, most social network-based detection schemes construct a graph whose vertices and edges indicate callers and relationships between callers. If all callers (vertices) are within same VoIP/SIP provider, the graph can be constructed from their call history. However, some callers can belong to other providers and thus the graph might be incomplete and it might lead to insufficient trustworthiness. Also, content-based detection schemes cannot avoid privacy concerns. On the other hand, feature-based SPIT detection schemes are not influenced by the above problems, since the features of each caller can be calculated from its own CDR. Therefore, we summarize the feature-based SPITters detection schemes in the following.

Shin *et al.* propose Progressive Multi Gray-leveling (PMG) which is a call frequency based SPIT caller detection [79]. They calculate the two gray levels of callers: one for short-term gray level and the other one for long-term gray level. Whether to connect a call is decided by whether the summation of these two levels exceeds its threshold or not. If the summation falls below a certain threshold, the connection is made; otherwise, the connection is blocked. This scheme uses call frequency to distinguish the SPIT callers from legitimate ones. By combining short-term and long-term gray level, a high frequency SPIT caller remains over the threshold and are detected as a SPIT caller. This method needs tuning as many as five parameters.

Yang *et al.* propose the supervised decision tree-based SPITters detection [170]. They used six features, which are the number of callees it sends out, ratio of number of calls outgoing and incoming, number of total calls, and number of failed, canceled and completed calls in order to classify the callers. They use labeled training data to

construct the decision tree.

Bai *et al.* point out that there is a fundamental difference between legitimate users and spammers on making and receiving calls [82]. A legitimate caller typically makes and receives calls, while a spammer makes a large number of calls but seldom receives calls. Apparently, a small ratio of answered calls and dialed calls can be used to distinguish a legitimate caller and a spammer. Based on the above analysis, they propose three features to identify spam calls, Interaction Ratio (IR), which is the ratio of answered calls to the dialed calls, Historical Ratio (HR), which is the ratio of repeated calls to distinct calls, and Social Ratio (SR), which is the ratio of unknown callees to the total number of callees. They set thresholds X, Y, and Z for IR, HR, and SR, respectively. They compare each threshold with the corresponding features one by one when the caller initiates a call in order to check the legitimacy of a caller.

Bokharaei *et al.* propose some features to separate unusual callers from a real phone call dataset in North America [164]. They show that most legitimate callers in their dataset spend most of their talk time with only 4-5 people and they refer to this feature as the ST property. Thus, an ST property is the ratio of the total call duration of the top 5 callees to the total call time. In addition, for most callees, the received SPIT calls must be unsolicited and this could result in shorter call durations than legitimate one. They take advantage of this feature by defining the Weak Ties (WT) property, which is the fraction of callees that talk for more than 60 sec. The WT value must be very small for SPIT callers since the estimated average SPIT call duration must be shorter than 60 sec. By using these features, they can filter suspicious accounts in their dataset. They introduce F (say 90%) as the threshold against ST and WT and identify the common outstanding callers of ST and WT property as SPITters.

Sengar *et al.* propose two SPIT detection methods [87]. In the first approach, they detect high frequency and low call durations callers as SPITters. They prepare the common reference model of legitimate caller whose call arrival $\sim Poisson$(180 sec) and call duration $\sim Exponential$(60 sec). In this approach, they check whether an inspected caller calls five calls within 15 min and if true, they calculate the Maha-

lanobis distance of the call duration between each inspected caller and the common reference model using the recent n observations. If the distance deviates from the trained threshold, the initiating call is rejected. The second approach focuses on the entropy of the call duration aggregated from the entire call flow. Since most callees soon hang up SPIT calls, the call duration of a SPIT caller is skewed towards a shorter duration and brings about low entropy. Thus, the second approach can detect whether SPIT calls occur in the network.

Wang *et al.* propose call/receive ratio and normalized call frequency based features CI and F_{CD} which are input into the k-means clustering algorithm [88]. The scheme finds the center mass of a legitimate callers and classifies each caller by comparing the distance between the caller and a common reference model with the trained threshold.

3.3.1 Shortcomings in Conventional SPITters Detection Schemes

Although there exist many features to distinguish SPITters, all methods must set the threshold or select the reference models in order to differentiate legitimate callers from SPITters. In other words, if a newly feature is found, it is difficult to integrate it into SPIT detection since thresholds for each feature must be individually set. Amanian *et al.* propose to weigh each feature by inferring the effectiveness of the features [90], however, this scheme still cannot avoid threshold-based detection. Although, classification-based machine learning approaches can deal with multiple features, e.g., decision tree-based detection [170], may solve the problem, the training phase is necessary for such classification algorithms. Typically, it is difficult to obtain training data labeled as "SPITter" or "legitimate caller" due to privacy concerns since the content of the calls must be inspected to check whether a caller is legitimate. The above two shortcomings motivate us to research an unsupervised SPITters detection approach which does not require any threshold setting nor training data.

3.4 Proposed Scheme

Here, we propose an unsupervised and threshold-free SPITters detection scheme by using a clustering algorithm. The proposed scheme turns complex threshold setting and training problems into clustering the callers and identifying the SPITters cluster which is overall much easier. The idea of our scheme is to separate the callers into two clusters, one is legitimate cluster and the other is SPITters based on multiple features. In other words, the features are used not to directly trap SPITters but to find the dissimilarity among callers. This way avoids the complex threshold tuning and training phase prevalent in the conventional schemes. Although clustering itself does not give us the SPITter cluster, we can identify which cluster is a "SPITters cluster" by comparing the average of a feature, e.g., calls per day, calculated within each cluster. That is, we leverage the fact that the call duration of SPITters is relatively short compared to legitimate ones and the call frequency of a SPITter is relatively higher than legitimate ones.

The classification accuracy of the proposed scheme highly depends on the combination of dissimilarity measure and clustering algorithm. We introduce three combinations, (i) k-means clustering, (ii) the ED + PAM clustering, and (iii) RF dissimilarity and PAM clustering.

The proposed scheme is superior to the conventional schemes because of the following reasons.

- *It does not suffer from complicated thresholds tuning.*

  This is because the proposed scheme does not use any thresholds in order to distinguish each inspected caller and thus it is much easier to implement in testbed than traditional schemes.

- *It can easily and reasonably adopt new features into a SPITters detection system.*

  This is true because if a more superior feature were found, most conventional works would tune the threshold of the feature again. In contrast, the proposed scheme can easily involve such a feature since the features are used to find the dissimilarity among callers.

- *It does not change the existing SIP message format nor modify in any way the SIP terminal.*

  This is because the proposed scheme needs only the CDRs of each inspected caller and thus it does not change any SIP format or terminal. It is also an important point for implementation.

- *It does not delay SIP connection.*

  This is true since our scheme can be executed as an off-line process of the VoIP/SIP service. A SIP server judges whether a caller is legitimate or not by simply checking the classified list.

3.4.1 System Model

Before describing the operation of the scheme, we define the system model in order to clarify where and when the operation is executed, and what information is used. A SPITters detection system is deployed in a VoIP/SIP service provider and our task is to identify SPITters who belong to its own VoIP/SIP service provider. The SPITters detection scheme is executed at regular intervals, say once a day, and any calls are rejected until the next SPIT detection phase if the caller is judged as a SPITter. As many as $N_{callers}$ callers exist in the VoIP/SIP service provider and they involve both legitimate callers and SPITters. The task of a detection system is to correctly classify each legitimate and SPITter by using N_{days} CDR of each user. This simple construction avoids any complicated procedures during the call establishment and thus it does not delay the SIP connection. In addition, our scheme is also applicable for mobile environment since only call logs are required for SPITters detection.

3.4.2 Procedures of SPITters Detection

Next, the operation of our scheme will be explained. The proposed scheme consists of following three steps, namely (i) calculating calling pattern, (ii) clustering callers based on calling features, and (iii) identifying the SPITters cluster.

Table 3.2. CDR of a caller for $N_{days} = 7$ days.

DATE[DD/MM/YYYY H:M:S]	CALLER/CALLEE	DIRECTION	DURATION [s]
01/01/2013 12:02:32	sip:eve@bar.com	outgoing	34
01/01/2013 13:40:21	sip:dave@foo.com	incoming	45
...	...	...	...
07/01/2013 21:07:35	sip:dave@foo.com	outgoing	285

Table 3.3. Example of feature vectors.

CALLER	f_{ACD}	f_{CPD}	f_{ST}	f_{WT}	f_{IOR}
Alice	119.65	2.86	0.69	0.61	0.72
Bob	104	6.25	0.66	0.52	0.43
Carrol	61.17	507.12	0.85	0.02	0.1

1. *Calculating calling pattern*

   At the first step, calling features, which are considered to differentiate SPITters from legitimate callers, are calculated for each caller.

2. *Clustering callers based on calling features*

   The calling features calculated at the first step are used to cluster callers. By inputting them into a clustering algorithm, each caller is clustered into two classes.

3. *Identifying the SPITters cluster*

   Two clusters are obtained in the second step, which are the SPITters and legitimate callers clusters, and it is necessary to correctly identify which cluster is the SPITter cluster. In order to do that, we leverage the typical characteristics of SPITters. For example, SPITters are assumed to more frequently call than legitimate callers.

Calculating Calling Pattern

We explain the first step of operation, namely calculating calling pattern. The calling features represent the calling pattern of caller behavior and are calculated from the CDR. Table 3.2 shows a simple CDR example of a caller. As many as $N_{features}$

features are calculated for each caller from the CDR for the latest N_{days} days. Let $\boldsymbol{f}$ denote a feature vector of a caller. $\boldsymbol{f}_i$ is occasionally used to indicate caller i's feature vector. In our setting, $N_{features} = 5$ features are used to describe each caller, i.e., $\boldsymbol{f} = (f_{\mathrm{ACD}}, f_{\mathrm{CPD}}, f_{\mathrm{ST}}, f_{\mathrm{WT}}, f_{\mathrm{IOR}})$ since they are considered to be effective in distinguishing SPITters from legitimate callers. Note that if an effective feature was found, it is easily integrated into the feature vector. Table 3.3 shows an example of feature vectors. Now each caller's calling behavior is represented as a feature vector. We cannot conclude that the five features (f_{ACD}, f_{CPD}, f_{ST}, f_{WT} and f_{IOR}) are enough to characterize each caller's pattern. Certainly, other features can be also considered, e.g., missed call ratio, how calls are distributed throughout the day, and the distribution of call hour. However, since the aim of the proposed scheme is to show how to use multiple effective features without complex threshold tuning and any training phase, we use features that have already been proven to be effective in the literature.

- Average Call Duration (ACD)

  The average call duration is a fundamental feature for the SPITters detection. Since most SPIT calls are unsolicited, this could result in shorter call duration than legitimate ones. ACD is calculated as follows.

$$f_{\mathrm{ACD}} = \frac{\sum(\text{duration} \mid \text{duration} > 0 \;\&\&\; \text{direction} == \text{outgoing})}{\#(\text{duration} > 0 \;\&\&\; \text{direction} == \text{outgoing})}, \tag{3.3}$$

  where (X|Y) denotes values X which satisfies conditions Y and #(Y) denotes the number of entries which satisfies conditions Y, respectively.

- Call frequency Per Day (CPD)

  The call frequency is also a fundamental feature for SPITters detection since most SPIT calls make more frequent calls than legitimate callers. CPD is calculated as follows.

$$f_{\mathrm{CPD}} = \frac{\#(\text{duration} > 0 \;\&\&\; \text{direction} == \text{outgoing})}{N_{days}}. \tag{3.4}$$

- Strong Ties property (ST)

The ST property characterizes the fact that most of legitimate callers spend most of their talk time to only 4-5 people [164]. The ST property is the ratio of the total call duration of the top 5 callees to the total call time.

$$f_{\mathrm{ST}} = \frac{\sum(\text{duration} \mid \text{callee} == \text{top 5 callees})}{\sum(\text{duration})}, \tag{3.5}$$

where the "top 5 callees" indicates the five most frequent callees.

- Weak Ties property (WT)

  For most callees, the received SPIT calls must be unsolicited and this could result in shorter call durations than legitimate calls. The WT property is the fraction of callees that talk for more than 60 sec [164]. The WT property must be very small for SPIT callers since the estimated average SPIT call duration must be shorter than 60 sec.

$$f_{\mathrm{WT}} = \frac{\#(\text{callee} \mid \overline{\text{duration}} > 60 \text{ sec})}{\#(\text{callee})}, \tag{3.6}$$

  where $\overline{\mathrm{X}}$ denotes the average value of feature X.

- Incoming/Outgoing Ratio (IOR)

  A legitimate caller typically makes and receives calls, while a spammer makes a large number of calls but seldom receives a call. Hence, we can leverage the ratio between incoming calls and outgoing calls as a feature for discrimination. Although many features characterize this fact, e.g., IR, HR, SR [82], CI [88], the ratio between outgoing and incoming calls [170], BDR (Bi-Directional Ratio) and IOR [171], we only use IOR in this research. Both BDR and IOR focus on "the number of callees" instead of "the number of outgoing calls" used in IR, HR, SR, and CI. Because of this, BDR and IOR are more robust against colluding SPITters. In addition, as seen from the result of [171], it is not necessary to use both BDR and IOR since these are very similar properties. f_{IOR} is calculated

as follows.

$$f_{\text{IOR}} = \frac{\#(\text{Incoming})}{\#(\text{Incoming} \cup \text{Outgoing})}, \tag{3.7}$$

where Incoming = (callee | direction == incoming) and Outgoing = (callee | direction == outgoing).

After the features are calculated for all callers, each feature is normalized in order to make the features all the same scale. Without normalizing, relatively small value feature will be ignored by large value one. For example, f_{ST} ranges $[0, 1]$ but f_{CPD} can be $[0, 1, 000]$. In this case, f_{ST} is almost ignored because of the difference of magnitude.

Clustering Callers with Calling Features

Since the calling features are basically to distinguish SPITters and legitimate ones, we consider the features can be used to cluster callers into two clusters, namely the SPITters cluster and legitimate callers one. After calculating calling features for each caller, we cluster the inspected callers into two clusters by using a clustering algorithm. A clustering algorithm bundles objects (in our case, callers) who resemble each other. Hence, callers who have similar feature values resemble each other and are bundled into the same cluster.

Although there exist mainly three major clustering algorithms, namely hierarchical clustering, k-means algorithm, and k-medoids algorithm, we introduce the k-means and k-medoids algorithms. This is because most hierarchical clustering algorithms are memory and time exhaustive (the complexities are $\mathcal{O}(N^3_{callers})$ for agglomerative hierarchical clustering or $\mathcal{O}(2^{N_{callers}})$ for divisive hierarchical clustering) thus they are not suitable for large datasets. Note that it is enough to use only one clustering algorithm, i.e., the k-means or k-medoids clustering algorithms, but both are introduced for comparison.

k-means clustering:

We first introduce the k-means clustering [172]. The k-means clustering is origi-

Algorithm 1 k-MEANS CLUSTERING FOR THE PROPOSED SCHEME.

1: Randomly select centroids which are two $N_{features}$-dimensions points and indicate the center of cluster 1 and 2, respectively.
2: **while** No centroids changed **do**
3: Assign each caller ($\boldsymbol{f}_i$) to the closer centroid.
4: Recalculate the two centroids by using newly assigned callers.
5: **end while**
6: Assign the callers which are closer to centroid 1(2) with cluster 1(2).

nally to cluster objects into k clusters and, in our case, $k = 2$ because callers should be clustered into the SPITters cluster and legitimate callers one. Algorithm 1 shows the algorithm which has been implemented. The final goal of the k-means clustering is to find two centroids which are the center points of each cluster and to assign each caller to the nearer centroid. At first, two randomly generated feature vectors are assigned as centroids 1 and 2. Then each caller's feature vector is assigned to a more closer cluster 1 or 2. For example if a caller's feature vector is closer to centroid 1, it is assigned to cluster 1. After all callers are assigned to a cluster either 1 or 2, two centers of gravity is calculated within each cluster and centroids are replaced by them. The algorithm iterates these procedures until no centroids change.

The merit of k-means is fast and scalable since the calculation time is $\mathcal{O}(kN_{callers})$. However, at step 3 in Algorithm 1, the ED is the only choice for the dissimilarity measure since it has to calculate the distance between callers and center "points" of each cluster. The ED cannot weigh each feature and thus it does not consider how each feature contribute for clustering.

k-medoids clustering: Algorithm 2 shows PAM clustering which is the most classical implementation of the k-medoids clustering algorithm [173]. The basic procedures are almost same with the k-means clustering. However, in contrast to k-means, PAM selects a caller's feature vector as the center of cluster. This enables us to use variate dissimilarity measures other than the ED at step 6 of the algorithm.

On the other hand, the complexity of PAM is $\mathcal{O}(kN^2_{callers})$ and more exhaustive than k-means. In order to mitigate the complexity, CLARA (Clustering LARge Application) [173] or CLARANS (CLARA based upon RANdomized Search) [174]

Algorithm 2 PAM clustering for the proposed scheme.

```
Randomly select two callers as medoids.
while No medoids changed do
    Assign each caller to the closer medoid.
    for each medoid l ∈ {1, 2} do
        for each non-medoid vector m do
            Swap l and m and calculate the cost based on dissimilarity measure.
            Select the situation with the lowest cost.
        end for
    end for
end while
Assign the callers closer to medoid 1(2) with cluster 1(2).
```

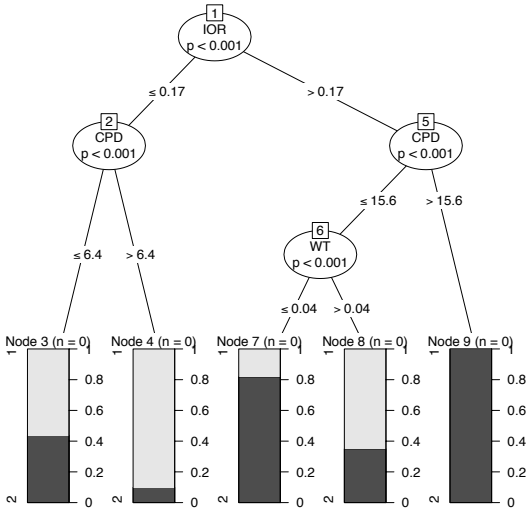


Figure 3-2: Example of a decision tree.

can be substituted, which are both the derivative algorithms of PAM.

RF dissimilarity

Since any dissimilarity can be used in PAM clustering, we introduce another novel one, RF dissimilarity. RF is one of the ensemble decision tree-based classifiers [175]. Figure 3-2 shows an example of decision tree. Basically, a tree has several splits and a features vector is input to the root of tree and goes down to the terminal node. Finally, the vector is classified as a class labelled at the terminal node. The reason why RF dissimilarity is introduced is that it considers the importance of features during tree construction. Although RF is originally a classification algorithm, it

Table 3.4. Outcome of clustering.

CALLER	CLUSTER
Alice	1
Bob	1
Carrol	2

Table 3.5. Labeled callers.

CALLER	CLUSTER
Alice	legitimate caller
Bob	legitimate caller
Carrol	SPITter

outputs the similarity among callers while constructing decision trees and thus the similarity can be used as a dissimilarity measurement. In RF, as many as T decision trees are constructed and M_{try} out of $N_{features}$ features are randomly chosen as the split criterion. The intuition of RF similarity is that if the two feature vectors of caller i and caller j are input into the root of a decision tree and both land in the same leaf node, it can be inferred that both are similar to a certain extent and $S_{i,j}$, which is the similarity between caller i and caller j, is increased by one. After decision trees are constructed, each caller's similarity is output and normalized by the number of trees. Thus a similarity-matrix among inspected callers is obtained and each element $S_{i,j}$ of the matrix takes a $[0, 1]$ value. Then the dissimilarity among caller i and j is obtained as $\sqrt{1 - S_{i,j}}$. Since feature vectors are unlabeled, it is necessary to construct decision trees with only unlabeled data. In order to solve this problem, researchers propose to label given callers as class 1 and generate "synthetic" callers from the given callers which are labeled as class 2 [175], [176]. The synthetic data are used to generate as many as $N_{callers}$ by randomly sampling from the product of the empirical marginal distributions of the features. In Figure 3-2, the bar plots underneath the figure indicate the ratio of the class 1 to the class 2 callers who land in the terminal node. In constructing a decision tree, since the feature which is the most distinctive split criterion is selected at each node, the dissimilarity obtained by RF implicitly weighs each feature according to the importance of classification. In the later section, we will compare the RF dissimilarity with the ED by means of computer simulation.

Table 3.6. Conditions to identify the SPITter cluster with single feature.

FEATURE	CONDITION
CPD	If the average of CPD in cluster 1 is bigger than that in cluster 2, we can judge cluster 1 as the SPITter cluster.
ACD	If the average of ACD in cluster 1 is less than that in cluster 2, we can judge cluster 1 as the SPITter cluster.
ST	If the average of ST in cluster 1 is less than that in cluster 2, we can judge cluster 1 as the SPITter cluster.
WT	If the average of WT in cluster 1 is less than that in cluster 2, we can judge cluster 1 as the SPITter cluster.
IOR	If the average of IOR in cluster 1 is less than that in cluster 2, we can judge cluster 1 as the SPITter cluster.

3.4.3 Identifying SPITters' Cluster

After clustering the callers, a list whose entries are labeled as cluster 1 or cluster 2 is obtained. Table 3.4 is an example of the list. Since the aim of SPITters detection is to identify which callers are SPITters, it is necessary to decide which cluster is SPITters one. Table 3.5 shows an example of the result of identified clusters. The proposed scheme accomplishes this by comparing the average value of a feature, e.g., f_{CPD} within each cluster. Certainly, we stated that f_{CPD} cannot be simply used to trap every type of SPITters but it does not mean that all SPITters call less frequently. In most cases, high frequency SPITters typically exist: thus a hypothesis that SPITters call more frequently than legitimate callers may still hold. In other words, if the clusters are successfully made, the tendency that SPITters call more frequently than legitimate callers may be observed even though some SPITters are low frequent SPITters. For this reason, the higher f_{CPD} cluster is labelled as the SPITter cluster. This way solves the problem of complex threshold tuning, as found in the conventional schemes. Here we introduce f_{CPD} as the representative feature to identify the SPITters cluster. Equation (3.8) denotes the average f_{CPD} in cluster k.

$$\overline{f_{\mathrm{CPD}}}^{(k)} = \frac{1}{N_k} \sum_{i=1}^{N_k} f_{CPD,i}^{(k)}, \tag{3.8}$$

where N_k denotes the number of callers in cluster k. For instance, if $\overline{f_{\mathrm{CPD}}}^{(1)}$ is larger than $\overline{f_{\mathrm{CPD}}}^{(2)}$, cluster 1 is labelled as the SPITter cluster. Although we use f_{CPD} as the feature to identify the cluster, other features can also do the job. Table 3.6 shows each condition of feature to identify the SPITters cluster. In order to confirm the legitimacy of our idea, we evaluate the classification accuracy by changing the feature to identify the SPITters cluster in Section 3.5.3.

After the above steps, the callers list whose entries are labeled as "SPITter" or "legitimate caller" is obtained. Finally, the list is disseminated to its own SIP servers and is used for deciding if the call should be established.

3.5 Performance Evaluation

In order to show the efficiency of the proposed scheme, several performance metrics are evaluated by means of computer simulation. We use a dataset which contains real legitimate caller's call logs [177] and self-generated SPIT caller's call data for evaluation. In the evaluation, each caller is classified into "SPITter" or "legitimate caller" in an off-line fashion: thus, actual VoIP/SIP messages are not simulated. We use R version 3.0.2, the randomForest package [178] for the Random Forests classifier, and the PAM clustering package [179] to implement the proposed scheme. All simulations are executed on an off-the-shelf computer which has 3.4 GHz quad cores CPU and 16 GB RAM. Each result is obtained by repeated simulation with as many as 100 trials.

The performance metrics include the classification accuracy and computation time. We first evaluate the classification accuracy. More specifically, TP (True Positive rate) and FP (False Positive rate) are measured against the number of days, the ratio of SPITters to the entire caller, the SPITter model, and each feature. Here, TP denotes the ratio of correctly identified SPITters and FP rate is the ratio of mistakenly identified legitimate callers as SPITters, respectively. We compare three combinations of dissimilarity and clustering algorithm, (i) the k-means clustering, (ii) ED + PAM: the ED as the dissimilarity with PAM clustering, and (iii) RF +

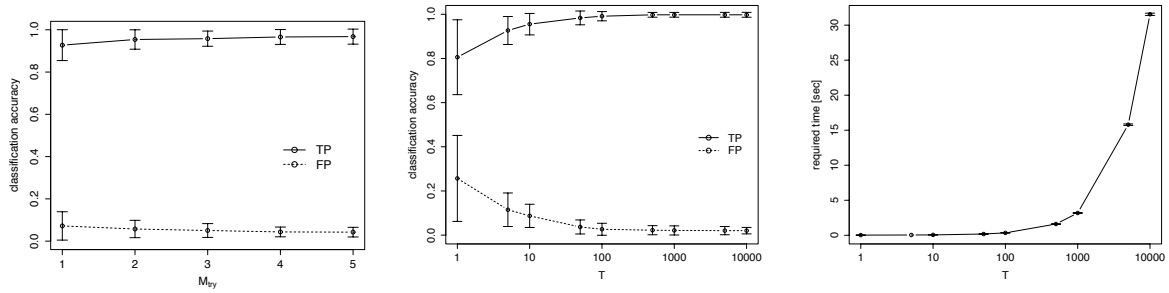
PAM: the RF dissimilarity with the PAM clustering. We compare them with two conventional schemes [79], [164]. We then measure the computation time to evaluate scalability.

3.5.1 Dataset

Before showing the performance metrics, we describe the dataset used for evaluation. We use the Reality Mining dataset as the legitimate caller's call data, which includes 94 callers' call data as collected by the MIT Media Lab [177]. In this dataset, 68 were colleagues working in the building on MIT campus (90% graduate students, 10% staff) while the remaining 26 callers were incoming students at the university's business school. Although there exist two other call log datasets [180], [181], MIT Media Lab's Reality Mining dataset is the best choice for evaluation. In [180], the datasets are based on anonymized Call Detail Records (CDR) of phone calls and SMS exchanges between five million of Orange's customers in Ivory Coast between December 1, 2011 and April 28, 2012. However, the dataset is available for the NetMob[1] conference and cannot be used for other purposes. The other one, Nodobo [181], consists of smartphone usage logs of 27 students in a Scottish state high school but 27 persons' data are too small to evaluate the performances of the proposed schemes. Although the Reality Mining dataset is not VoIP call logs but on mobile phone call logs, it can be assumed that VoIP takes place in a conventional telephony network and thus call characteristics of VoIP call is the same as that of mobile phone telephony. In contrast, to the best of our knowledge, no SPITter's dataset is publicly disclosed. Hence we artificially generate SPITters call logs from the model described in Section 3.2 and then mix them with the legitimate callers dataset for evaluation.

We randomly choose 100 callers (including both legitimate callers and SPITters) in the simulation except for Section 3.5.4. Let $R_{SPITters}$ denote the ratio of SPITters to all callers. We set $R_{SPITters} = 0.2$ if not stated otherwise. That is, we randomly select 80 legitimate callers and 20 SPITters. It is a similar setting to [98] which considers 25% of all callers as SPITters. It seems to excessive to consider the cost to

[1]http://www.netmob.org/



(a) Classification accuracy versus M_{try}.

(b) Classification accuracy versus T.

(c) Required time in RF versus T.

Figure 3-3: Parameters tuning for RF + PAM.

obtain SIP addresses. However, nowadays, many SIP service providers emerge that offer free SIP accounts, e.g., Call Centric[2], Voiptalk[3] and Onsip[4]. In addition, as the cost of obtaining e-mail accounts has dropped in the last decade, the cost of obtaining VoIP/SIP accounts might drop due to price competition in the near future. Due to the above reason, it can be the case that SPITters account for 20% of all callers. However, since no VoIP/SIP calls/callers statistics are publicly available, we cannot guess how many SPITters account for all callers and conclude whether the setting $R_{SPITters} = 0.2$, is reasonable. Therefore, we vary $R_{SPITters}$ from 1% to 50% and evaluate the classification accuracy.

3.5.2 Parameter Tuning for RF

Before the simulations, we tune the two parameters M_{try} and T to find the RF dissimilarity and use them for the following evaluation. In this setting, we use 7 days' call logs.

At first, M_{try} is tuned under the condition of $T = 25$ which is the recommended choice for tuning T [175]. Figure 3-3(a) shows classification accuracy versus M_{try}. Since five features are used, M_{try} can take from 1 to 5. From this figure, we can see that both TP and FP get slightly better as M_{try} gets larger. In our situation,

[2]http://www.callcentric.com/
[3]https://www.voiptalk.org
[4]http://www.onsip.com/about-voip/sip/sip-account

the number of features is relatively small and thus we use $M_{try} = 5$ in the following simulation.

We also tune T, which is the number of generated trees. Figure 3-3(b) shows the classification accuracy versus T. From this figure, it can be seen that both TP and FP get better as T gets larger. More specifically, our scheme achieves TP = 0.95 and FP = 0.014 when $T = 500$ but both TP and FP are consistent when $T \geq 500$. Figure 3-3(c) shows the computation time to find dissimilarities among callers versus log-scaled T. From this figure, we can see that the calculation time linearly increases when T gets larger. From these results, it is no use to set $T > 500$ by considering the balance between calculation time and accuracy: therefore, $T = 500$ is used in the following simulation.

3.5.3 Performance Evaluation of the Classification

In this section, we evaluate TP and FP under the several conditions. We first show the comparison of features to identify the SPITters cluster. The aim of this is to clarify that any feature can be used to identify the SPITters cluster. Second, we also evaluate TP and FP versus the number of days N_{days} to clarify how many days are required to successfully identify SPITters. Third, TP and FP are evaluated against the ratio of SPITters to all callers. It is important to know how much SPITters are required for the classification since we use a clustering. Fourth, it is shown the characteristics of each feature. In this evaluation, we use only one single feature for the classification. Finally, we evaluate how accurately our scheme classifies each SPITter model.

Performance of Chosen Feature to Identify the SPITter Cluster

We first compare features to identify the SPITter cluster. We use the RF + PAM scheme where $T = 500$ and $M_{try} = 5$, $N_{callers} = 100$, and $R_{SPITters} = 0.2$ and follow the condition of each feature to identify the SPITter cluster in Table 3.6. From this figure, the proposed scheme can achieve nearly the same TP and FP regardless of the

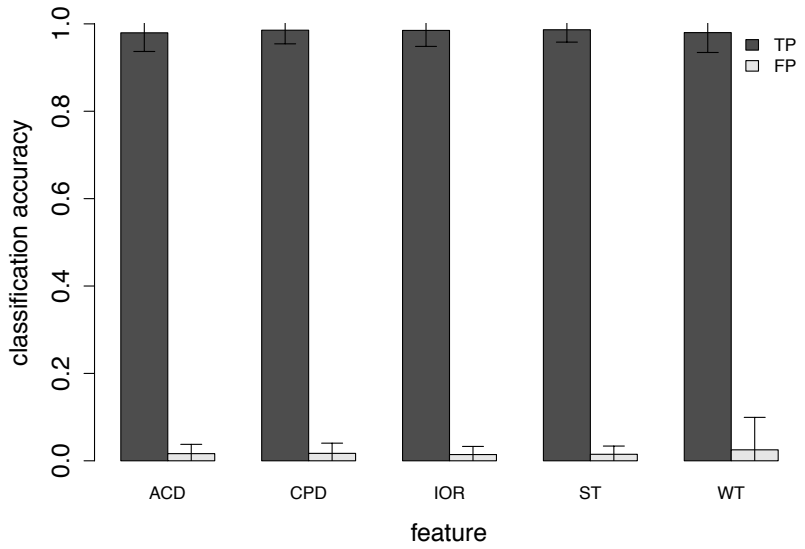
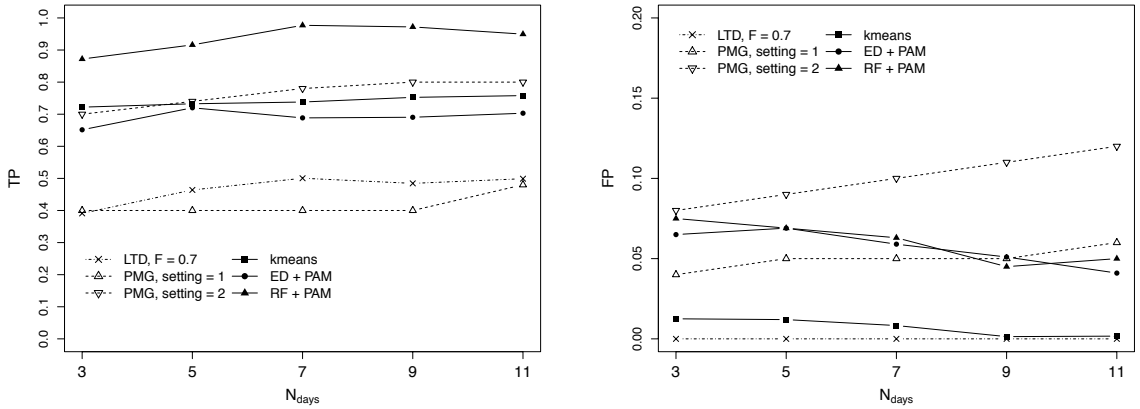


Figure 3-4: TP and FP versus a chosen feature to identify the SPITter cluster.

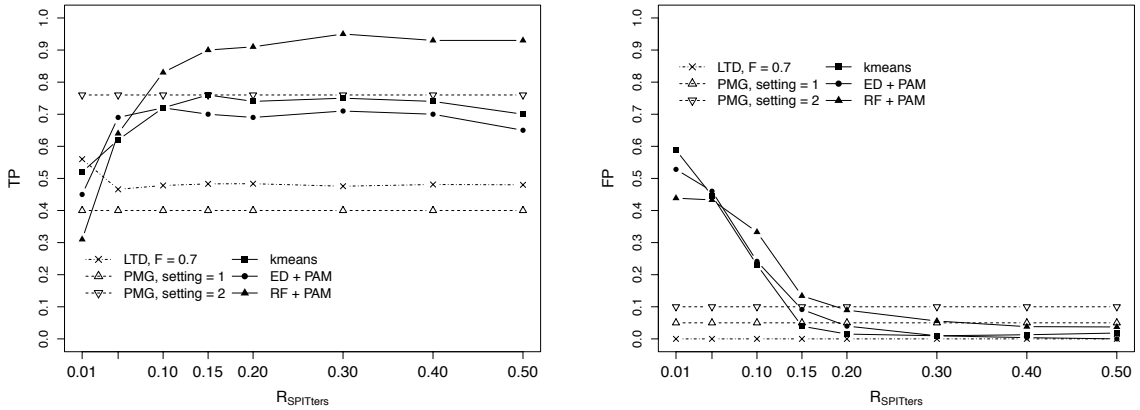
chosen feature and thus any feature can be used to identify the SPITter cluster. From this result, it can be concluded that the SPITter cluster is identified by comparing the average of any single feature if the clustering is successfully done.

We then show the TP and FP against N_{days}. We compare our methods with the conventional schemes LTD [164] and PMG [79]. The reason why LTD and PMG are chosen for comparison is that both methods are well-studied works and the authors gave clear descriptions for deciding the threshold setting and algorithm. Although there exist many other feature-based SPIT detection methods, e.g., [170], [82], [87], and [88], they give ambiguous setting for the threshold, reference model, or training method: thus, we do not compare them in order to avoid inaccurate comparisons. Both LTD and PMG need threshold tuning. Although LTD needs one parameter F and $F = 0.9$ is suggested in their original work, $F = 0.7$ is used which give more accurate detection against our dataset. For PMG, five parameters $TL1, TL2, C1, C2$, and T need to be tuned. Two settings for the parameters are suggested in their work [79]. Setting 1 is $(TL1, TL2, C1, C2, T)$ = (1 min, 1 hour, 3, 1, 1,000) and setting 2 is (10 min, 1 day, 1, 1, 1,000), respectively. Note that since PMG tries to detect SPIT calls but not callers when the call is going to be established, a caller is regarded as a



(a) TP (b) FP

Figure 3-5: TP and FP versus N_{days}.



(a) TP (b) FP

Figure 3-6: TP and FP versus $R_{SPITters}$.

SPITter once PMG detects a SPIT call.

Figure 3-5(a) shows the TP with varying collection periods (N_{days}). From this figure, it can be seen that TP slightly and gradually improves as N_{days} gets longer, regardless of schemes. This result is intuitive and understandable since it has more chance to trap SPITters as the collection period gets longer. We can also see that RF dissimilarity gives better TP than the ED by comparing RF + PAM, ED + PAM and k-means. This is because the ED does not consider the importance of features in finding dissimilarity. We then discuss that the performance of PMG is sensitive to threshold settings. Since PMG has to select as many as five thresholds

and parameters, it is very difficult to obtain the optimal setting. LTD achieves nearly 50% TP when $N_{days} = 7$ and does not improve anymore. This is because LTD detects the callers whose ST and WT both deviate from normal and thus it cannot detect the SPITters with colluding accounts since they try to approach the values of ST which are close to those of legitimate callers.

Figure 3-5(b) shows the FP against N_{days}. From this figure, we can see that PMGs, especially setting 2, get worse as the collection period gets longer. This is because callers are judged as SPITters once a call is judged as SPIT and the FP is gradually worse as N_{days} gets longer. It is also found that the FP of the proposed scheme gradually decreases. In contrast to PMG, the longer the collection period gets, the more accurate classification can be executed in the proposed scheme. This is the same reason why the TP of the proposed scheme gets better.

From Figure 3-5, it can be concluded that the proposed scheme works well even though SPITters are low-frequent ones and/or with colluding accounts.

We show the TP and FP against $R_{SPITters}$. Since our methods rely on how legitimate callers and SPITters resemble each other, the performance of classification depends on how much SPITters accounts for inspected callers. $R_{SPITters}$ is varied from 1% to 50% for the evaluation. We use not only the Reality Mining dataset but also the Nodobo dataset [181] which consists of 27 students' smartphone call logs in a Scottish state high school. This is because we cannot prepare the sufficient number of legitimate callers when $R_{SPITters} < 10\%$ since the Reality Mining dataset involves only 90 callers' log. Therefore, we randomly sample $100(1 - R_{SPITters})$ legitimate callers out of 117 (= 90 + 27) persons from both datasets. Figure 3-6 shows TP and FP versus $R_{SPITters}$. From these figures, we can see that both TP and FP of the proposed schemes get better as $R_{SPITters}$ increases. Especially, RF + PAM outperforms the others when $R_{SPITters} \geq 20\%$. In contrast, for $R_{SPITters} < 15\%$, the TP in the proposed schemes gets worse as $R_{SPITters}$ becomes lower. From Figure 3-6(b), the tendency can be also seen for the FP. We consider there are two reasons why the proposed schemes get worse against low $R_{SPITters}$. The first reason is that classification itself cannot cluster well since the most of callers are legitimate when $R_{SPITters}$ is low.

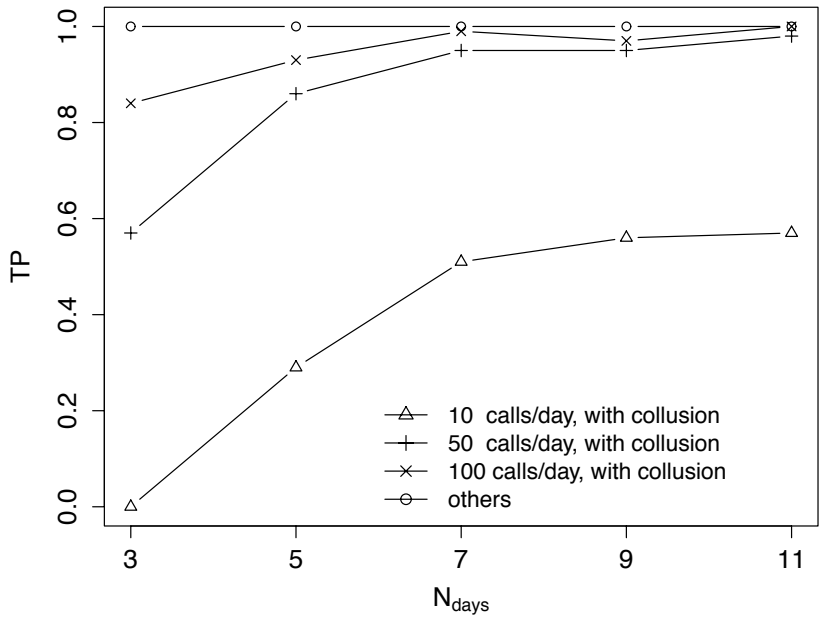


Figure 3-7: TP versus SPITTer types.

The second reason is that the representative feature mistakenly identifies the SPITTer cluster when the chosen SPITTers are only sophisticated SPITTers. This situation can occur when both $N_{callers}$ and $R_{SPITters}$ are too low or few. When $R_{SPITters} = 0.01$, only one SPITTer exists in the inspected callers since $N_{callers} = 100$. It cannot be concluded that whether the proposed scheme is robust against low $R_{SPITters}$ and bigger $N_{callers}$ since the performance is not evaluated with a sufficient number of legitimate callers' call log.

Performance Comparison for SPITTer Types

We then discuss which SPITTer types are difficult to be identified. The ten types of SPITTers discussed in Section 3.2 evaluated. These include both low-rate (as low as 10 calls per day) and high-rate SPITTers (as much as 1,000 calls per day) and with or without collusions. In this simulation, we use only RF + PAM as the core algorithm. Figure 3-7 indicates TP versus each SPITTer model. In Figure 3-7, the results of seven types are merged as "others", which are SPITTers without collusions and high call rate SPITTers ($f_{\text{CPD}} \geq 500$) with collusions, since they are correctly identified without fail. From this figure, it is obvious that low-rate SPITTers ($f_{\text{CPD}} \in [10, 100]$)

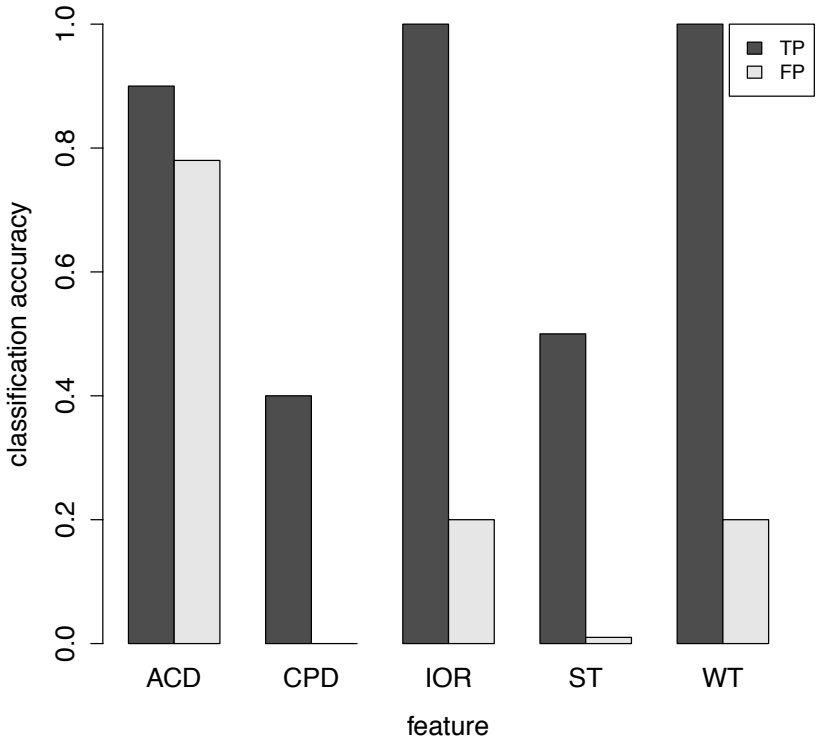


Figure 3-8: Classification performance with a single feature.

with colluding accounts are difficult to be detected. This is because these models are more similar to legitimate and ordinary callers. However, even against such SPITters, the TP gradually improves as time goes on.

Comparison of Each Feature

We clarify how each feature affects to classify legitimate callers and SPITters. In order to clarify the efficiency of each feature, only a single feature is used. In this evaluation, k-means clustering is chosen for a clustering algorithm. Figure 3-8 shows the TP and FP against each single feature. In addition, in order to ease the discussion, we show the statistics of each feature by SPITters and legitimate callers in Table 3.7. We discuss each result from ACD to WT one by one. In the result of ACD, most of legitimate callers are classified as SPITters. Typically, the ACD of SPITters is shorter compared with the legitimate callers. However, since we assume the SPITters

Table 3.7. Statistics of each feature by the type of callers.

(a) SPITters

Statistics	ACD	CPD	ST	WT	IOR
Minimum	13.17	8.14	0.010	0.0000	0.000
1st Quartile	15.46	49.53	0.040	0.0200	0.010
Median	25.19	97.07	0.465	0.0200	0.010
Mean	67.37	323.61	0.474	0.0296	0.018
3rd Quartile	85.58	492.36	0.910	0.0300	0.020
Maximum	238.28	980.43	0.980	0.1300	0.110

(b) Legitimate callers

Statistics	ACD	CPD	ST	WT	IOR
Minimum	1.00	0.000	0.4500	0.0000	0.0000
1st Quartile	56.51	2.330	0.7700	0.2700	0.3000
Median	94.11	4.830	0.9100	0.5000	0.4300
Mean	140.63	5.929	0.8655	0.4509	0.4157
3rd Quartile	178.25	7.640	0.9950	0.6050	0.5000
Maximum	839.23	26.000	1.0000	1.0000	1.0000

with colluding accounts, the mean value of ACD within SPITters is more than 60 sec as can be seen from Table 3.7. In addition, almost 25% of legitimate ones call less than 60 sec on average. Therefore, many legitimate callers are mistakenly classified as SPITters. We then discuss the result of CPD. TP when only CPD is used is 0.4 and relatively low value whereas FP is almost 0. This is because some of the SPITters high frequently call, e.g., a hundred of calls, and low frequent SPITters are classified as legitimate callers. The similar results are obtained for ST. As seen from Table 3.7, the ST of the legitimate caller is more than 0.45 and the median of ST within the SPITters is 0.465. From this result, a half of SPITters is classified as legitimate callers. Since the value of ST is easily increased with collusion accounts by calling with five colluding accounts, it is not useful when such sophisticated SPITters exist. The results of IOR and WT are similar: TP is almost 1 while FP is almost 0.2. As seen from Table 3.7, both SPITters' values of WT and IOR are significantly lower than legitimate callers' mean values of them and thus the TP is good. In contrast, some of legitimate callers' WT and IOR are low. Therefore such callers are classified

Table 3.8. Required time in our methods.

	DISSIMILARITY		CLUSTERING	
$N_{callers}$	RF	ED	PAM	k-means
100	1.6E-1 s	2.4E-4 s	8.7E-4 s	6.4E-4 s
1,000	3.1 s	6.7E-3 s	5.3E-2 s	1.5E-3 s
10,000	1.5E+2 s	8.3E-1 s	6.6 s	9.8E-3 s
100,000	9.4E+4 s	1.2E+1 s	4.6E+2 s	1.2E-1 s

as SPITters.

3.5.4 Computation Time

Finally, the computation time of our schemes is measured by varying $N_{callers}$ between 100 and 100,000. We measure the time spent for calculating the dissimilarity and clustering, respectively since our schemes mainly consume the most of time to calculate dissimilarity and clustering. Table 3.8 shows the calculation time of the proposed scheme. From this table, we can see that the required time follows the time complexity for PAM ($\mathcal{O}(N_{callers}^2)$) and k-means ($\mathcal{O}(N_{callers})$), respectively. In addition, RF requires the highest calculation time and about three order of magnitude than the ED. RF takes about three hours to find the dissimilarity among 100,000 callers. This might be a problem if we consider a larger VoIP/SIP service provider and a calculation reduction algorithm might be necessary.

3.6 Conclusions

We have proposed an unsupervised SPIT callers detection with a clustering algorithm. The proposed scheme turns complex threshold setting and training problems into clustering the callers and identifying the cluster. In contrast to the conventional schemes, calling features are used to find the dissimilarity among callers and this avoids the threshold tuning and training phases. By computer simulation, it has been concluded that the proposed scheme with RF dissimilarity and PAM clustering outperforms the conventional schemes in terms of classification performance when

SPITters account for more than 20% of inspected callers. We have also shown that the proposed scheme can tolerate as many as 100,000 callers with an off-the-shelf computer.

Chapter 4

Secure Products Distribution Scheme in RFID-enabled Supply Chains

4.1 Introduction

RFID (Radio Frequency IDentification) technology is getting much attention in supply chains to ease many complicated operations e.g., traceability, recall problem, and quality management. In RFID-enabled supply chains, a manufacturer attaches RFID tags into products and ships toward distributors or retailers. In order to identify the detail of items, an EPC (Electronic Product Code) is attached to each item. Many EPC formats e.g., SGTIN (Serialized Global Trade Item Number), SSCC (Serial Shipping Container Code), CPI (Component and Part Identifier), or GID (General ID), are defined by GS1 [182]. For example, SGTIN and SSCC are used to assign an identifier to a product and a container, respectively. However, counterfeit in the RFID-enabled supply chains is an open issue in the industry and the academia due to the nature of RFID: the RFID reader can freely interrogate tags [121], [122]. This could be a problem once genuine tags are interrogated in the public area by an attacker, he/she can create counterfeits that have genuine EPCs. The OECD (Organisation for Economic Co-operation and Development) announced that the counterfeit goods in international trade could amount about $250 billion in 2007 [183]. Therefore, the anti-counterfeit technology is an urgent demand for RFID-enabled supply chains.

Many schemes have been proposed to protect EPCs from illegal interrogation by an attacker [121]–[128], [184]. For example, Juels *et al.* proposed to encrypt EPCs with a symmetric encryption scheme and to distribute an encryption key by splitting it into multiple shares and writing them to tags with a (τ, n) secret sharing scheme [156]. The secret sharing scheme realizes that one can extract the key if he/she can obtain more than τ unique shares out of n shares [157].

The above schemes are effective to avoid EPCs from being leaked to an attacker who tries to interrogate them during the public transportation. However, an attacker can repeatedly try each key candidate without being detected by any party. After an attacker interrogates all tags, he/she can try every key candidate to obtain genuine EPCs. Although it has great importance to limit the number of illegal attempts and grasp the existence of an attacker, to our knowledge, no remedy exists.

In this thesis, we propose an illegal interrogation detectable products distribution scheme with an authentication server in RFID-enabled supply chains. The idea of our scheme is to detect an attacker at an authentication server. Our scheme generates random sequences and XORs with each EPC. EPCs masked with random sequences are written into genuine tags on products whereas the random sequences are placed on an authentication server with an access code. An access code is divided into shares with a secret sharing scheme and shares are also written into tags on products. We also prepare dummy tags and write bogus shares into them. Since an attacker who wants to know genuine EPCs may obtain a large number of access code candidates, he/she must try each on an authentication server to obtain random sequences. This construction not only avoids genuine EPCs from being revealed, but also limits the number of illegal authentication attempts and detects an attacker on the server. We prove that our construction is secure against both *privacy* attacker that tries to discover genuine EPCs and *robustness* attacker that tries to tamper with the contents of tags. We implement our scheme with off-the-shelf RFID devices and computer to clarify its latency.

The rest of this chapter is organized as follows. The preliminaries are described in Section 4.2. We summarize related work in Section 4.3, respectively. The proposed

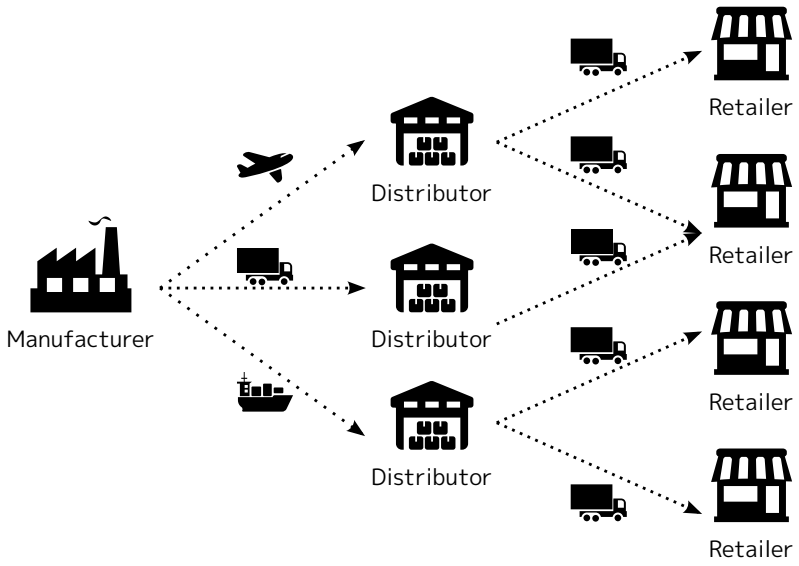


Figure 4-1: Example of supply chains.

Table 4.1. Example of SGTIN-96 EPC.

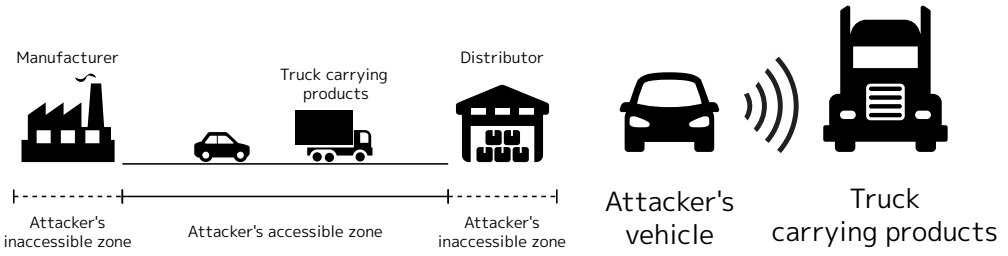
Header (8 bits)	Filter Value (3 bits)	Partition Value (3 bits)	Company Prefix + Item Reference (44 bits)	Serial Number (38 bits)
00110000	010	001	1001...0	1101...1

scheme is described in Section 4.5. Security analysis is shown in Section 4.6. We show performance evaluation in Section 4.7. We conclude our discussion in Section 4.8.

4.2 Preliminaries

4.2.1 System Model

Figure 4-1 shows an example of RFID-enabled supply chains. A manufacturer produces, composes, and ships products toward distributors. A manufacturer also generates an EPC to each product and attaches it to a product. Table 4.1 shows an example of EPC format, which is referred to SGTIN-96. As can be seen from this table, the information, such as product company and a serial number, is involved in the EPC of SGTIN-96. After distributors receive products, they decompose cases and recompose products to deliver them to retailers. Finally, retailers stock and sell them to customers. Every party possesses RFID readers and interrogates tags when



(a) Attacker's accessible zone.

(b) Example of attacker's position in the accessible zone.

Figure 4-2: Attacker's accessible area.

they arrive and leave to manage products flow.

4.2.2 Attacker Model

An EPC involves very sensitive information e.g., a product code, a company code, or serial number. Although UHF RFID system facilitates many functions, e.g., traceability, quality management, or recall problem, it also brings about a privacy issue. That is, tags attached to products could be interrogated when they are conveyed in the public transportation system e.g., highway. Figure 4-2 shows the attacker model we assume in this research. The attacker can interrogate all tags' memories but cannot tamper it unless he/she knows the genuine access password of tags. The goal of the attacker is to obtain genuine EPCs of products.

4.3 Related Work

In order to avoid EPCs from being revealed to an attacker in the RFID-enabled supply chains, it is effective to write encrypted EPCs instead of plaintext one [156]. Though an encryption key must be distributed toward a partner side, many researchers solve this problem by splitting an encryption key into multiple shares with a secret sharing scheme and writing into tags on products [121]–[128]. A secret sharing scheme realizes that one can extract a secret if he/she can obtain more than τ unique shares out of n shares [157]. Shamir proposed the first secret sharing scheme based on the polynomial interpolation over the finite Galois Field [157]. The intuition behind this scheme is

that one can determine a $(\tau - 1)$ degree polynomial curve with more than or equal to τ points on the curve while one cannot determine it when less than $\tau - 1$ points are given. More specifically, let $f(x)$ denote a secret and a degree $\tau - 1$ curve over finite field $\mathbb{Z}_q$ where q is a prime. $f(x)$ can be constructed as follows.

$$f(x) = s + a_1 x + a_2 x^2 + \cdots + a_{\tau-1} x^{\tau-1}, \tag{4.1}$$

where $s \in GF(q)$ denotes a secret and $a_i \in GF(q), i \in [1, \tau - 1]$ denotes coefficients, respectively. Then, n points $(x_i, f(x_i)), i \in [1, n]$ on Eq. (4.1) are called as 'shares'. With more than or equal to τ points on $f(x)$, the intercept of $f(x)$, i.e., the secret $s = f(0)$ can be extracted by Lagrange interpolation as follows:

$$f(x) = \sum_{j=1}^{\tau} L_j(x) f(x_j), \tag{4.2}$$

where

$$L_j(x) = \prod_{l=1, l \neq j}^{\tau} \frac{x - x_l}{x_j - x_l}. \tag{4.3}$$

By using Shamir's secret sharing, Langheinrich and Marti first proposed a secret sharing based EPC distribution scheme [121], [122]. Juels *et al.* adopted the Reed-Solomon ECC (Error Correcting Code) based secret sharing scheme [185] instead of Shamir's one to reduce the size of shares and to enable a party to recover a key even some shares are erased or in error [156]. Lv *et al.* pointed out that Shamir's and Reed-Solomon ECC based secret sharing schemes are computationally heavy due to the multiplication and division over the high degree finite field [124], [125]. In order to reduce the computation cost, they proposed a secret sharing scheme that only requires XOR and addition operations.

Many researchers also point out that the content of tags is unchanged throughout the supply chains and thus tags can be tracked by an attacker. Therefore, many schemes have been proposed to securely update the contents of tags e.g., written shares and encrypted EPCs. Cai *et al.* proposed a tag-reader authentication scheme to securely update the contents of tags [123]. Although this scheme realizes the secure

update of tags contents, it needs modification on tags and an extra hash value. Alfaro *et al.* proposed another approach to securely update the contents of tags by using a proactive (τ, n) secret sharing scheme [126]. Abughazalah *et al.* proposed to use two keys, one for cases' tags and the other for products' tags [128]. If the distributor ships tag-attached products, it newly generates the keys for cases, re-encrypts cases' EPCs, and divides the cases' keys with the secret sharing scheme to avoid an attacker from tracking products.

4.4 Shortcomings on Conventional Secure Product Distribution Schemes

The conventional schemes are effective to avoid genuine EPCs from being leaked to an attacker. However, we point out that there are mainly three issues in the conventional schemes. The first one is that no scheme can notice whether tags have been interrogated by an attacker and thus he/she can unlimitedly try each key candidate and might eventually find a correct key that decrypts EPCs. This can be a serious problem since an attacker can try to decrypt EPCs after all tags are interrogated without being detected. Therefore, it has a great importance to limit the number of attacker's attempts and to detect the existence of an attacker.

The second one is that even though a key is split into shares and written into multiple tags, an attacker can recover the key when he/she collects sufficient (more than or equal to τ) shares. In general, the secret sharing based unidirectional key distribution is secure against a so-called "hit-and-run" attacker, which cannot interrogate sufficient shares [121], [123]. However, in reality, an attacker may be able to easily collect desired shares since a hundred of tags could be interrogated within a second. Therefore the assumption of an attacker is too weak in the conventional schemes and it is necessary to propose a more secure scheme even if an attacker can collect all shares. Moreover, if an attacker can access and tamper more than or equal to $n-\tau+1$ tags, the distributor that receives products cannot recover the key without

fail. Therefore, in order to deploy a unidirectional key distribution scheme in the real supply chains, it is important to propose a new key distribution scheme against an attacker who can collect unbound shares rather than the limited number of shares.

The third one is that an encrypted EPC may not fit into the EPC memory block, since there exist several EPC lengths (even user-defined variable length). Let us consider the following case: a manufacturer would like to encrypt EPCs with AES (Advanced Encryption Standard) with a 128 bits key. In this case, since the minimum block size of AES is 128 bits, the encrypted EPC takes a multiple of 128 bits and thus it cannot be written into the 96 bits or 198 bits EPC memory bank. Blowfish has many block length options though it cannot completely solve the problem yet. Therefore, a flexible approach is required which is applicable to any EPC format.

4.5 Proposed Scheme

We propose an illegal interrogation detectable products distribution scheme in RFID-enabled supply chains. The idea of the proposed scheme is to make an attacker access an authentication server and to detect him/her and to limit the number of illegal authentication attempts. The proposed scheme generates random sequences and executes XOR operation with each EPC. EPCs masked with random sequences are written into genuine tags on products whereas random sequences are placed on an authentication server with a pair of a transaction identifier t and an access code c. An access code is divided into shares with a secret sharing scheme and shares are also written into tags on products. We also prepare extra off-the-shelf tags, which we call dummy tags [184], and bogus shares are written into them. Since it can be assumed that an attacker approaches a products-carried vehicle in the public area and interrogates tags from outside of it, he/she may obtain a large number of access code candidates and must try each on an authentication server to obtain random sequences. This construction not only avoids genuine EPCs from being revealed, but also limits the number of illegal authentication attempts and detects an attacker on the server.

In the following, we first define the secret sharing scheme that is necessary for any party. Then, the procedures of both manufacturer and retailer are described with such scheme.

4.5.1 Assumptions

We define a secret sharing scheme algorithm $\Pi = (\mathsf{Share}, \mathsf{Recover}, \mathsf{DummyGen})$ that operates over a random access code space $\mathbb{C}$.

- Share is a probabilistic algorithm that takes an input $c \in \mathbb{C}$, the number of legitimate shares n_L, and the number of dummy shares n_D and outputs n_L shares $S = \{S_1, \cdots, S_{n_L}\}$ with a (τ, n_L)-secret sharing scheme, where $S_i = (x_i, y_i)$, $y_i = f(x_i)$, and any x_i is distinct and is chosen from the set $\{1, \cdots, n_L + n_D\}$, respectively. On invalid input $\hat{c} \in \mathbb{C}$, Share outputs n_L special ("undefined") symbols $\perp$.

- $\mathsf{Recover}$ is a deterministic algorithm that takes input $S' = \{S'_1, \cdots, S'_\tau\}$, τ, and n_L and outputs $c \leftarrow \mathsf{Recover}(S', \tau, n_L) \in \mathbb{C} \cup \perp$, where $\perp$ is a distinguished value that indicates recovery failure.

- $\mathsf{DummyGen}$ is a probabilistic algorithm that takes an input $c \in \mathbb{C}$, n_L legitimate shares $S = \{S_1, \cdots, S_{n_L}\}$ and outputs n_D shares $\tilde{S} = \{\tilde{S}_1, \cdots, \tilde{S}_{n_D}\}$, where $\tilde{S}_i = (\tilde{x}_i, \tilde{y}_i)$, any $\tilde{x}_i$ is chosen from the set $\{1, \cdots, n_L + n_D\} \cap \overline{\{x_1, \cdots, x_{n_L}\}}$ and $\tilde{y}_i \in \mathrm{GF}(q)$ but $\tilde{y}_i \neq f(\tilde{x}_i)$. On invalid input $c \in \mathbb{C}$, $\mathsf{DummyGen}$ outputs a special ("undefined") symbol $\perp$.

We assume that both manufacturer and retailer are honest and follow the defined procedures.

4.5.2 Manufacturer's Procedure

At first, a manufacturer creates n_L products and a unique EPC EPC_i where $i \in [1, n_L]$ is assigned to each product i. A manufacturer also generates a transaction identifier t and a random access code c. A pair of (t, c) is used later to authenti-

cate that a recipient, which includes a distributor or retailer, possesses the correct c on the authentication server. Then, in order to mask genuine EPCs, n_L random sequences $\mathsf{RND} = \{\mathsf{RND}_1, \mathsf{RND}_2, \cdots, \mathsf{RND}_{n_L}\}$ are generated and XORed EPCs $\hat{\mathsf{EPC}} = \{\hat{\mathsf{EPC}}_1, \hat{\mathsf{EPC}}_2, \cdots, \hat{\mathsf{EPC}}_{n_L}\}$ are calculated as follows.

$$\hat{\mathsf{EPC}}_i = \mathsf{EPC}_i \oplus \mathsf{RND}_i, \tag{4.4}$$

where $\oplus$ denotes XOR operation and the length of RND_i is the same as EPC_i. We use XOR operation to conceal genuine EPCs. This realises so-called "one-time pad" that no one can extract genuine EPCs without correct RND if each RND_i is truly random and never reused. A manufacturer places RND with an access code c and transaction identifier t on its own authentication server.

To securely distribute an access code c to a recipient, a manufacturer splits c with a (τ, n_L) secret sharing scheme $\mathsf{Share}(c, \tau, n_L)$ and obtains n_L shares $S = \{S_1, S_2, \cdots, S_{n_L}\}$. For each tag i on a product, EPC_i and S_i are written into its EPC memory bank and USER memory bank, respectively. In addition, t and the URL (or IP address) of the authentication server are also written into another tag. This tag is attached to a container or pallet that products are composed.

Simultaneously, a manufacturer prepares n_D off-the-shelf RFID tags as dummy tags. The objective of introducing dummy tags is to make it infeasible for an attacker to extract the correct access code c even if he/she collects all shares in the public area. The manufacturer obtains n_D bogus shares $\tilde{S} = \{\tilde{S}_1, \tilde{S}_2, \cdots, \tilde{S}_{n_D}\}$ with $\mathsf{DummyGen}(c)$. In order to avoid an attacker from distinguishing legitimate tags from dummy ones, the manufacturer writes $\tilde{S}_i$ exactly the same way as genuine tags. Finally, a manufacturer writes masked EPC $\hat{\mathsf{EPC}}_i$ and a share S_i to tag i with *Lock* command to avoid them from being tampered by an attacker.

A manufacturer composes products into cases or pallets after attaching legitimate tags to products. In order for a recipient to soon distinguish legitimate tags and dummy ones, dummy tags are not attached to any product and delivered.

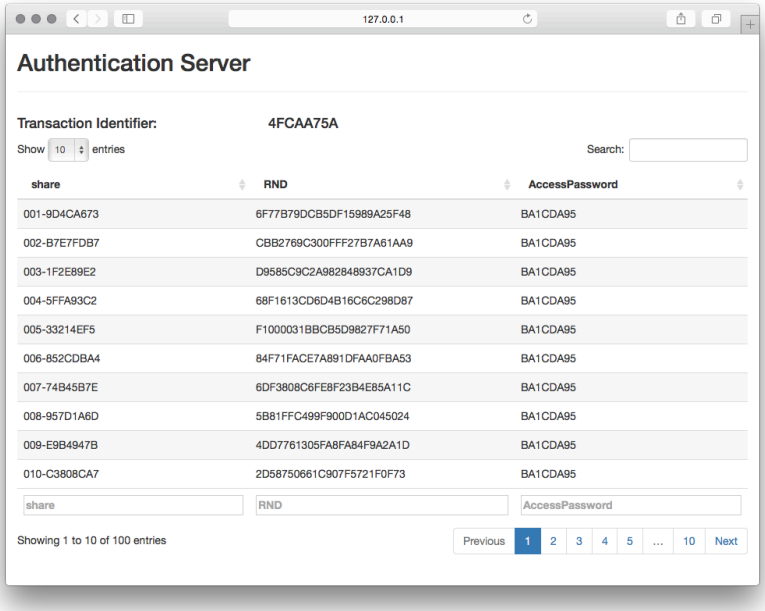


Figure 4-3: Interface of authentication server when authentication is successful.

Setup of Authentication Server

A manufacturer also setups an authentication server. The aim of this server is to detect an attacker who tries to reveal genuine EPCs. A manufacturer registers the series of a pair $(S_i, \mathsf{RND}_i, \mathsf{AP}_i)$, where $i \in [1, n_L]$ and AP_i denotes tag i's access password, with the correspondent c and t in the database of the server. As we describe later, $\mathsf{AP} = \{\mathsf{AP}_1, \cdots, \mathsf{AP}_{n_L}\}$ is necessary for a recipient to update the contents of tags after EPCs are successfully interrogated. An authentication server has a web interface that accepts a pair of transaction identifier t and an access code c. If t and c are given in the forms and such pair exists on a database, the server returns the series of a pair $(S_i, \mathsf{RND}_i, \mathsf{AP}_i)$. The server also records a source identifier e.g., source IP address, and the number of attempts input on the web form. If such attempts from a specific source is abnormal, a manufacturer detects an attack and takes an appropriate measure against it. For instance, many authentication attempts from a specific IP address fail or many authentication attempts for a specific transaction identifier t fail. Since the scheme is identical to the password authentication over the Internet, many countermeasures against illegal authentication attempts can be

executed [186]. In this scheme, when the number of attempts from a specific source IP address is larger than a predefined threshold th, the server rejects any authentication attempts from the source IP address and the existence of an attacker is detected. This way avoids attacks without disabling (legitimate) recipient's authentication attempt.

Figure 4-3 shows a screen image of the proposed authentication system when authentication is successful. A prototype of the proposed authentication server has been implemented with Shiny package in the R language [187]. As can be seen from Figure 4-3, a recipient can identify RND_i and AP_i that correspond with EPC_i from S_i.

4.5.3 Recipient's Procedure

When products arrive, a recipient first unpacks product-cases or pallets. Then, naked tags, i.e., dummy tags, are soon detected and put aside to avoid legitimate tags from being mixed with them, and interrogates only legitimate ones with a reader. After a recipient interrogates tags attached to products and a container (or a pallet) and obtains a share set S, a transaction identifier t, and the URL of an authentication server, an access code c can be recovered with $\mathsf{Recover}(S, \tau, n_L)$. The recipient accesses to manufacturer's authentication server and inputs a recovered access code c and a transaction identifier t. If it is successfully authorized, n_L pairs of $(S_i, \mathsf{RND}_i, \mathsf{AP}_i)$ are returned and genuine EPCs are recovered by Eq. (4.5).

$$\mathsf{EPC}_i = \mathsf{RND}_i \oplus \hat{\mathsf{EPC}}_i. \tag{4.5}$$

Finally, a recipient updates the contents of dummy tags with $\mathsf{DummyGen}(c)$. Dummy tags are reusable and thus the recipient returns them to the source party. Therefore, the process to update the contents of dummy tags is necessary to avoid an attacker from knowing which interrogated tags are dummy. As a manufacturer has sent *Lock* commands to avoid contents of tags from being tampered, a recipient once unlocks them by giving an access password AP_i.

If a recipient further transfers products toward other distributors or retailers, it

Table 4.2. Comparison between dummy tags and normal tags.

Tag Type	Specification	Written Content
Normal tags	EPCglobal C1G2 tag	Genuine masked EPC & share
Dummy tags	EPCglobal C1G2 tag	Random masked EPC & share

should also execute the same procedures as a manufacturer.

4.5.4 Discussion

Comparison Between Dummy Tags and Normal Tags

The difference between dummy and normal tags should be clarified. Table 4.2 shows the comparison between dummy tags and normal tags. Although we name it as "dummy tags", dummy tags are totally same as normal ones from the specification perspective. This is because dummy tags should be indistinguishable from an attacker. Due to the same reason, masked EPCs and shares are also written into dummy tags as with normal ones. However, the only difference is that masked EPCs and share for dummy tags are randomly generated from the EPC and share spaces, respectively.

Advantages and Disadvantages

The main contribution of this research is to enable a manufacturer to notice an attacker who tries to know genuine EPCs and to limit the number of attempts. Although we might be able to use a special device[1] that notices RFID signal to detect illegal interrogation, it also mis-detects the legitimate interrogation when we consider the RFID-enabled ITS (Intelligent Transportation Systems) [188], [189]. In the RFID-enabled ITS, RFID readers are deployed on the road and a tag is attached to a vehicle to enable smarter transportation system. Therefore, if we judge any interrogation during transportation with such a special device, it causes mis-detection.

The downside of our scheme is to manage an authentication server and dummy

[1] www.montiegear.com/uploads/Field_Detector_900MHz_Color.pdf

tags. We measure the latency and increased interrogation time regarding these factors in the Section 4.7.

Attacker's Revealed Information

A manufacturer not only detects an attacker who illegally interrogates product tags but also identifies where and when the tags are interrogated. This is because a transaction t is unique for each shipping between partners. Hence, if an authentication server observes many authentication attempts against a specific t, it can narrow down the area and time that the tags are interrogated by an attacker.

The authentication server can distinguish an attacker who tries to pass authentication without interrogating tags from one that actually approaches tags on products. Since the former attacker does not know any transaction identifier t, he/she may authenticate with many t candidates. In contrast, the latter one wants to know the information for a specific t and thus an authentication server may observe a pair of a specific t and many c.

Relationships between Key and Access Code Distribution

Conventional schemes distribute a symmetric key to encrypt/decrypt EPCs with tags. In contrast, in the proposed scheme, the key is on the authentication server and an access code is distributed to tags instead. In other words, it is identical to securely distribute a key and access code. Therefore, the idea of dummy tags also strengthen the conventional key distribution scheme.

Length of t and c and Required Memory on Tags

We discuss the required memory on tags to implement the scheme. A share S_i and a pair of (t, URL) must be stored on product tag's USER memory bank and container tag, respectively. Therefore the length of t and c must be chosen so that they fit into tag's memory space [2].

[2]In Shamir's secret sharing scheme, the lengths of share and its secret are same.

We first mention the memory constraint on a tag attached to a product. The length of c is especially important to achieve better security. By considering the fact that our scheme is identical to the password authentication on the web server, it should be long enough, e.g., 32 bits, to avoid an attacker from passing authentication. This satisfies memory space constraint of major off-the-shelf tags, e.g., Impinj's Monza 4E series or Alien Technology's Higgs 3 series which have more than 128 bits.

We then check the memory constraint on a tag on a container or a pallet. Obviously, a manufacturer should not set the same t for different transactions simultaneously. Therefore, the length of t should be longer than the number of transactions that a manufacturer deals. The length of URL can be fixed to 32 bits if an IPv4 address is written into a tag. When both lengths of $|t|$ and URL are set as 32 bits, 64 bits are totally required on the USER memory bank. This also satisfies memory space constraint of major off-the-shelf tags.

Unlike the conventional schemes, e.g., [156], [184], the memory space for access passwords is saved since our scheme places them on an authentication server.

Computation Complexity

Our scheme consists of the following four procedures:

1. Splitting an access code into shares;
2. Recovering an access code from shares;
3. Generating RND, and
4. Masking EPCs with RND by XOR operation.

Since Shamir's secret sharing scheme is used, the computation complexities of splitting and recovering an access code are $\mathcal{O}(\tau n_L)$ and $\mathcal{O}(\tau \log^2 \tau)$, respectively [157]. The computation complexity for generating RND and XOR operation are $\mathcal{O}(n_L + n_D)$ and $\mathcal{O}(n_L)$, respectively. Therefore the computationally heaviest part is recovering an access code from shares. latency with the real implementation will be shown in Section 4.7.

4.6 Security Analysis

We prove that our scheme is secure against attackers who want to know genuine EPCs and try to tamper contents of tags. The security analysis is based on the work by Juels *et al.* [156]. $\mathcal{A}_{privacy}$ is defined as a *privacy* attacker that can approach a products-carried vehicle in the public transportation and can access a correspondent authentication server. The goal of $\mathcal{A}_{privacy}$ is to pass authentication on an authentication server and to obtain RND that yields genuine EPCs for a specific t. A *robustness* attacker $\mathcal{A}_{robustness}$ tries to tamper contents of tags. Other possible attacks against our scheme are discussed in Section 4.6.3.

Definition 1. *We call that the proposed scheme is* $(\tau, n_L, n_D, \epsilon_p, \epsilon_r)$*-secure against a probabilistic polynomial time attacker who is given unbounded shares.*

4.6.1 Privacy Attacker

$\mathcal{A}_{privacy}$ tries to pass authentication on an authentication server by giving c and t. An attacker $\mathcal{A}_{privacy}$ can use the following oracles:

$\mathcal{O}_{\mathsf{Collect}}()$: This oracle returns share sets S mixed from both legitimate and dummy tags,

$\mathcal{O}_{\mathsf{Recover}}(S)$: This oracle returns $\tilde{c}$ by inputting shares S. If $|S| < \tau$, it outputs $\perp$,

$\mathcal{O}_{\mathsf{Choose}}(S, \tau)$: This oracle returns by randomly choosing $S' = \{S'_1, \cdots, S'_\tau\}$ from S. If $|S| < \tau$, it returns $\perp$, and

$\mathcal{O}_{\mathsf{Auth}}(\tilde{c}, c, t)$: This oracle returns 1 if $c = \tilde{c}$ for an existent pair (c, t). Otherwise, it returns $\perp$.

By using the above oracles, a privacy attacker tries the privacy challenge defined as follows:

Challenge $\mathbf{Cha}_{privacy}[\Pi, \mathbb{C}]$:

Input: τ, c, t, EPC

Procedure:

$\hat{S} \leftarrow \mathcal{O}_{\mathsf{Collect}}()$

$\hat{S}' \leftarrow \mathcal{O}_{\mathsf{Choose}}(\hat{S}, \tau)$

$\tilde{c} \leftarrow \mathcal{O}_{\mathsf{Recover}}(\hat{S}')$

Output: $\mathcal{O}_{\mathsf{Auth}}(\tilde{c}, c, t)$

Claim 1. *Against the above settings, the probability that an attacker $\mathcal{A}_{privacy}$ who possesses n_{IP} IP addresses identifies genuine EPCs is bounded by ϵ_p.*

$$\mathsf{Pr}[\mathbf{Cha}_{privacy}[\Pi, \mathbb{C}] \Rightarrow 1] = \epsilon_p = 1 - \left(1 - \frac{\binom{n_L}{\tau}}{\binom{n_L+n_D}{\tau}}\right)^{th \cdot n_{IP}}.$$

Proof. An attacker can collect all shares interrogated from both legitimate tags and dummy ones. Since an attacker cannot distinguish them from outside of a carrying vehicle, he/she must try each combination of shares. Therefore, the probability that a privacy attacker finds the correct access code c is equivalent to the probability of choosing τ legitimate shares out of totally $(n_L + n_D)$ shares. This probability is represented as $\binom{n_L}{\tau}/\binom{n_L+n_D}{\tau}$. However, an attacker cannot try every candidate and can only attempt th times for one IP address. Since an attacker is assumed to have n_{IP} IP addresses, the probability that $\mathcal{A}_{privacy}$ passes authentication and obtains correct RND is represented as the equation in the Claim 1. □

4.6.2 Robustness Attacker

A robustness attacker tries to tamper the contents of tags so that a recipient cannot recover correct shares or EPCs.

Claim 2. *Given our construction, the advantage of an attacker who tries to tamper the contents of tags is bounded by ϵ_r.*

$$\mathsf{Pr}[\mathbf{Cha}_{robustness}[\Pi, \mathbb{C}] \Rightarrow 1] = \epsilon_r < \max\left(\epsilon_p, 2^{-|c|}\right),$$

where $\max(a, b)$ *returns the bigger value between* a *and* b.

Proof. In our scheme, an attacker cannot tamper shares unless he/she knows the correct access password. Therefore, the attacker's success probability is bounded by the probability of identifying the correct access password AP. There are two strategies for the robustness attacker to obtain AP. The first one is to pass authentication on the server and this probability is bounded by ϵ_p. The other one is to find the AP with random guessing. This probability is $2^{-|c|}$. Therefore, the success probability of robustness attacker is bounded by the bigger one of them. Thus, the Claim 2 has been proven. □

4.6.3 Other Possible Attacks against Our Scheme

A passive attacker may want to reveal genuine EPCs without approaching a products-carried vehicle but with accessing an authentication server. This attack will fail due to the following two reasons. The first one is that an attacker cannot obtain a legitimate transaction ID t because t is only obtained by interrogating tags in a truck. As the second reason, even if an attacker guessed the correct pair (t, c), every information, i.e., RND, share, access passwords, obtained from an authentication server would be useless for such an attacker because he/she does not know masked EPCs $\hat{\mathsf{EPC}}$.

Products can be avoided from being tracked throughout the end-to-end path, i.e., path from a manufacturer to a retailer. This is because if a recipient further ships products to other parties, e.g., retailers, it also prepares its own authentication server and regenerates RND for each product and thus the revealed (masked) EPCs vary every time they are shipped.

An attacker may be able to launch the DoS (Denial of Service) attack against an authentication server. That is, an attacker tries to halt the service and makes a legitimate recipient infeasible to obtain genuine RND. Since our proposed authentication server can be seen as a generic password authentication server, many conventional countermeasures are applicable for the DoS attack, e.g., [190], [191].

TABLE 4.3. REQUIRED n_D VERSUS n_L AND r_τ.

(a) $|c| = 16$ bits

n_L	Required n_D	
	$r_\tau = 0.2$	$r_\tau = 0.9$
10	2,420	13
50	93	6
100	67	6
500	53	5
1,000	52	5

(b) $|c| = 32$ bits

n_L	Required n_D	
	$r_\tau = 0.2$	$r_\tau = 0.9$
10	621,720	58
50	372	15
100	184	12
500	112	11
1,000	106	10

4.7 Performance Evaluation

In order to show the efficiency of our scheme, we evaluate our scheme with both theoretical calculation and measurements with an off-the-shelf devices. We first show the required number of dummy tags and then the detection accuracy that an authentication server can successfully detect an attacker. We then measure the computation time to split and extract an access code c and the interrogation time with off-the-shelf RFID devices.

4.7.1 Required Number of Dummy Tags n_D and τ

The number of dummy tags is a key factor in order to securely distribute an access code to a recipient. The required number of dummy tags is theoretically obtained when the probability that an attacker chooses τ correct shares from $(n_L + n_D)$ ones i.e., $\binom{n_L}{\tau}/\binom{n_L+n_D}{\tau}$ is less than the probability that an attacker randomly guesses c i.e., $2^{-|c|}$. For the ease of discussion, let r_τ denote the ratio of τ to n_L, i.e., $r_\tau = \tau/n_L$. Table 4.3 shows the required n_D versus n_L and r_τ when (a) $|c| = 16$ bits and (b) $|c| = 32$ bits, respectively. In [126], [156], the authors mention that totally about thousands of products e.g., pharmaceuticals or DVDs, are initially assembled then shipped toward distributors. Distributors also disperse them toward retailers. Finally, about ten products are on the consumer side. Therefore, the order of magnitude of products ranges between $3 \rightarrow 2 \rightarrow 1$ and we vary n_L from 10^1 to 10^3. In Table 4.3, as the number of legitimate tags increases, the less number of dummy tags is required. Intu-

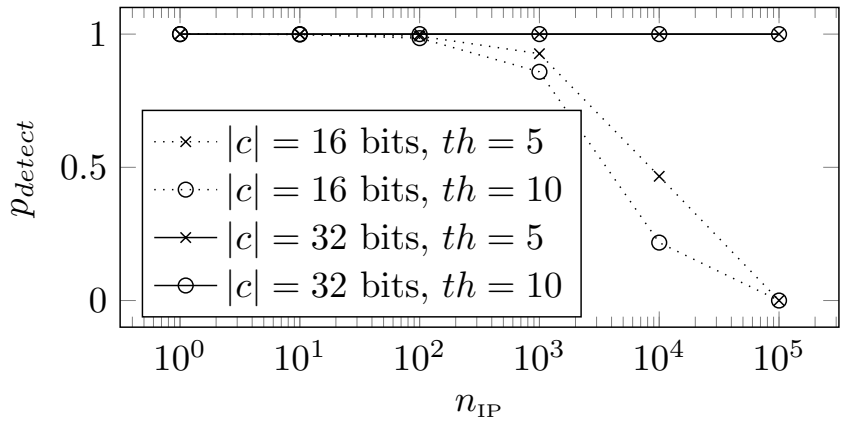


Figure 4-4: The probability that an authentication server can detect an attacker versus n_{IP}.

itively, this is because combinations that an attacker should try get sharply increased when n_L gets large. In addition, as $|c|$ gets larger, the required n_D gets also increased.

Then we discuss how to set r_τ. As r_τ gets larger, n_D gets decreased. In contrast, as r_τ get smaller, n_D increases but the computation time for splitting and extracting c is also decreased because the degree of Shamir's secret sharing, i.e., τ is also decreased. As it will be shown in Section 4.7.3, when n_L is large, e.g., $n_L \geq 500$, the computation time for c is sharply increased. Therefore we argue that when n_L is small, r_τ should be set large, whereas when n_L is large, r_τ should be set small.

4.7.2 Detection Probability

We show the detection probability of our scheme. Let p_{detect} denote the probability that an authentication server can detect an attacker. p_{detect} is the probability that an attacker who possesses n_{IP} IP addresses cannot pass any authentication, and it can be represented as Eq. (4.6).

$$p_{detect} = \left(1 - \frac{\binom{n_L}{\tau}}{\binom{n_L+n_D}{\tau}}\right)^{th \cdot n_{\mathrm{IP}}} \approx \left(1 - 2^{-|c|}\right)^{th \cdot n_{\mathrm{IP}}}. \tag{4.6}$$

Figure 4-4 shows p_{detect} versus n_{IP}. Although it cannot be presumed how many number of IP addresses an attacker can possess, we set n_{IP} from 1 to 100,000. From

Table 4.4. Computation time for an access code ($|c| = 32$ bits).

n_L	r_τ	splitting c [ms]	extracting c [ms]
10	0.9	5.1	6.1
50	0.9	7.5	11
100	0.9	18	8.9×10^2
500	0.9	3.4×10^2	1.2×10^5
500	0.2	80	1.2×10^2
1,000	0.9	-	-
1,000	0.2	3.2×10^2	1.1×10^4
1,000	0.1	1.7×10^2	1.3×10^3

Figure 4-4, on the one hand, when the length of an access code c is 32 bits, an authentication server can detect an attacker without fail even if he/she prepares 100,000 IP addresses. On the other hand, when $|c| = 16$ bits, p_{detect} gets decreased with n_{IP}. From this result, $|c|$ should be set more than 32 bits for the security.

4.7.3 Computation Time

We measure the computation time to split and extract c whose size $|c| = 32$ bits with a laptop machine, which is a MacBookPro Late 2013 equipping a dual-core Intel Core i7 2.8 GHz and a 16 GB RAM memory. The following trial is executed as many as 100 times and the computation times are averaged: we randomly generate c, split into n_L shares, and extract c with ssss which is an implementation of Shamir's secret sharing scheme [192]. Table 4.4 shows the computation time to split and extract c. In Table 4.4, when $n_L = 1,000$ and $r_\tau = 0.9$, our machine cannot execute it due to memory failure and the results are represented as '-'. From Table 4.4, the computation time to extract c takes much longer time than that of splitting it. It can also seen that the computation time exponentially increases with n_L. For example, when $n_L = 500$ and $r_\tau = 0.9$, the computation time for extracting c is about 2 minutes and it is not acceptable in the real implementation. When we deal large n_L, say 500, we should set low r_τ. For example, when $n_L = 500$ and $r_\tau = 0.2$, the computation time for extracting c is decreased to 120 ms and it can be acceptable in the real situation. Even when $n_L = 1,000$, the time to extract c can be shorten to 1.3 sec by setting r_τ

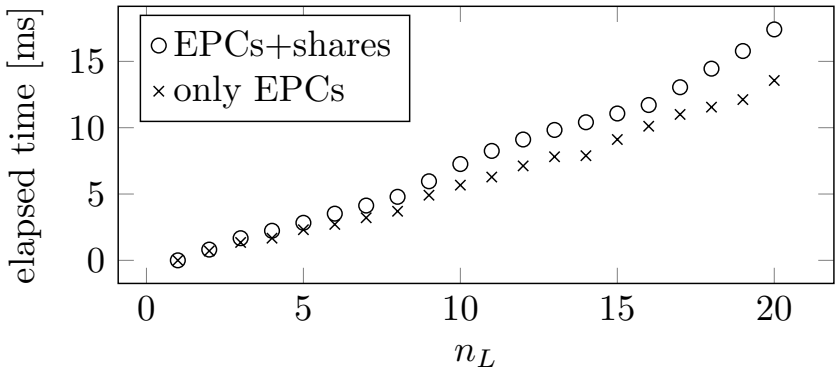


Figure 4-5: Interrogation time versus the number of tags.

to 0.1. Although, the required number of dummy tags n_D is increased, only tens of dummy tags are required when low r_τ and large n_L are used.

We also measure the interrogation time to read masked EPCs and shares with RFID devices. We use an Impinj SpeedwayRevolution R420 as an RFID reader, Times-7 A7030C circular polarised UHF shelf antenna, and Alien Technology ALN-9640 as tags. The RFID reader is connected with the above laptop computer via Ethernet. A 32 bit access code c is generated and split into shares with the (15, 20)-Shamir's secret sharing scheme with ssss. Then, 96 bit EPCs and the shares are written into 20 tags by using Impinj Octane SDK [193]. The tags are deployed two meters away from the reader's antenna in LOS (Line of Sight) environment. Figure 4-5 shows the comparison of interrogation time (i) when tags are interrogated together with shares and (ii) only EPCs. In this measurement, the interrogation time against 20 tags averaged over ten trials. From this figure, the interrogation time almost linearly increases with the number of tags. The interrogation time is slightly increased by reading shares as well. However, the interrogation time is millisecond order and thus it can be concluded that the latency is trivial even if shares are additionally read.

4.8 Conclusions

We have proposed an illegal interrogation detectable EPC distribution scheme in RFID-enabled supply chains. The idea of the proposed scheme is to mask EPCs with

random sequences and place them on an authentication server. The access code for authentication is split into genuine tags by using a secret sharing scheme and we also involve dummy tags that a genuine recipient can distinguish. An attacker who wants to reveal genuine EPCs must pass authentication and thus the proposed scheme can detect illegal attempts by attacker and even limits such attempts. We show that the proposed scheme is provably secure and easily implementable with off-the-shelf RFID devices and a generic computer. From the performance evaluation, a 32-bit access code is enough to detect an attacker and suffices the memory requirement on off-the-shelf tags. It has been also shown that the computation time for splitting and recovering an access code can be controlled within a second by varying the ratio of required shares to number of products.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This dissertation has discussed a study on security and privacy for ad-hoc network, VoIP service and RFID-enabled supply chains system. More and more smart wireless devices are emerging, e.g., wireless sensor devices, smartphones, and RFID. They yield not only new beneficial systems and services for our lives but also unprecedented threats that cannot be solved by the traditional defence approaches. Therefore, it is an urgent demand to solve each unprecedented issue. In this dissertation, we have solved three issues regarding to wireless sensor device, smartphone, and RFID-based system and services. The contribution of this paper is summarized as follows.

In Chapter 2, we have proposed a provably secure lightweight verification scheme in FFS protocol. The basic idea to reduce computation cost in the verification is to make most of elements in a challenge $\mathbf{e}$ to 0 when generating a challenge. To avoid lowering security, a challenge is divided into multiple ones and restrict the number of elements set as 1 in each challenge. Since a prover must pass every challenge, the proposed scheme can achieve the sufficient security by setting appropriate upper bounds for each challenge. The proposed scheme is proved as ZKP by referring the security analysis in [78]. By the theoretical computation, it has been shown that the number of division $d = 2$ and upper bounds for each challenge $\mathbf{u} = (2, 4)$ give the lowest computation on the verification for the security parameter $kt = 20$. The

number of multiplication is reduced by 48-54% and 61-78% when no malicious provers exist and when only malicious provers exist, respectively. We have shown that the computation time is also shorten on an Android device.

In Chapter 3, we have proposed an unsupervised SPIT callers detection with a clustering algorithm. The proposed scheme turns complex threshold setting and training problems into clustering the callers and identifying the cluster. In contrast to conventional schemes, we use the features to find the dissimilarity among callers and this avoids threshold tuning and the training phase. By the computer simulation, it has been shown that the proposed scheme using RF dissimilarity and PAM clustering outperforms the conventional schemes by means of classification performance when SPITters account for more than 20% of inspected callers accuracy against our dataset. We have also shown that the proposed scheme can tolerate as many as 100,000 callers using an off-the-shelf computer.

In Chapter 4, we have proposed an illegal interrogation detectable EPC distribution scheme in RFID-enabled supply chains. The idea of the scheme is to mask EPCs with random sequences and place them on an authentication server. The access code for authentication is split into genuine tags by using a secret sharing scheme and we also involve dummy tags that a genuine recipient can distinguish. An attacker who wants to reveal genuine EPCs must pass authentication and thus a manufacturer can detect illegal attempts by attacker and even limits such attempts on an authentication server. It has been shown that the proposed scheme is provably secure and easily implementable with off-the-shelf RFID devices and a generic computer. From the performance evaluation, a 32-bit access code is enough to detect an attacker and suffices the memory requirement on off-the-shelf tags. We have also shown that the computation time for splitting and recovering an access code can be controlled within a second by varying the ratio of required shares to number of products.

5.2 Future Work

The work carried out in this thesis could be completed by several extensions or be a starting point for other interesting research initiatives. The paper did not address all issues and we describe a few of them below.

In Chapter 2, we have proposed a modified FFS protocol to reduce the computation cost on the verifier. However, in an ad-hoc network, a verifier may authenticate more requests from a large number of devices. In this case, one-to-many authentication scheme is necessary. One possible idea for solution is to leverage SMC (Secure Multi-party Computation) where n users have a unique secret x_i and calculate a common function $f(x_1, \cdots, x_n)$ without revealing x_i [194]. For instance, let x_i denote a private key for a device i and a verifier checks the validity of a verification function $f(x_1, \cdots, x_n)$ to authenticate devices at once.

We have proposed an unsupervised SPITters detection scheme in Chapter 3. However there are at least a few issues. The first one is that our scheme is only valid when SPITters account for more than 20% of entire caller and a more sophisticated scheme is necessary to deal with the case when SPITters are few. One solution of this issue is to add artificial SPITters' feature vectors to be balanced. The second issue is that a detection party, i.e., a service provider, can only use its own callers' information. If it could use information from other competitors' information, the classification performance would improve. This might be solvable with SMC. If x_i and $f(x_1, \cdots, x_n)$ denote feature vectors of a service provider i and detection algorithm, respectively. The last one is regarding to computational complexity.

Although we have solved an issue of secure distribution for RFID-enabled supply chains in Chapter 4, still some issues exist. For example, we assume that any parties, including distributors and retailers, are honest and follow the pre-defined procedures. However, it can be considered a scenario where operators of RFID readers break this assumption and disclose obtained genuine EPCs. Therefore, it is necessary to propose an approach that is secure against such scenario. The other one is inefficiency. Since each tag must be read and written a share and masked EPC, it will cause delay in the

procedures of supply chains. To solve this problem, some mitigation methods must be considered.

Bibliography

[1] Z. Shelby and C. Bormann, *6LoWPAN: The wireless embedded Internet*. 2011, vol. 43.

[2] D.-M. Han and J.-H. Lim, "Smart home energy management system using ieee 802.15.4 and zigbee," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1403–1410, 2010.

[3] A. Brandt and J. Buron, "Home automation routing requirements in low-power and lossy networks," *RFC5826*, 2010.

[4] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Wireless sensor networks for structural health monitoring," in *ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2006, pp. 427–428.

[5] J. P. Lynch and K. J. Loh, "A summary review of wireless sensors and sensor networks for structural health monitoring," *Shock and Vibration Digest*, vol. 38, no. 2, pp. 91–130, 2006.

[6] A. Basharat, N. Catbas, and M. Shah, "A framework for intelligent sensor network with video camera for structural health monitoring of bridges," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2005, pp. 385–389.

[7] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *International Symposium on Information Processing in Sensor Networks (IPSN)*, 2007, pp. 254–263.

[8] X. Hu, B. Wang, and H. Ji, "A wireless sensor network-based structural health monitoring system for highway bridges," *Computer-Aided Civil and Infrastructure Engineering*, vol. 28, no. 3, pp. 193–209, 2013.

[9] F. M. Al-Turjman, H. S. Hassanein, and M. Ibnkahla, "Connectivity optimization for wireless sensor networks applied to forest monitoring," in *IEEE International Conference on Communications (ICC)*, 2009, pp. 1–6.

[10] Z. G. Kovács, G. E. Marosy, and G. Horváth, "Case study of a simple, low power WSN implementation for forest monitoring," in *Biennial Baltic Electronics Conference (BEC)*, 2010, pp. 161–164.

[11] B. Li, H. Wang, B. Yan, and C. Zhang, "The research of applying wireless sensor networks to intelligent transportation system (ITS) based on IEEE 802.15. 4," in *International Conference on ITS Telecommunications*, 2006, pp. 939–942.

[12] M. Tubaishat, P. Zhuang, Q. Qi, and Y. Shang, "Wireless sensor networks in intelligent transportation systems," *Wireless communications and mobile computing*, vol. 9, no. 3, pp. 287–302, 2009.

[13] D. Tacconi, D. Miorandi, I. Carreras, F. Chiti, and R. Fantacci, "Using wireless sensor networks to support intelligent transportation systems," *Ad Hoc Networks*, vol. 8, no. 5, pp. 462–473, 2010.

[14] V. W. Tang, Y. Zheng, and J. Cao, "An intelligent car park management system based on wireless sensor networks," in *International Symposium on Pervasive Computing and Applications*, IEEE, 2006, pp. 65–70.

[15] R. Lu, X. Lin, H. Zhu, and X. S. Shen, "Spark: A new vanet-based smart parking scheme for large parking lots," in *IEEE INFOCOM*, 2009, pp. 1413–1421.

[16] Z. Fan, P. Kulkarni, S. Gormus, C. Efthymiou, G. Kalogridis, M. Sooriyabandara, Z. Zhu, S. Lambotharan, and W. H. Chin, "Smart grid communications: Overview of research challenges, solutions, and standardization activi-

ties," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 21–38, 2013.

[17] T. O'Donovan, J. O'Donoghue, C. Sreenan, D. Sammon, P. O'Reilly, and K. O'Connor, "A context aware wireless body area network (ban)," in *IEEE International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, Apr. 2009, pp. 1–8.

[18] H. Yan, H. Huo, Y. Xu, and M. Gidlund, "Wireless sensor network based e-health systems: Implementation and experimental results," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2288–2295, 2010.

[19] B. Goode, "Voice over internet protocol (VoIP)," *Proceedings of the IEEE*, vol. 90, no. 9, pp. 1495–1517, 2002.

[20] H. Sinnreich and A. B. Johnston, *Internet communications using SIP: Delivering VoIP and multimedia services with Session Initiation Protocol*. John Wiley & Sons, 2012, vol. 27.

[21] J. Constine, *Facebook messenger launches free voip video calls over cellular and wi-fi | techcrunch*, `http://techcrunch.com/2015/04/27/facebook-messenger-video-chat/`, (Visited on 12/29/2015), Apr. 2015.

[22] G. Developers, *Google places api web service*, `https://developers.google.com/places/web-service/`, (Visited on 12/29/2015), Dec. 2015.

[23] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *ACM International Conference on Mobile Systems, Applications, and services (MobiSys)*, 2009, pp. 55–68.

[24] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140–150, 2010.

[25] R. Want, "An introduction to RFID technology," *Pervasive Computing, IEEE*, vol. 5, no. 1, pp. 25–33, 2006.

[26] R. Angeles, "RFID technologies: Supply-chain applications and implementation issues," *Information systems management*, vol. 22, no. 1, pp. 51–65, 2005.

[27] M. Harrison, "EPC information service (EPCIS)," in *Auto-ID Labs Research Workshop*, 2004, pp. 29–30.

[28] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

[29] J. Deng, R. Han, and S. Mishra, "Insens: Intrusion-tolerant routing for wireless sensor networks," *Computer Communications*, vol. 29, no. 2, pp. 216–230, 2006.

[30] Y. Challal, A. Ouadjaout, N. Lasla, M. Bagaa, and A. Hadjidj, "Secure and efficient disjoint multipath construction for fault tolerant routing in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1380–1397, 2011.

[31] M. A. Moustafa, M. Youssef, and M. N. El-Derini, "Msr: A multipath secure reliable routing protocol for WSNs," in *IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, 2011, pp. 54–59.

[32] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen, "Secure and energy-efficient disjoint multipath routing for WSNs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 7, pp. 3255–3265, 2012.

[33] A. Kimura, E. Kohno, and Y. Kakuda, "Security and dependability enhancement of wireless sensor networks with multipath routing utilizing the connectedness of joint nodes," in *IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2012, pp. 342–348.

[34] H. Al-Hamadi and R. Chen, "Redundancy management of multipath routing for intrusion tolerance in heterogeneous wireless sensor networks," *IEEE*

Transactions on Network and Service Management, vol. 10, no. 2, pp. 189–203, 2013.

[35] K. Sha, J. Gehlot, and R. Greve, "Multipath routing techniques in wireless sensor networks: A survey," *Wireless Personal Communications*, vol. 70, no. 2, pp. 807–829, 2013.

[36] A. D. Wood, L. Fang, J. A. Stankovic, and T. He, "Sigf: A family of configurable, secure routing protocols for wireless sensor networks," in *ACM Workshop on Security of Ad hoc and Sensor Networks*, 2006, pp. 35–48.

[37] G. Zhan, W. Shi, and J. Deng, "Design and implementation of TARF: A trust-aware routing framework for WSNs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 2, pp. 184–197, 2012.

[38] F. Bao, I.-R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169–183, 2012.

[39] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 867–880, 2012.

[40] Y. Sun, H. Luo, and S. K. Das, "A trust-based framework for fault-tolerant data aggregation in wireless multimedia sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 785–797, 2012.

[41] X. Li, F. Zhou, and J. Du, "Ldts: A lightweight and dependable trust system for clustered wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 924–935, 2013.

[42] O. Khalid, S. U. Khan, S. A. Madani, K. Hayat, M. I. Khan, N. Min-Allah, J. Kolodziej, L. Wang, S. Zeadally, and D. Chen, "Comparative study of trust and reputation systems for wireless sensor networks," *Security and Communication Networks*, vol. 6, no. 6, pp. 669–688, 2013.

[43] J. Duan, D. Yang, H. Zhu, S. Zhang, and J. Zhao, "Tsrf: A trust-aware secure routing framework in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, 2014.

[44] N. Labraoui, "A reliable trust management scheme in wireless sensor networks," in *IEEE International Symposium on Programming and Systems (ISPS)*, 2015, pp. 1–6.

[45] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: A link layer security architecture for wireless sensor networks," in *ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2004, pp. 162–175.

[46] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2005, pp. 324–328.

[47] A. Liu and P. Ning, "Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks," in *IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2008, pp. 245–256.

[48] W. K. Koo, H. Lee, Y. H. Kim, and D. H. Lee, "Implementation and analysis of new lightweight cryptographic algorithm suitable for wireless sensor networks," in *IEEE International Conference on Information Security and Assurance (ISA)*, 2008, pp. 73–76.

[49] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, "Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks," in *Wireless sensor networks*, Springer, 2008, pp. 305–320.

[50] P. Szczechowiak, A. Kargl, M. Scott, and M. Collier, "On the application of pairing based cryptography to wireless sensor networks," in *ACM Conference on Wireless Network Security (WiSec)*, 2009, pp. 1–12.

[51] L. B. Oliveira, D. F. Aranha, C. P. Gouvêa, M. Scott, D. F. Câmara, J. López, and R. Dahab, "Tinypbc: Pairings for authenticated identity-based non-interactive key distribution in sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 485–493, 2011.

[52] C. Lederer, R. Mader, M. Koschuch, J. GroSSschädl, A. Szekely, and S. Tillich, "Energy-efficient implementation of ecdh key exchange for wireless sensor networks," in *Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks*, Springer, 2009, pp. 112–127.

[53] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security (TISSEC)*, vol. 8, no. 2, pp. 228–258, 2005.

[54] F. Delgosha and F. Fekri, "Key pre-distribution in wireless sensor networks using multivariate polynomials," in *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2005, pp. 118–129.

[55] R. Di Pietro, L. V. Mancini, and A. Mei, "Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks," *Wirel. Netw.*, vol. 12, no. 6, pp. 709–721, Nov. 2006.

[56] C. Castelluccia and A. Spognardi, "Rok: A robust key pre-distribution protocol for multi-phase wireless sensor networks," in *IEEE International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm)*, 2007, pp. 351–360.

[57] S. Ruj and B. Roy, "Key predistribution using combinatorial designs for grid-group deployment scheme in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 1, pp. 1–28, 2009.

[58] B. Kong, H. Chen, X. Tang, and K. Sezaki, "Key pre-distribution schemes for large-scale wireless sensor networks using hexagon partition," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2010, pp. 1–5.

[59] W. Bechkit, Y. Challal, A. Bouabdallah, and V. Tarokh, "A highly scalable key pre-distribution scheme for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 948–959, 2013.

[60] S. J. Choi, K. T. Kim, and H. Y. Youn, "An energy-efficient key predistribution scheme for secure wireless sensor networks using eigenvector," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[61] S. Ruj, A. Nayak, and I. Stojmenovic, "Pairwise and triple key distribution in wireless sensor networks with applications," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2224–2237, 2013.

[62] R. D. Pietro, L. V. Mancini, Y. W. Law, S. Etall, and P. Havinga, "Lkhw: A directed diffusion-based secure multicast scheme for wireless sensor networks," in *IEEE International Conference on Parallel Processing Workshops*, 2003, pp. 397–406.

[63] P. Traynor, H. Choi, G. Cao, S. Zhu, and T. La Porta, "Establishing pair-wise keys in heterogeneous sensor networks.," in *IEEE INFOCOM*, 2006, pp. 1–12.

[64] R. Dutta, E.-C. Chang, and S. Mukhopadhyay, "Efficient self-healing key distribution with revocation for wireless sensor networks using one way key chains," in *Applied Cryptography and Network Security*, Springer, 2007, pp. 385–400.

[65] S. Hussain, F. Kausar, and A. Masood, "An efficient key distribution scheme for heterogeneous sensor networks," in *ACM International Conference on Wireless Communications and Mobile Computing (IWCMC)*, 2007, pp. 388–392.

[66] Y. Jiang, C. Lin, M. Shi, and X. S. Shen, "Self-healing group key distribution with time-limited node revocation for wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 14–23, 2007.

[67] Y. Wang and B. Ramamurthy, "Group rekeying schemes for secure group communication in wireless sensor networks," in *IEEE International Conference on Communications (ICC)*, 2007, pp. 3419–3424.

[68] M. S. Bouassida, I. Chrisment, and O. Festor, "Group key management in manets.," *International Journal of Network Security*, vol. 6, no. 1, pp. 67–79, 2008.

[69] L. Lu, J. Han, Y. Liu, L. Hu, J. Huai, L. M. Ni, and J. Ma, "Pseudo trust: Zero-knowledge authentication in anonymous P2Ps," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 10, pp. 1325–1337, 2008.

[70] F. Le and S. M. Faccin, *Ipv6 address ownership solution based on zero-knowledge identification protocols or based on one time password*, Jun. 2009.

[71] J. M. Kizza, "Feige-Fiat-Shamir ZKP scheme revisited," *International Journal of Computing and ICT Research*, vol. 4, no. 1, pp. 9–19, 2010.

[72] S. Grzonkowski and P. M. Corcoran, "Sharing cloud services: User authentication for social enhancement of home networking," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1424–1432, 2011.

[73] M. Hashim, S. G., and S. A., "Authentication in wireless sensor networks using zero knowledge protocol," in *Computer Networks and Intelligent Computing*, ser. Communications in Computer and Information Science, vol. 157, 2011, pp. 416–421.

[74] S. K. Udgata, A. Mubeen, and S. L. Sabat, "Wireless sensor network security model using zero knowledge protocol," 2011, pp. 1–5.

[75] M. Sandhya and T. R. Rangaswamy, "Zero knowledge and hash-based secure access control scheme for mobile RFID systems," *Arabian Journal for Science and Engineering*, vol. 39, no. 3, pp. 1897–1906, 2014.

[76] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 2, pp. 52–73, 2009.

[77] H. Chan and A. Perrig, "Pike: Peer intermediaries for key establishment in sensor networks," in *IEEE INFOCOM*, vol. 1, 2005, pp. 524–535.

[78] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *Journal of Cryptology*, vol. 1, no. 2, pp. 77–94, 1988.

[79] D. Shin, J. Ahn, and C. Shim, "Progressive multi gray-leveling: A voice spam protection algorithm," *IEEE Network*, vol. 20, no. 5, pp. 18–24, Sep. 2006.

[80] "Enhancing the blockage of spam over internet telephony (SPIT) using adaptive PMG algorithm," R. Lee, Ed., ser. Studies in Computational Intelligence, vol. 149, Springer Berlin Heidelberg, 2008, pp. 15–26.

[81] Y.-S. Wu, S. Bagchi, N. Singh, and R. Wita, "Spam detection in voice-over-ip calls through semi-supervised clustering," in *IEEE/IFIP International Conference on Dependable Systems Networks (DSN)*, 2009, pp. 307–316.

[82] Y. Bai, X. Su, and B. Bhargava, "Adaptive voice spam control with user behavior analysis," in *IEEE International Conference on High Performance Computing and Communications (HPCC)*, 2009, pp. 354–361.

[83] ——, "Detection and filtering spam over internet telephony: A user-behavior-aware intermediate-network-based approach," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2009, pp. 726–729.

[84] H. Sengar, X. Wang, and A. Nichols, "Thwarting spam over internet telephony (SPIT) attacks on VoIP networks," in *IEEE International Workshop on Quality of Service (IWQoS)*, 2011, pp. 1–3.

[85] T. Jung, S. Martin, D. Ernst, and G. Leduc, "SPRT for SPIT: Using the sequential probability ratio test for spam in VoIP prevention," in *Dependable Networks and Services*, Springer Berlin Heidelberg, 2012, pp. 74–85.

[86] H. Sengar, X. Wang, and A. Nichols, "Call behavioral analysis to thwart SPIT attacks on VoIP networks," in *Security and Privacy in Communication Networks*, ser. LNICS, Social Informatics and Telecommunications Engineering, vol. 96, Springer Berlin Heidelberg, 2012, pp. 501–510.

[87] ——, "Call behavioral analysis to thwart SPIT attacks on VoIP networks," in *Security and Privacy in Communication Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 96, 2012, pp. 501–510.

[88] F. Wang, M. Feng, and K. Yan, "Voice spam detecting technique based on user behavior pattern model," in *IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2012, pp. 1–5.

[89] S. Chiappetta, C. Mazzariello, R. Presta, and S. P. Romano, "An anomaly-based approach to the analysis of the social behavior of VoIP users," *Elsevier Computer Networks*, vol. 57, no. 6, pp. 1545–1559, 2013.

[90] M. Amanian, M. H. Yaghmaee Moghaddam, and H. Khosravi Roshkhari, "New method for evaluating anti-SPIT in VoIP networks," in *International eConference on Computer and Knowledge Engineering (ICCKE)*, 2013, pp. 374–379.

[91] V. A. Balasubramaniyan, A. Mustaque, and P. Haesun, "CallRank: combating SPIT using call duration, social networks and global reputation," in *Conference on Email and Anti-Spam (CEAS)*, 2007.

[92] H. Juho and A. Gurtov, "Filtering SPAM in P2PSIP communities with web of trust," in *International ICST Conference, MobiSec*, vol. 17, 2009, pp. 110–121.

[93] T. Kusumoto, E. Y. Chen, and M. Itoh, "Using call patterns to detect unwanted communication callers," in *IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)*, 2009, pp. 64–70.

[94] J. Seedorf, N. D'Heureuse, S. Niccolini, and M. Cornolti, "Detecting trustworthy real-time communications using a web-of-trust," in *IEEE GLOBECOM*, 2009, pp. 1–8.

[95] R. Zhang and A. Gurtov, "Collaborative reputation-based voice spam filtering," in *International Workshop on Database and Expert Systems Application (DEXA)*, 2009, pp. 33–37.

[96] N. Chaisamran, T. Okuda, G. Blanc, and S. Yamaguchi, "Trust-based VoIP spam detection based on call duration and human relationships," in *IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)*, 2011, pp. 451–456.

[97] M. A. Azad and R. Morla, "Mitigating SPIT with social strength," in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2012, pp. 1393–1398.

[98] M. A. Azad and R. Morla, "Caller-rep: Detecting unwanted calls with caller social strength," *Computers & Security*, vol. 39, Part B, pp. 219–236, 2013.

[99] N. Chaisamran, T. Okuda, and S. Yamaguchi, "Using a trust model to reduce false positives of SIP flooding attack detection in IMS," in *IEEE Computer Software and Applications Conference Workshops (COMPSACW)*, 2013, pp. 254–259.

[100] S. Chen, G. Wang, and W. Jia, "A trust model using implicit call behavioral graph for mobile cloud computing," *Cyberspace Safety and Security*, LNCS, vol. 8300, pp. 387–402, 2013.

[101] J. Quittek, S. Niccolini, S. Tartarelli, M. Stiemerling, M. Brunner, and T. Ewald, "Detecting SPIT calls by checking human communication patterns," 2007, pp. 1979–1984.

[102] H. Hai, Y. Hong-tao, and F. Xiao-Lei, "A SPIT detection method using voice activity analysis," in *International Conference on Multimedia Information Networking and Security (MINES)*, vol. 2, 2009, pp. 370–373.

[103] D. Lentzen, G. Grutzek, H. Knospe, and C. Porschmann, "Content-based detection and prevention of spam over IP telephony - system design, prototype and first results," in *IEEE International Conference on Communications (ICC)*, 2011, pp. 1–5.

[104] J. Strobl, B. Mainka, G. Grutzek, and H. Knospe, "An efficient search method for the content-based identification of telephone-SPAM," in *IEEE International Conference on Communications (ICC)*, 2012, pp. 2623–2627.

[105] J. Bilien, E. Eliasson, J. Orrblad, and J.-O. Vatn, "Secure VoIP: Call establishment and media protection," in *Workshop on Securing Voice over IP*, Citeseer, 2005.

[106] A. Talevski, E. Chang, and T. Dillon, "Secure and mobile VoIP," in *International Conference on Convergence Information Technology*, 2007, pp. 2108–2113.

[107] C.-H. Wang, M.-W. Li, and W. Liao, "A distributed key-changing mechanism for secure voice over IP (VoIP) service," in *IEEE International Conference on Multimedia and Expo*, 2007, pp. 895–898.

[108] J. Kim, S. Yoon, H. Jeong, and Y. Won, "Implementation and evaluation of SIP-based secure VoIP communication system," in *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 2, Dec. 2008, pp. 356–360.

[109] F. Palmieri and U. Fiore, "Providing true end-to-end security in converged voice over IP infrastructures," *Computers & Security*, vol. 28, no. 6, pp. 433–449, 2009.

[110] C. A. Melchor, Y. Deswarte, and J. Iguchi-Cartigny, "Closed-circuit unobservable voice over IP," in *Annual Computer Security Applications Conference (ACSAC)*, 2007, pp. 119–128.

[111] M. Srivatsa, L. Liu, and A. Iyengar, "Preserving caller anonymity in voice-over-ip networks," in *IEEE Symposium on Security and Privacy*, 2008, pp. 50–63.

[112] G. Zhang and S. Fischer-Hübner, "Peer-to-peer VoIP communications using anonymisation overlay networks," in *Communications and Multimedia Security*, Springer, 2010, pp. 130–141.

[113] Z. Sabra and H. Artail, "Preserving anonymity and quality of service for VoIP applications over hybrid networks," in *IEEE Mediterranean Electrotechnical Conference (MELECON)*, IEEE, 2014, pp. 421–425.

[114] W. Mazurczyk and Z. Kotulski, "New VoIP traffic security scheme with digital watermarking," in *Computer Safety, Reliability, and Security*, Springer, 2006, pp. 170–181.

[115] D. Geneiatakis, G. Kambourakis, and C. Lambrinoudakis, "A mechanism for ensuring the validity and accuracy of the billing services in IP telephony," in *Trust, Privacy and Security in Digital Business*, Springer, 2008, pp. 59–68.

[116] N. Shekokar and S. Devane, "A novel approach to avoide billing attack on VoIP system," *World Academy of Science, Engineering and Technology*, vol. 62, pp. 993–997, 2010.

[117] A. D. Keromytis, "A comprehensive survey of voice over IP security research," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 514–537, 2012.

[118] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson, "Language identification of encrypted VoIP traffic: Alejandra y roberto or alice and bob?" In *USENIX Security Symposium*, vol. 3, 2007, pp. 43–54.

[119] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations," in *IEEE Symposium on Security and Privacy*, 2008, pp. 35–49.

[120] R. Zhang, X. Wang, X. Yang, and X. Jiang, "Billing attacks on SIP-based VoIP systems.," *USENIX Workshop on Offensive Technologies (WOOT)*, vol. 7, pp. 1–8, 2007.

[121] M. Langheinrich and R. Marti, "Practical minimalist cryptography for RFID privacy," *Systems Journal, IEEE*, vol. 1, no. 2, pp. 115–128, 2007.

[122] ——, "RFID privacy using spatially distributed shared secrets," in *Ubiquitous Computing Systems*, 2007, pp. 1–16.

[123] S. Cai, T. Li, C. Ma, Y. Li, and R. H. Deng, "Enabling secure secret updating for unidirectional key distribution in RFID-enabled supply chains," in *Information and Communications Security*, 2009, pp. 150–164.

[124] C. Lv, X. Jia, J. Lin, J. Jing, and L. Tian, "An efficient group-based secret sharing scheme," in *Information Security Practice and Experience*, 2011, pp. 288–301.

[125] C. Lv, X. Jia, J. Lin, J. Jing, L. Tian, and M. Sun, "Efficient secret sharing schemes," in *Secure and Trust Computing, Data Management and Applications*, 2011, pp. 114–121.

[126] J. G. Alfaro, M. Barbeau, and E. Kranakis, "Proactive threshold cryptosystem for EPC tags," *Ad hoc & sensor wireless networks*, vol. 12, no. 3-4, pp. 187–208, 2011.

[127] T. Li, Y. Li, and G. Wang, "Secure and practical key distribution for RFID-enabled supply chains," in *Security and Privacy in Communication Networks*, 2012, pp. 356–372.

[128] S. Abughazalah, K. Markantonakis, and K. Mayes, "Enhancing the key distribution model in the RFID-enabled supply chains," in *International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2014, pp. 871–878.

[129] T. Staake, F. Thiesse, and E. Fleisch, "Extending the EPC network: The potential of RFID in anti-counterfeiting," in *ACM Symposium on Applied Computing*, 2005, pp. 1607–1612.

[130] M. Lehtonen, F. Michahelles, and E. Fleisch, "Probabilistic approach for location-based authentication," in *International Workshop on Security for Spontaneous Interaction (IWSSI)*, vol. 2007, 2007.

[131] K. Ouafi and S. Vaudenay, "Pathchecker: An RFID application for tracing products in suply-chains," in *Workshop on RFID Security and Privacy (RFIDSec)*, vol. 9, 2009, pp. 1–14.

[132] M. Lehtonen, F. Michahelles, and E. Fleisch, "How to detect cloned tags in a reliable way from incomplete RFID traces," in *IEEE International Conference on RFID*, 2009, pp. 257–264.

[133] O. Nina, "Tracking based product authentication: Catching intruders in the supply chain," in *European Conference on Information Systems*, 2009, pp. 1017–1028.

[134] D. Zanetti, L. Fellmann, and S. Capkun, "Privacy-preserving clone detection for RFID-enabled supply chains," in *IEEE International Conference on RFID*, 2010, pp. 37–44.

[135] F. Kerschbaum and N. Oertel, "Privacy-preserving pattern matching for anomaly detection in RFID anti-counterfeiting," in *Radio Frequency Identification: Security and Privacy Issues*, Springer, 2010, pp. 124–137.

[136] E.-O. Blass, K. Elkhiyaoui, and R. Molva, "Tracker: Security and privacy for RFID-based supply chains.," in *Network and Distributed System Security Symposium (NDSS)*, 2011.

[137] F. Kerschbaum, "Public-key encrypted bloom filters with applications to supply chain integrity," in *Data and Applications Security and Privacy*, Springer, 2011, pp. 60–75.

[138] W. Xin, H. Sun, T. Yang, Z. Guan, and Z. Chen, "A privacy-preserving path-checking solution for RFID-based supply chains," in *International Conference on Information and Communications Security (ICICS)*, LNCS 7618, Springer, 2012, pp. 400–407.

[139] S. Cai, R. H. Deng, Y. Li, and Y. Zhao, "A new framework for privacy of RFID path authentication," in *Applied Cryptography and Network Security*, Springer, 2012, pp. 473–488.

[140] S. Cai, Y. Li, and Y. Zhao, "Distributed path authentication for dynamic RFID-enabled supply chains," in *Information Security and Privacy Research*, Springer, 2012, pp. 501–512.

[141] F. Guo, Y. Mu, W. Susilo, and V. Varadharajan, "A pre-computable signature scheme with efficient verification for RFID," in *Information Security Practice and Experience*, Springer, 2012, pp. 1–16.

[142] K. Elkhiyaoui, E.-O. Blass, and R. Molva, "Checker: On-site checking in RFID-based supply chains," in *ACM conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2012, pp. 173–184.

[143] H. Wang, Y. Li, Z. Zhang, and Z. Cao, "Two-level path authentication in EPC-global network," in *IEEE International Conference on RFID*, 2012, pp. 24–31.

[144] D. Moriyama, "An RFID authentication protocol with flexible path authentication," in *IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, 2013, pp. 1–6.

[145] M. Khalfaoui, R. Molva, and L. Gomez, "Secure product tracking in supply chain," in *Information Security and Cryptology*, LNCS 7763, 2013, pp. 351–370.

[146] D. Zanetti, S. Capkun, and A. Juels, "Tailing RFID tags for clone detection," in *Network and Distributed System Security Symposium (NDSS)*, 2013.

[147] J. Shi, S. M. Kywe, and Y. Li, "Batch clone detection in RFID-enabled supply chain," in *IEEE International Conference on RFID*, Apr. 2014, pp. 118–125.

[148] Z. Bilal and K. Martin, "A hierarchical anti-counterfeit mechanism: Securing the supply chain using RFIDs," in *Foundations and Practice of Security*, Springer, 2014, pp. 291–305.

[149] M. S. I. Mamun and A. Miyaji, "RFID path authentication, revisited," in *IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2014, pp. 245–252.

[150] J. Huang, X. Li, C. Xing, W. Wang, K. Hua, and S. Guo, "Dtd: A novel double-track approach to clone detection for RFID-enabled supply chains," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. PP, PP, 2015.

[151] F. Kerschbaum and L. Weiss Ferreira Chaves, "Encryption-enforced access control for an RFID discovery service," in *ACM symposium on Access Control Models and Technologies*, 2012.

[152] J. Shi, Y. Li, and R. H. Deng, "A secure and efficient discovery service system in EPCglobal network," *Computers & Security*, vol. 31, no. 8, pp. 870–885, 2012.

[153] B. Fabian, T. Ermakova, and C. Müller, "Shardis: A privacy-enhanced discovery service for RFID-based product information," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 3, pp. 707–718, 2012.

[154] J. Shi, Y. Li, W. He, and D. Sim, "Sectts: A secure track & trace system for RFID-enabled supply chains," *Computers in Industry*, vol. 63, no. 6, pp. 574–585, 2012.

[155] S. M. Kywe, Y. Li, and J. Shi, "Attack and defense mechanisms of malicious EPC event injection in EPC discovery service," in *IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, 2013, pp. 1–6.

[156] A. Juels, R. Pappu, and B. Parno, "Unidirectional key distribution across time and space with applications to RFID security," in *USENIX Security Symposium*, 2008, pp. 75–90.

[157] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[158] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[159] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," in *ACM Symposium on Theory of Computing (STOC)*, New York, NY, USA, 1985, pp. 291–304.

[160] L. C. Guillou and J.-J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory," in *Advances in Cryptology - EUROCRYPT*, Springer, 1988, pp. 123–128.

[161] C.-P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[162] M. Blum, "Coin flipping by telephone: A protocol for solving impossible problems," *Advances in Cryptology - CRYPTO*, 1982.

[163] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology - CRYPTO*, 1987, pp. 186–194.

[164] H. Bokharaei, A. Sahraei, Y. Ganjali, R. Keralapura, and A. Nucci, "You can SPIT, but you can't hide: Spammer identification in telephony networks," in *IEEE INFOCOM*, 2011, pp. 41–45.

[165] D. Waiting and N. Ventura, "A multilayered architecture for preventing automated spam in the IP multimedia subsystem," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2007, pp. 2140–2145.

[166] N. D'Heureuse, J. Seedorf, S. Niccolini, and T. Ewald, "Protecting SIP-based networks and services from unwanted communications," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2008, pp. 1–5.

[167] S. M. a. Salehin and N. Ventura, in *Blocking Unsolicited Voice Calls Using Decoys for the IMS*, Jun. 2007, pp. 1961–1966.

[168] Y.-S. Wu, S. Bagchi, N. Singh, and R. Wita, "Spam detection in voice-over-IP calls through semi-supervised clustering," in *IEEE/IFIP International Conference on Dependable Systems Networks (DSN)*, 2009, pp. 307–316.

[169] M. Falomi, R. Garroppo, and S. Niccolini, "Simulation and optimization of SPIT detection frameworks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2007, pp. 2156–2161.

[170] W. Yang and P. Judge, "VISOR: VoIP security using reputation," in *IEEE International Conference on Communications (ICC)*, 2008, pp. 1489–1493.

[171] K. Toyoda and I. Sasase, "SPIT callers detection with unsupervised random forests classifier," in *IEEE International Conference on Communications (ICC)*, Jun. 2013, pp. 2068–2072.

[172] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, 1967, pp. 281–297.

[173] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data*, ser. An Introduction to Cluster Analysis. 2009.

[174] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *International Conference on Very Large Data Bases (VLDB)*, 1994, pp. 144–155.

[175] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[176] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *Journal of Computational and Graphical Statistics*, vol. 15, no. 1, pp. 118–138, 2006.

[177] N. Eagle and A. Pentland, "Reality mining: Sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.

[178] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: `http://CRAN.R-project.org/doc/Rnews/`.

[179] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik, *Cluster: Cluster analysis basics and extensions*, 2013.

[180] V. D. Blondel, M. Esch, C. Chan, F. Clerot, P. Deville, E. Huens, F. Morlot, Z. Smoreda, and C. Ziemlicki, "Data for development: The D4D challenge on mobile phone data," *ArXiv preprint arXiv:1210.0137*, 2012.

[181] S. Bell, A. McDiarmid, and J. Irvine, "Nodobo: Mobile phone as a software sensor for social network research," in *IEEE Vehicular Technology Conference (VTC Spring)*, 2011, pp. 1–5.

[182] EPCglobal. (2014). EPC tag data standard (tds), [Online]. Available: http://www.gs1.org/gsmp/kc/epcglobal/tds.

[183] P. Avery, *The economic impact of counterfeiting and piracy*. OECD Publishing, 2008.

[184] K. Toyoda and I. Sasase, "Secret sharing based unidirectional key distribution with dummy tags in Gen2v2 RFID-enabled supply chains," in *IEEE International Conference on RFID*, 2015, pp. 84–90.

[185] R. J. McEliece and D. V. Sarwate, "On sharing secrets and reed-solomon codes," *Communications of the ACM*, vol. 24, no. 9, pp. 583–584, 1981.

[186] B. Mark. (2007). Blocking brute force attacks, [Online]. Available: http://www.cs.virginia.edu/~csadmin/gen_support/brute_force.php.

[187] I. RStudio. (2013). Easy web applications in R., [Online]. Available: http://www.rstudio.com/shiny/.

[188] F.-Y. Wang, D. Zeng, and L. Yang, "Smart cars on smart roads: IEEE intelligent transportation systems society update," *IEEE Pervasive Computing*, vol. 5, no. 4, pp. 68–69, 2006.

[189] K. Ali and H. Hassanein, "Passive RFID for intelligent transportation systems," in *IEEE Consumer Communications and Networking Conference (CCNC)*, 2009, pp. 1–2.

[190] T. Aura, P. Nikander, and J. Leiwo, "Dos-resistant authentication with client puzzles," in *Security Protocols*, Springer, 2001, pp. 170–177.

[191] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[192] point-at-infinity.org. (2014). Ssss: Shamir's secret sharing scheme, [Online]. Available: http://point-at-infinity.org/ssss.

[193] Impinj. (2014). Octane sdk - impinj support portal, [Online]. Available: https://support.impinj.com/hc/en-us/articles/202755268-Octane-SDK.

[194] R. Pass, "Bounded-concurrent secure multi-party computation with a dishonest majority," in *ACM Symposium on Theory of Computing (STOC)*, 2004, pp. 232–241.

Appendix A

Publication List

A.1 Journals

[1] K. Toyoda and I. Sasase, "Illegal interrogation detectable products distribution scheme in RFID-enabled supply chains," *IEICE Transactions on Communications*, (*accepted.*).

[2] N. Okazaki, K. Toyoda, E. Yokoyama, H. So, T. Katayama, and M. Park, "Countermeasure against fingerprinting attack in Tor by separated contents retrieval," *IEICE ComEX*, vol. 4, no. 12, pp.370-375, 2015.

[3] K. Iuchi, T. Matsunaga, K. Toyoda, and I. Sasase, "Secure parent node selection scheme in route construction to exclude attacking nodes from RPL network," *IEICE ComEX*, vol. 4, no. 11, pp.340-345, 2015.

[4] Y. Utsunomiya, K. Toyoda, and I. Sasase, "LPCQP: Lightweight private circular query protocol with divided POI-table and somewhat homomorphic encryption for privacy-preserving k-NN Search," *Journal of Information Processing*, (*accepted.*).

[5] M. Kurata, K. Toyoda, and I. Sasase, "Two-stage SPIT detection scheme with betweenness centrality and social trust," *IEICE ComEX*, vol. 4, no. 7, pp.239-244, 2015.

[6] T. Matsunaga, K. Toyoda, and I. Sasase, "Low false alarm attackers detection in RPL by considering timing inconstancy between the rank measurements," *IEICE ComEX*, vol. 4, no. 2, pp.44-49, 2015.

[7] R. Hattori, K. Toyoda, and I. Sasase, "Deterministic blocker tag detection scheme by comparing slot status in UHF RFID inventory management system," *IEICE ComEX*, vol. 4, no. 1, pp. 26-30, 2015.

[8] K. Toyoda and I. Sasase, "Unsupervised clustering-based SPITters detection scheme," *Journal of Information Processing*, vol. 23, no. 1, pp. 81-92, 2015.

[9] T. Koga, K. Toyoda, and I. Sasase, "Priority based routing for forest fire monitoring in wireless sensor network," *Journal of Telecommunications and Information Technology*, vol. 2014, no. 3, pp. 90-97, 2014.

[10] K. Toyoda and I. Sasase, "Divided challenge method to reduce the number of multiplication in FFS identification protocol," *Journal of Information Processing*, vol. 55, no. 5, pp. 1518-1529, 2014 (*in Japanese*).

A.2 Conferences Proceedings (peer-reviewed)

[1] K. Toyoda, M. Park, and N. Okazaki, "Unsupervised SPITters detection scheme for unbalanced callers," in *International Workshop on Big Data Processing in Online Social Network (BOSON)*, Crans-Montana, Switzerland, Mar, 2016 (*accepted.*).

[2] Y. Tamura, K. Toyoda, and I. Sasase, "Closer destination selection scheme for mobile sink and charger enabled WRSNs," in *IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, USA, Jan, 2016.

[3] A. Arno, K. Toyoda, and I. Sasase, "Accelerometer assisted authentication scheme for smart bicycle lock," in *IEEE World Forum on Internet of Things (WF-IoT)*, Milan, Italy, Dec, 2015.

[4] T. Hirayama, K. Toyoda, and I. Sasase, "Fast target link flooding attack detection scheme by analyzing traceroute packets flow," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Rome, Italy, Nov, 2015.

[5] S. Haruta, K. Toyoda, and I. Sasase, "Trust-based Sybil nodes detection with robust seed selection and graph pruning on SNS," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Rome, Italy, Nov, 2015.

[6] K. Toyoda and I. Sasase, "Superimposed SPIT fraud prevention with semi-supervised call pattern analysis," in *International Workshop on Vision, Communications and Circuits (IWVCC)*, Yokohama, Japan, Oct-Nov, 2015.

[7] H. Asahina, H. Yamamoto, K. Toyoda, and I. Sasase, "Path metrics for lower throughput fluctuation for video streaming service in wireless mesh networks," in *IEICE Asia-Pacific Conference on Communications (APCC)*, pp. 567-571, Kyoto, Japan, Oct, 2015.

[8] K. Iuchi, T. Matsunaga, K. Toyoda, and I. Sasase, "Secure parent node selection scheme in route construction to exclude attacking nodes from RPL network," in *IEICE Asia-Pacific Conference on Communications (APCC)*, pp. 304-308, Kyoto, Japan, Oct, 2015.

[9] M. Kurata, K. Toyoda, and I. Sasase, "Two-stage SPIT detection scheme with betweenness centrality and social trust," in *IEICE Asia-Pacific Conference on Communications (APCC)*, pp. 294-298, Kyoto, Japan, Oct, 2015.

[10] K. Toyoda and I. Sasase, "Illegal interrogation detectable EPC distribution scheme in RFID-enabled supply chains," in *IEEE International Conference on RFID Technology and Applications (RFID-TA)*, pp. 159-164, Tokyo, Japan, Sep, 2015.

[11] K. Toyoda and I. Sasase, "Secure and fast missing RFID tags identification with lightweight MAC and rateless coding," in *IEEE ICC Workshop on Security and*

Privacy for Internet of Things and Cyber-Physical Systems, pp. 10385-10390, London, UK, Jun, 2015.

[12] C. Inamura, K. Toyoda, and I. Sasase, "Monetary fair battery-based load hiding scheme for two households with one battery in automatic meter reading system," in *IEICE Information and Communication Technology Forum (ICTF)*, pp. 1-6, Manchester, UK, Jun, 2015.

[13] Y. Usami, K. Toyoda, and I. Sasase, "Reliable EH-WSNs based bridge monitoring system by adjusting sleep timing with beacon signal and forwarding overheard packets," in *IEICE Information and Communication Technology Forum (ICTF)*, pp. 1-6, Manchester, UK, Jun, 2015.

[14] K. Toyoda and I. Sasase, "Secret sharing based unidirectional key distribution with dummy tags in Gen2v2 RFID-enabled supply chains," in *IEEE International Conference on RFID (RFID)*, pp. 84-90, San Diego, USA, Jan, 2015.

[15] Y. Utsunomiya, K. Toyoda, and I. Sasase, "LPCQP: Lightweight private circular query protocol for privacy-preserving k-NN search," in *IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 65-70, Las Vegas, USA, Jan, 2015.

[16] T. Koga, K. Toyoda, and I. Sasase, "Adaptive relay selection with energy and channel information in energy harvesting WSNs," in *IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 657-662, Las Vegas, USA, Jan, 2015.

[17] R. Negishi, K. Toyoda, and I. Sasase, "Opportunistic routing protocol with grid-based relay slot selection in energy harvesting WSNs," in *Asia-Pacific Conference on Communications (APCC)*, pp. 1-5, Pattaya, Thailand, Oct, 2014.

[18] T. Matsunaga, K. Toyoda, and I. Sasase, "Low false alarm rate RPL network monitoring system by considering timing inconstancy between the rank mea-

surements," in *IEEE International Symposium on Wireless Communication Systems (ISWCS)*, pp. 1-5, Barcelona, Spain, Aug, 2014.

[19] R. Hattori, K. Toyoda, and I. Sasase, "Deterministic Blocker Tag detection scheme by comparing expected and observed slot status in UHF RFID inventory management systems," in *IEEE International Conference on High Performance Computing and Communications Workshop (HPCC Workshop)*, pp. 1-4, Paris, France, Aug, 2014.

[20] Y. Tamura, T. Koga, S. Hara, K. Toyoda, and I. Sasase, "Concurrent moving-based connection restoration scheme between actors to ensure the continuous connectivity in WSANs," in *IEEE International Conference on High Performance Computing and Communications Workshop (HPCC Workshop)*, pp. 1-4, Paris, France, Aug, 2014.

[21] T. Koga, S. Hara, K. Toyoda, and I. Sasase, "Priority based routing for forest fire monitoring in wireless sensor network," in *IEICE Information and Communication Technology Forum (ICTF)*, pp. 1-5, Poznan, Poland, May, 2014.

[22] K. Toyoda and I. Sasase, "SPIT callers detection with unsupervised Random Forests classifier," in *IEEE International Conference on Communications (ICC)*, pp. 2068-2072, Budapest, Hungary, Jun, 2013.

[23] K. Toyoda, Y. Kamiguchi, S. Inoue, and I. Sasase, "Efficient solution to decrease the effect of DoS attack against IP address ownership proof in Mobile IPv6," in *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pp. 1223-1227, Toronto, Canada, Sep, 2011.

A.3 Conferences Proceedings (in Japanese, without peer-review)

[1] A. Arno, K. Toyoda, and I. Sasase, "Accelerometer assisted authentication scheme for smart bicycle lock ," in *IEICE-CS*, Hokkaido, November, 2015.

[2] Y. Tamura, K. Toyoda, and I. Sasase, "Closer destination selection scheme for mobile sink and charger enabled WRSNs," in *IEICE-CS*, Hokkaido, November, 2015.

[3] T. Hirayama, K. Toyoda, and I. Sasase, "Fast target link flooding attack detection scheme by analyzing traceroute packets flow," in *IEICE-CS*, Hokkaido, November, 2015.

[4] A. Arno, K. Toyoda, Y. Watanabe, and I. Sasase, "Private key generation in two smart devices on the desk using vibration," in *IPSJ-CSS*, Nagasaki, October, 2015.

[5] K. Toyoda and I. Sasase, "Illegal interrogation detectable products distribution scheme in RFID-enabled supply chains," in *IEICE-CS*, Okinawa, July, 2015, (*invited talk.*).

[6] T. Matsunaga, K. Toyoda, and I. Sasase, "Privacy-preserving biometric authentication with homomorphic encryption by comparing partial bio-information," in *IEICE-CS*, Okinawa, July, 2015.

[7] K. Toyoda and I. Sasase, "Secret sharing-based key distribution with dummy tags in RFID-enabled supply chains," in *IPSJ-CSEC*, Ohita, May, 2015.

[8] S. Haruta, K. Toyoda, and I. Sasase, "Trust-based Sybil nodes detection with robust seed selection and graph pruning on SNS," in *IPSJ-CSEC*, Ohita, May, 2015.

[9] C. Inamura, K. Toyoda, and I. Sasase, "Power usage hiding technique considering monetary fairness in the automatic meter reading system," in *IPSJ-CSEC*, Tokyo, December, 2014.

[10] Y. Utsunomiya, K. Toyoda, and I. Sasase, "Low-complexity privacy-preserving k-POIs search scheme by dividing and aggregating POI-table," in *IPSJ-CSS*, Hokkaido, November, 2014.

[11] K. Toyoda and I. Sasase, "Secure and fast UHF RFID missing tags detection with rateless coding," in *IEICE-CS*, Hokkaido, November, 2014.

[12] T. Koga, K. Toyoda, and I. Sasase, "Flexible relay selection with side information Regarding energy and channel in energy harvesting WSNs," in *IEICE-CS*, Hokkaido, November, 2014.

[13] Y. Usami, K. Toyoda, and I. Sasase, "Sleep timing adjusting scheme for high packet delivery rate in bridge monitoring by energy harvesting," in *IEICE-CS*, Hokkaido, November, 2014.

[14] K. Toyoda, and I. Sasase, "Preventing voice-based spam by hijacked account," in *IEICE-CS*, Hokkaido, November, 2013.

[15] S. Hara, K. Toyoda, and I. Sasase, "Adjacent small cell pre-activation scheme based on UE's mobility prediction in heterogeneous cellular network," in *IEICE-CS*, Hokkaido, November, 2013.

[16] T. Koga, S. Hara, K. Toyoda, and I. Sasase, "Priority based routing for forest fire monitoring in wireless sensor network," in *IEICE-CS*, Hokkaido, November, 2013.

[17] K. Toyoda, and I. Sasase, "SPIT callers detection with unsupervised Random Forests classifier," in *IPSJ-CSEC*, Kanagawa, December, 2012.

[18] K. Toyoda and I. Sasase, "Evaluation of calculation amount and trade-off in FFS authentication with divided challenge method to mitigate the effect of DoS attack," in *IEICE-CS*, Kagoshima, July, 2012, (*invited talk.*).

[19] E. Fushimoto, K. Toyoda, Y. Kamiguchi, and I. Sasase, "Solution to prevent the failure of IP address ownership proof in mobile IPv6 by encrypting IP address using hash value," in *IEICE-Society*, Okayama, March, 2012.

[20] K. Toyoda, Y. Kamiguchi, S. Inoue, and I. Sasase, "Efficient solution to decrease the effect of DoS attack against IP address ownership proof in Mobile IPv6," in *IEICE-CS*, Kagoshima, April, 2011.

A.4 Awards

[1] 2015 IEICE Communication Systems Encouragement Award

[2] 2015 30th Telecom System Technology Encouragement Award

[3] 2012 IEICE Communication Systems Encouragement Award

A.5 Others

[1] K. Toyoda, "Unsupervised clustering-based SPITters detection scheme (invited talk)," in *IEICE Information and Communication Technology Forum (ICTF)*, Poznan, Poland, May, 2014. (*invited talk.*)

[2] K. Toyoda and I. Sasase, "Secure products distribution with dummy tags in RFID-enabled supply chains," *IEEE Internet of Things Journal*, (*under review.*)