UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

SECURING DATA TRANSMISSION IN INTERNET OF THINGS

by

Xuan Zha

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2019

Certificate of Authorship/Originality

I, Xuan Zha declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney. This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. I certify that the work in this thesis has not been previously submitted for a degree nor has it been submitted as a part of the requirements for other degree except as fully acknowledged within the text. This thesis is the result of a research candidature jointly delivered with Beijing University of Posts and Telecommunications as part of a Collaborative Doctoral Research Degree. This research is supported by the Australian Government Research Training Program.

Production Note:

Signature: Signature removed prior to publication.

Date: 26/09/2019

Dedication

To my dear husband

To my loving parents

To my supportive supervisors

To my wonderful friends

Acknowledgements

I sincerely convey my deepest gratitude to my principal supervisor Prof. Y. Jay Guo for his experienced supervision throughout my doctoral program. I would show my honest and sincere appreciation to my co-supervisor Prof. Ren Ping Liu and Dr. Wei Ni from Commonwealth Scientific and Industrial Research Organisation (CSIRO). Without their consistent support and supervision, I would not have been able to complete this thesis. I would also like to express my honest appreciation to my supervisors at Beijing University of Posts and Telecommunications (BUPT) for their experienced supervision and continuous encouragement.

I thank the University of Technology Sydney (UTS) and the Faculty of Engineering and IT (FEIT) for providing me an IRS Scholarship throughout my doctoral program. I would also like to thank staff members, previous and current colleagues, friends at UTS and CSIRO for their kind help. Special thanks to Dr. Xu Wang, Dr. Bo Song, Dr. Saber Yu, Dr. Ping Yu, Dr. Shangjing Lin, Dr. Chen Qing, Dr. Shoulu Hou, Dr. Xin Yuan, Dr. Zishan Liu, Dr. Xinchen Lyu, Dr. Haihan Sun, Dr. Fangfang Dai, and Dr. Yixun Hu.

Last but not least, I am deeply grateful to my husband Xiaoming Peng, my mother Guiying Zhang, my father Rihui Zha, my aunt Wenfang Zha and the rest of my family for their support and constant encouragement.

Xuan Zha
Sydney, Australia, 2019.

List of Publications

The author has published four journal papers, including two IEEE transaction journal papers, and three international conference papers, as the (co-)first author. The impact factor (IF) of the journal papers is also stated*.

Journal Papers

J-1. **X. Zha**, W. Ni, X. Wang, R. P. Liu, Y. J. Guo, X. Niu and K. Zheng, "The Impact of Link Duration on the Integrity of Distributed Mobile Networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2240-2255, Sep. 2018. (IF: 5.824)

J-2. **X. Zha**, W. Ni, K. Zheng, R. P. Liu and X. Niu, "Collaborative Authentication in Decentralized Dense Mobile Networks With Key Predistribution," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2261-2275, Oct. 2017. (IF: 5.824)

J-3. X. Wang, **X. Zha**, W. Ni, R. P. Liu, Y. J. Guo, X. Niu and K. Zheng, "Survey on Blockchain for Internet of Things", *Computer Communications*, vol. 136, pp. 10-29, Feb. 2019 (IF: 2.613) **(X. Zha and X. Wang contributed equally to this paper)**

J-4. **X. Zha**, X. Wang, W. Ni, R. P. Liu, Y. J. Guo, X. Niu and K. Zheng, "Blockchain for IoT: the Tradeoff between Consistency and Capacity", *China Journal on Internet of Things*, vol. 1, no. 1, pp. 21-33, 2017

J-5. X. Wang, **X. Zha**, W. Ni, R. P. Liu, Y. J. Guo, X. Niu and K. Zheng, "Game Theoretic Suppression of Forged Messages in Online Social Networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Accepted, 2019.

*IF: Impact Factor. Refer to http://wokinfo.com/essays/impact-factor/ for details.

J-6. K. Zheng, X. Wang, **X. Zha** and H. Xiao, "A New Network Coding Mechanism Balancing Coding Opportunities, Energy and QoS in WSNs," *China Communications*, vol. 11, no. 6, pp. 108-118, June 2014.

Conference Papers

C-1. **X. Zha**, X. Wang, W. Ni, R. P. Liu, Y. J. Guo, X. Niu and K. Zheng, "Analytic Model on Data Security in VANETs", *Int. Symposium on Communications and Information Technologies*, pp. 1-6, Sep. 25-27, 2017

C-2. **X. Zha**, K. Zheng, D. Zhang, "Anti-Pollution Source Location Privacy Preserving Scheme in Wireless Sensor Networks,", *Proc. IEEE Int. Conf. on Sensing, Communication, and Networking (SECON)*, pp. 1-8, Jun. 27-30, 2016

C-3. **X. Zha**, W. Ni, R. P. Liu, K. Zheng and X. Niu, "Secure Data Transmission and Modelling in Vehicular Ad Hoc Networks", *Proc. IEEE Int. Conf. on Globecom Workshop*, pp. 1-6, Dec. 6-10, 2015.

C-4. X. Wang, **X. Zha**, G. Yu, W. Ni, R. P. Liu, Y. J. Guo, X. Niu and K. Zheng, "Attack and Defence of Ethereum Remote APIs", *Proc. IEEE Int. Conf. on Globecom Workshop*, Dec. 9-13, 2018

C-5. G. Yu, X. Wang, **X. Zha**, J. A. Zhang, R. P. Liu, "An Optimized Round-Robin Scheduling of Speakers for Peers-to-Peers-based Byzantine Faulty Tolerance", *Proc. IEEE Int. Conf. on Globecom Workshop*, Dec. 9-13, 2018

Contents

List of Figures

Abbreviation

IoT: Internet of Things

VANET: Vehicular Ad hoc Network

WSN: Wireless Sensor Network

GEV: Global Encoding Vector

LEV: Local Encoding Vector

2D: Two-dimensional

3D: Three-dimensional

4D: Four-dimensional

CSMA/CA: Carrier Sense Multiple Access/Collision Detection

DCF: Distributed Coordination Function

MAC: Message Authentication Code

ACK: Acknowledgement

NACK: Non-acknowledgement

ABSTRACT

SECURING DATA TRANSMISSION IN INTERNET OF THINGS

by

Xuan Zha

The Internet of Things (IoT) is poised to transform our lives and unleash enormous economic benefit. With the rise in the number of connected IoT devices, the potential vulnerabilities in IoT increase as well. The IoT security faces severe challenges arising from its specific characteristics. This thesis studies the location privacy protection and secure data transmission issues in IoT to ensure the data confidentiality, integrity, non-repudiation and availability. Markov models are proposed to analyse the network performance of secure data transmission mechanisms, providing quantified criteria for selecting appropriate secure transmission protocols in various network environments. The main contributions of this thesis are as follows,

(1) An anti-pollution source-location privacy scheme is proposed to tackle the conflict between the source-location protection and authentications. The proposed protocol consists of key predistribution mechanisms and a homomorphic signature algorithm, for filtering out polluted and dummy packets at intermediate nodes while concealing the packet trajectory. The proposed protocol improves the message delivery rate and saves energy as compared with previous works.

(2) A probabilistic encrypted data transmission protocol is proposed to transmit messages in confidentiality in an adaptive manner. It avoids the communication overhead caused by handshaking in previous data encryption protocols. In addition, a three-dimensional (3D) Markov model is constructed to analyse the impact of wireless communication collisions and key predistributions on the performance of encrypted data transmissions. The analysis and simulation results prove the accuracy of the 3D Markov model.

(3) An authentication protocol is proposed in opportunistic routing based IoT networks. In order to improve the authentication efficiency, the proposed protocol generates authentication information based on the combination of the new message and previous non-conflict but unauthenticated messages while attempting different keys. A new 3D Markov model is designed to accurately capture the interaction process among non-coordinated transmissions, key selections and packet lifetime. The proposed protocol substantially improves the tolerance against changing topologies and resistance against collusion attacks, as compared to the prior art.

(4) A four-dimensional (4D) Markov model is designed to analyse the impact of dynamic topology on opportunistic authentication protocols. Three cross-layer data authentication protocols are proposed with opportunistic authentication and channel access coupled to different extent. According to the simulation results, the 4D model is general and accurate. The analysis results prove that opportunistic data authentication protocols significantly improve the authentication rate, reduce authentication delay and enhance scalability to dense mobile distributed networks.

Dissertation directed by Professor Y. Jay Guo
Faculty of Engineering and Information Technology

Chapter 1

Introduction

This chapter provides an overview of the targeted network, Internet of Things (IoT), together with an analysis of its features and an introduction to Vehicular Ad hoc Networks (VANET) and Wireless Sensor Networks (WSN), two typical IoT networks. Research motivation, contributions of the work as well as thesis structure are also presented in this chapter.

1.1 Internet of Things

The Internet of Things (IoT) connects a large number of devices through a ubiquitous wireless network. By connecting objects and people, IoT transforms the physical world into a digital system. There is currently no consensus on a unified definition of IoT, as it encompasses a host of concepts. The term "Internet of Things" was first proposed by Kevin Ashton at Auto-ID Center of Massachusetts Institute of Technology (MIT) in 1999 [40], and afterwards in 2005, the International Telecommunication Union (ITU) released its Internet of Things report, which extended the concept of IoT and predicted the emergence of an IoT era [2]. Smart computing technologies enable things and humans to intercommunicate with omnipresent network and networked devices. Entities in the Internet of Things communicate with each other to both provide and receive services at any time and any place. With its potential for a well-connected future, IoT applications can affect every aspect of daily life. IoT has been widely developed in military, industry, agriculture, smart grid, transportation, logistics and other areas [126, 14]. It is estimated that by 2025, IoT may generate $11 trillion of global GDP [77].

A typical IoT architecture consists of *Perception, Network,* and *Application Layers*, from bottom to top [8]. The perception layer is an indispensable part of a variety of IoT applications [90]. It consists of sensors and actuators collecting and processing environmental information to perform functions, such as querying temperature, location, motion, acceleration. Various types of end devices can be adopted in the perception layer to bridge the physical world and the digital world. Typical end devices include Radio-Frequency IDentification (RFID) [138], wireless sensors and actuators [61, 143], vehicles [136, 86], Unmanned Airborne Vehicles (UAV) [66, 65] and so on. For example, an RFID tag is a small microchip attached to an antenna. By attaching RFID tags to objects, the objects can be identified, tracked, and monitored in logistics, retailing, and supply chain applications. The network layer is responsible for connecting smart things, network devices, and servers. The application layer creates and manages specific services to meet the IoT application requirements. It covers various application domains, including transportation and logistics, healthcare, smart environment (including smart home), and personal and social applications [14]. There are other IoT architectures proposed with the development of IoT, as summarized in [8]. For example, a four-level architecture consists of *Perception, Networking, Service,* and *Interface Layers* [129]. Among all these proposed architectures, however, the perception layer is the most distinctive and indispensable layer of IoT, helping to differentiate IoT from the traditional concept of Internet, upon which IoT was born.

Implemented with heterogeneous end devices and protocols, IoT networks have some IoT-specific characteristics as follows,

- Enormous number of nodes and big IoT data: The number of IoT devices will continuously increase. The number of connected devices in IoT is expected to increase up to 20.4 billion by 2020 [1]. IoT faces not only a large number of nodes but also growing demand for capacity, as numerous end devices sense and collect mass data. Such characteristic raises a high demand on the robust of data collection and processing technologies against the growth of the data and users.

- Decentralization: Decentralization and heterogeneity are the two major characteristics of IoT [116, 101]. Decentralization is essential given the large number of IoT nodes because the data to be processed at the same time is considerably huge [114]. Decentralized algorithms in IoT, e.g., clustering algorithms in wireless sensor network (WSN) and decentralized computing, can contribute to the capacity and scalability of IoT [114].

- Heterogeneity: IoT integrates heterogeneous networks and technologies based on the traditional computer network. Data is collected by heterogeneous end devices including low-capacity nodes (e.g., sensors and RFID) and powerful nodes (e.g., vehicles), before transmitted to the processing platform via various wireless communication protocols, such as ZigBee and NB-IoT [63, 99]. However, the relevant standard specifications are still under development, and a unified security system has not been established. In this case, the heterogeneous devices and protocols in IoT pose great challenges to security, such as data encryption, data authentication, and key distribution.

- Unstable and unpredictable connections: The wireless communication in IoT can be divided into an infrastructure mode and an ad-hoc mode. In the case of the infrastructure mode, the mobility of devices leads to a growth of signaling and control messages, due to the fact that packets are transmitted among end devices and network infrastructures (base stations) [144]. In the case of the ad-hoc mode, the network does not rely on pre-existing infrastructures and each node forwards data for other nodes [25]. In this case, the mobility of devices can lead to unpredictable network connectivities. Even if the devices are stable, the network connections vary with the failure of IoT devices. For example, sensors in WSN with tiny size and limited battery may run out of battery or switch to the sleep/idle mode for the energy concern. The typical unreliable wireless links to IoT devices also lead to unstable connections.

1.2 Typical IoT Networks

IoT network prevails with its ability to interconnect numerous devices possessing various sensing and computing abilities with little human interventions [96]. Sensing and actuating devices form heterogeneous IoT networks to provide various applications. Typical IoT applications include smart home, smart transport, eHealth and smart grid [90]. These applications have common characteristics such as an enormous number of nodes and data, decentralization, heterogeneity, and unpredictable connections, as aforementioned. On the other hand, each application has its own characteristics. Two kinds of typical IoT networks, Wireless Sensor Networks (WSN) and Vehicular Ad hoc Networks (VANET), are introduced in this section.

1.2.1 Wireless Sensor Networks

Wireless sensor networks are widely seen in a variety of IoT applications, such as environmental monitoring, target field imaging, smart home, traffic monitoring, and disaster management. Nodes in a wireless sensor network can be divided into three categories: sensors, the sink, and the management center. Sensors are low-cost autonomous devices with limited storage, computational ability, and energy supply. A large number of sensors, with capabilities to produce high-quality information of the physical environment, are randomly deployed in an area to monitor the presence of a predefined event. When a sensor senses the event, it generates and routes a message to inform the sink about the event. The sink is capable to aggregate and process the collected data. The management center, with the highest level of security, is responsible for the configuration and management of the entire network.

Distinct from the traditional IP-based Internet, WSN is a data-centric network, which indicates that data and its contextual information such as the time, location and transmission path of the packet is more critical than the node identity [27]. Once a defined event happens in an event-triggered WSN, the sensor nearest to the event location sends out its sensory data to the sink as the source node. In other words, the contextual information of the packet indicates the information about the monitored event. Therefore, the contextual information, including the transmission time [32], source location [67], and the destination location [134, 74], as well as the data, is private and requires protections in WSN.

Take the source location privacy (SLP) as an example. The Panda-Hunter game [32] is a typical application about SLP. In this application, scientists deploy sensors in the wild field to monitor the wild pandas, while hunters wander around to hunt them. When a panda appears, the sensor closest to the panda sends an alert to the sink node. As hunters eavesdrop on the network traffic and reversely follow the transmission path to trace back to the source node, they can hunt the panda around the source location. Therefore, preserving source location privacy is of practical importance. If a packet is not encrypted, adversaries can obtain the source location directly through eavesdropping on the traffic in the open channel. Even if the packet is encrypted, adversaries can trace back to the source location hop-by-hop by performing time-correlation, size-correlation and content-correlation analysis on the encrypted packets. The disclosure of the source location indicates that the monitored event is open to adversaries as well. Protecting the data confidentiality and breaking the traffic traceability is key to preserving source location privacy.

The process of data collection and transmission is also vulnerable to interruption, interception, modification, and fabrication, due to the wireless communication media

[28]. If the network is lack of access controls and authentications, adversaries can insert or tamper with messages to degrade the network performance. Given the fact that conventional defense mechanism proposed for the Internet cannot be directly applied in WSN to prevent adversaries due to the WSN-specific characteristics such as data-centric and limited capacity [75], the research on preserving privacy and security remains essential to secure wireless sensor networks.

1.2.2 Vehicular Ad hoc Networks

Vehicular Ad hoc Network (VANET), a subclass of mobile ad hoc networks (MANET), form an important part of the intelligent transportation systems (ITS). Vehicles driving on and infrastructures built along the road form an ad hoc network where data is relayed among them. With the real-time traffic information exchanged, VANET has the potential to provide both safety-critical and non-safety-critical services, thereby improving the safety and experience of driving [71]. Applications have been developed in areas such as lane change assistance, road safety warning, real-time traffic analysis, and road navigation. There are two types of nodes in VANET, i.e., intelligent vehicles and Road Side Units (RSU). Each intelligent vehicle is equipped with GPS and the On-Board Unit (OBU). OBU provides the necessary network interfaces for data exchange among vehicles, which is known as the Vehicle-to-Vehicle (V2V) communication, or for data exchange between vehicles and the RSU, which is known as the Vehicle-to-Infrastructure (V2I) communication [31]. RSUs are connected to form a backbone network [68].

Different from the stable WSN, VANET is a special class of mobile ad hoc networks (MANET) while exhibiting drastically different characteristics from MANETs. A case in point is that vehicles do not move arbitrarily but follow a predictable trajectory, which is constrained by predefined roads [110]. Moreover, the topology of VANET is more dynamic than that of MANET [35], due to the high speed of vehicles. It is pointed out in [107] that more than 57% links between vehicles last less than 15 seconds. The link between two vehicles breaks when one mobile vehicle moves out of the other one's communication range or the wireless communication is interrupted. Therefore the frequently changing topology and the uncoordinated transmissions still affect the communication among vehicles and RSUs [121], even if vehicles are much capable in terms of computation, communication and storage. All the above characteristics make the Internet transmission protocols unsuitable in VANET. IEEE has specified IEEE 802.11p [3] as the standard for wireless access in VANET [115]. Dedicated Short Range Communication (DSRC) enables wireless communication between vehicles [30] while CSMA/CA is applied to tackle uncoordinated transmissions.

As VANET is closely related to human lives, VANET security is a major concern in the deployment, especially large-scale deployment, of VANETs. VANET nodes, such as vehicles, react according to the data which is coordinately transmitted throughout the entire network. If adversaries can arbitrarily inject, drop or modify data, the network may make wrong decisions, leading to traffic congestion and even accidents. This is the very reason that data reliability and security is one prerequisite for VANET security. However, communicating in an open-access environment makes security issues a real challenge [97]. VANET lacks in defense mechanisms such as gateways and firewalls, which are widely used to protect traditional wired networks. In addition, VANET not only inherits vulnerabilities of MANET but also faces unknown attacks due to its unique features such as the fast-changing topology. Therefore, specific mechanisms need to be developed to meet the primary requirement of ensuring security prior to the practical deployment of VANETs [97].

Primary requirements for ensuring VANET security include authorization and privacy protection. Authentication algorithms have been widely studied to guarantee authorization [76, 100, 70], for example authentication algorithms generate signatures or message authentication codes based on transmitted data to verify whether the data originates from the stated source or it is be revised by any node during transmission. However, the use of authentication may lead to privacy risks. Vehicular data contains personal information of drivers, such as the daily routine, personal interests, and work location. Adversaries could conceal sensitive information of drivers by collecting and analyzing data transmitted in the wireless channel [102, 103]. Also, the information used for authentication may reveal the whereabouts of a specific driver at a specific time [36]. The balance between authentication and privacy has become one of the main challenges of implementing VANETs [127].

1.3 Research Motivation

With the rising number of connected IoT devices, potential vulnerabilities in IoT increase as well [6]. In recent years, the world has seen the successful launch of attacks on IoT infrastructures, e.g., Mirai, a worm-like family of malware that infected IoT devices and corralled them into a denial-of-service (DDoS) botnet, overwhelmed high-profile targets with massive distributed DDoS attacks [12]. The lack of reliable encryption and authentication mechanisms in IoT devices is a contributing factor in violating IoT security [19].

Adversaries could pose threats to user privacy through eavesdropping and traffic analysis. The lack of encryption presents a threat to data confidentiality and user privacy, as adversaries can directly gather sensitive information based on the data transmitted and stored in plaintext. In the case that encryption is applied in the network, traffic patterns still reveal sensitive information of users. Previous works [80, 10] introduced dummy traffic to conceal traffic patterns. However, the dummy traffic degrades transmission efficiency and shortens network lifetime due to the uncoordinated wireless communication and the restricted energy of some sensory nodes. Although several solutions [133, 118] were proposed to filter out dummy traffic in the privacy-preserving schemes, these solutions were not able to filter out dummy packets until they arrive at a specific proxy node. Network coding based solutions [43, 42], on the other hand, have the potential to resist content-correlation analysis due to its inherent coding/mixing characteristic. However, all the above solutions, especially the network coding based solutions, are vulnerable to pollution attacks [62], as in fact a message is relayed multiple times before reaching its intended destination [22, 51] and during this process pollution attackers can tamper with the message. Thus, it is crucial that, each time relayed, the message is verified to be genuine. To this end, authentication is of paramount importance as it is able to prevent malicious falsifications and provide data integrity in decentralized IoT networks [121]. However, as authentication requires the revealing of packet information, which conflicts with the target of privacy protection [36], the issue of preserving both privacy and integrity remains unsolved in decentralized IoT networks.

Another prominent challenge for both privacy and authentication in decentralized IoT networks such as VANET arises from the fast and frequently changing topology which is typically unknown a priori. In many cases, a node can have constantly changing neighbours. The one which can relay traffic for the node can be unknown a priori and change instantly, therefore route discovery needs to be accomplished in a short time while traffic is forwarded [22, 51, 5]. This leaves little time for establishing keys between neighbouring nodes for security concern. There is another challenge arising from the uncoordinated, contention-based transmissions in the networks [3]. Transmission collisions often occur and become increasingly severe, as the network gets dense and traffic burden gets heavy [16]. Randomly delayed or backed-off (re)transmissions, such as CSMA/CA, are often adopted to reduce collisions. However, the resultant delays and the residual packet loss are destructive to the key establishment. To the best of our knowledge, these challenges are yet to be addressed holistically. Existing encryption and authentication techniques either necessitate the presence of trusted third-parties to enable the key establishment [92, 48], or undergo significant authentication delays [91] or prohibitive computation overhead [111] or extra communication overhead for real-time delivery of keys and certificates [100, 70].

Unsuccessful (re)transmissions can be caused by either unmatched keys or transmission collisions. No existing key designs, e.g., [92, 100, 70, 38, 9], have taken this into account. The complex reasons to unsuccessful (re)transmissions also make the security analysis of decentralized IoT challenging. Originally developed to evaluate collisions in IEEE 802.11a/e [72], Markov models were extended to mobile networks [85] and VANETs [121], but cannot capture security. Other models that did analyse security were based on oversimplified transmission channels, such as ON/OFF wireless links [131]. The oversimplifications on transmission channels degrade the accuracy of the models.

This thesis provides privacy and secure transmission solutions for decentralized IoT networks in Chapter 3 to Chapter 6.

1.4 Contributions of the Work

The main contributions of this thesis are discussed in four separate chapters. Firstly, an anti-pollution source-location privacy solution is studied, which filters out dummy and polluted packets to provide efficient secure transmission. In the proposed solution, encoded and encrypted packets prevent passive attackers from revealing location privacy while homomorphic signatures prevent active attackers from tampering with packets. Simulation results demonstrate that the proposed location privacy solution saves energy and improves message transmission ratio. Furthermore, an encryption data transmission protocol is also proposed in this thesis to protect data privacy in decentralized IoT networks. A novel Markov model, which captures the impact of uncoordinated wireless transmissions and key selections, is designed to analyse the proposed protocol. Then this thesis provides opportunistic authentication transmission protocols to guarantee data integrity in dynamic IoT networks. Opportunistic authentication transmission protocols collaborated with the opportunistic routing and the location-aware routing protocol are studied separately. A general analytic framework is developed to quantify generic authentication protocol designs and provide analysis of the impact of link duration on the authentication protocols. The above contributions have been reported in the author's publications (see the section *List of Publications* for details).

i. The location privacy faces challenges arising from passive and active attackers. Previous works used authentication algorithms and dummy traffic to defend against active and passive attackers, respectively. However, the conflict between privacy and authentication degrades the performance of previous location privacy solutions in the presence of active attackers. This thesis proposes an anti-pollution location privacy solution, which integrates a homomorphic authentication algorithm with network coding based dummy traffic, to guarantee both data integrity and source-location privacy. Homomorphic signatures generated by the homomorphic authentication algorithm filter out dummy and polluted traffic during transmission, while the network coding based dummy traffic conceals the traffic pattern from the eavesdroppers. The proposed solution also designs probabilistic key predistribution schemes to distribute keys for the homomorphic authentication algorithm, with the probabilistically distributed keys prevent internal attackers from distinguishing the real traffic from dummy traffic.

ii. Data privacy is generally guaranteed by the use of encryption algorithms. However, challenges to the encryption in IoT arise from the fast and frequently changing topology, as well as the uncoordinated transmissions. This thesis proposes an opportunistic encryption algorithm to provide data confidentiality in dynamic IoT networks. To avoid the extra communication overhead for finding encryption keys between temporary neighbours, the proposed algorithm predistributes keys among nodes and tries keys with messages until a matched one is found at the receiver. New acknowledgement (ACK) messages are designed to distinguish the cause of a failed (re)transmission between a packet collision and a mismatched key. A transmitter adaptively switches between backing off transmissions and changing keys to increase success rate with matched keys. A new 3-dimensional (3D) Markov chain model is also proposed, which captures the impact of interactions between collisions and key selections on the encrypted transmission.

iii. Authentication is key to protecting data integrity in decentralized IoT networks. To tackle the fast-changing topology, this thesis proposes an opportunistic authentication protocol which embraces opportunistic routing. Delivered, unexpired but unauthenticated messages within a link duration can all be authenticated retrospectively at the receiver, if the receiver is preloaded with the key matching the one the transmitter has adopted. In this sense, the communication overhead for authentication can be reduced to be independent of the number of keys tried. The analytical results, based on a 3D Markov chain, confirm the tolerance of the proposed protocol against changing topologies, as well as the substantially improved resistance against collusion attacks, as compared to the prior art. The performance comparisons between symmetric and asymmetric keys are also provided in terms of authentication performance and the resistance against collusion attacks.

iv. As there lacks a generic analysis model to analyse a variety of on-the-fly authentication protocol designs, this thesis proposes a new four-dimensional (4D) model to characterize an ongoing opportunistic authentication process, where a receiver can be valid for only a short duration and replaced frequently as the result of mobility. The impact of dynamic topology is quantified as the link duration and captured by the 4D model. Furthermore, three on-the-fly authentication protocols, coupling opportunistic authentication and channel access to different extents, are proposed and compared in location-aware routing based IoT networks. The analysis reveals that cross-layer consideration to jointly design retransmissions and rekeying is the key to achieve the significant gain of on-the-fly authentication in distributed IoT networks.

1.5 Thesis Organization

The rest of this thesis is organised as follows:

- Chapter 2 analyses security threats to IoT from the bottom layer to the top layer. It is then followed with an analysis of the fundamental requirements for IoT security and cryptography based solutions. Furthermore, related works on how to secure IoT data transmissions are present in terms of source-location privacy, secure data transmission protocols, and analysis models. By analyzing existing security solutions, unsolved issues are pointed out to identify the need for studying security protocols and analysis models.

- Chapter 3 investigates the issue of protecting data integrity in source-location privacy solutions. This chapter proposes an Anti-Pollution Source-Location Privacy scheme (AP-SLP) to filter out dummy and polluted packets during transmissions. AP-SLP comprises of a homomorphic signature algorithm, which signs network coded packets and verifies signatures to recognize the corresponding packet type, and probabilistic key predistribution schemes, which provide keys for the homomorphic signature algorithm.

- Chapter 4 studies the problem of guaranteeing data confidentiality in dynamic IoT networks with constantly changing topologies. A new encryption data transmission protocol based on the opportunistic key distribution is proposed. A transmitter adaptively switches between backing off transmissions and changing keys to increase success rate with matched keys. A new 3D Markov chain model is also proposed, which captures the interactions among collisions and key selections.

- Chapter 5 explores the problem of providing data integrity in opportunistic routing based dynamic IoT networks. This chapter proposes a new protocol of joint transmission and authentication. To reduce the communication overhead for authentication, a node is designed to increasingly combine collision-free yet unauthenticated messages and a new message for digital signature or message authentication code (MAC) generation, while trying different keys on-the-fly. A 3D Markov chain is proposed to capture interactions among collisions, key selections, and the lifetime of unauthenticated messages.

- Chapter 6 solves the problem of providing data integrity in location-aware routing based mobile IoT networks, where the dynamic topology is quantified as the link duration. A 4D Markov model is designed to capture the impact of the link duration on the authentication protocols. A set of authentication transmission protocols, which couple opportunistic authentication and channel access to different extents, are proposed and analysed. The analysis reveals that the cross-layer consideration to jointly design retransmissions and rekeying is key to unlocking the potential of opportunistic authentication and outperforming the prior art over a wide range of link duration.

- Chapter 7 concludes contributions of the thesis and gives the future work.

Chapter 2

Background Studies and Related Works

Although IoT attracts global attention in recent years, IoT also faces severe network attacks. The Internet, on which IoT is based, is inherently insecure, where data security was an afterthought in the design, as can be evident from continual patches and manual handling [41]. Moreover, IoT has a substantially different architecture from the Internet, extending network connectivity and computing capability to objects with limited computing power, such as sensors and throw-away items, and allowing these devices to generate, exchange and consume data with minimal human interventions [106]. Simply extending computationally demanding and costly Internet security solutions to IoT is neither scalable nor practical [126]. In this chapter, the IoT security is surveyed from aspects of the IoT attack models, typical attacks on IoT, security assurances, and cryptographic-based solutions in Section 2.1, followed by related works on protecting IoT data transmission security in Section 2.2.

2.1 Survey on IoT Security

Specific characteristics of IoT make data security a severe problem in IoT [14]. Firstly, many IoT devices are deployed in human unfriendly and unattended areas, and it can be impossible to keep an eye on the huge number of devices all the time. This makes devices vulnerable to multi-dimensional harms [45]. For example, adversaries may physically capture and control these devices to invade IoT networks [11]. Furthermore, traditional security mechanisms [73], such as the asymmetric encryption, are computationally demanding for IoT devices with limited abilities. Data from sensors can be stored, forwarded and processed by many different intermediate systems, which increases the risk of being tampered and forged. Also, the unreliable and open wireless channels with broadcast nature bring additional challenges to data security. The complexity of the IoT system further increases the above challenges [104].

Attackers in IoT networks can be classified into passive and active attackers according to the adversarial behaviour [32]. Active attackers influence nodes or the network traffic by injecting bogus packets, modifying packets, dropping packets and so on. Passive attackers eavesdrop on wireless communication and analyse traffic to obtain sensitive information, which do not alter the traffic or node behaviour. Passive and active attackers can cooperate in performing attacks. For example, passive attackers can launch a traffic analysis attack to reveal the sink location based on the fact that the sink node attracts traffic from all other nodes and traffic concentrates around the sink node, while active attackers can further launch DDoS and other active attacks on the sink node. Active attackers can be detected by intrusion detection algorithms. Passive attackers, on the other hand, are hard to be detected but can be prevented by encryption or anonymous mechanisms. Attackers can also be classified into internal and external attackers according to the access level [89]. Internal attackers get access to components of the target network, by physically capturing legitimate nodes or keys. Such attackers can decrypt packets, drop packets as intermediate nodes, and generate legitimate signatures for forged packets. External attackers cannot get access to components due to the lack of legal identities and keys. They can launch traffic jamming, eavesdropping, traffic analysis, DDoS attacks and other attacks.

The following summarizes the typical attacks on IoT networks from the bottom layer to the top [69]. The architecture of the IoT network has been introduced in Section 1.1.

- Attacks to End Devices: Adversaries physically capture and control nodes via node capture attacks. The secret information stored in captured nodes, such as keys and certificates, become visible to adversaries [11]. The adversaries can further utilize the captured information to pretend as legitimate nodes and perform other attacks, such as the data injection attack [130].

- Attacks to Communication Channels: Adversaries may eavesdrop on and interfere with transmitting channels, exploiting the broadcast nature of radio. If signals are not encrypted, the adversaries can readily obtain the information. Even if signals are encrypted, adversaries are still able to analyse the streams of signals and infer private information, such as the locations of the sources or destinations [79]. The adversaries can also interfere and even jam the wireless channels by sending noisy signals [83].

- Attacks to Network Protocols: By exploiting the vulnerabilities of network protocols, adversaries can launch Sybil attack, reply attack, man-in-middle, blackhole, wormhole attacks and so on [137]. For example, a sybil device impersonates several legitimate identities in IoT systems. Such attacks would compromise the efficiency and accuracy of voting mechanisms and multi-path routing protocols [137].

- Attacks to Sensory Data: IoT networks communicate by using ad hoc protocols, i.e., messages are transmitted hop-by-hop till reaching their destination. This provides adversaries opportunities to modify, inject or drop data. For example, an adversary can modify messages and forward them to other nodes as a forwarder, known as the pollution attack, which can be prevented by authentication algorithms. False data injection attack refers that adversaries send false data across the targeted network with legitimate identities [130]. Once the false data is accepted, IoT applications may return erroneous instructions or provide wrong services, compromising the reliability of IoT applications and networks. For example, the traffic congestion may aggravate if vehicles accept false road assistant messages. False data injection attacks can hardly be prevented by authentication algorithms.

- Denial of Service (DoS) Attack: The DoS attack represents a category of attacks, which exhaust resources and congest services of IoT systems [83]. For example, a sleep deprivation attack [82] is to break the programmed sleep routines and keep devices or nodes awake all the time until they are out of battery power supply. IoT devices have limited network and communication resources, and thus the DoS attacks can be catastrophic. Such attacks exhaust the limited energy of sensory nodes, reduce the network connectivity, paralyze the entire network, and reduce network lifetime [82].

- Software Attacks: Software attacks refer to a series of attacks which utilize backdoors of software to modify software and control operations [93]. Typical software attacks include malicious virus/worm/scripts [123, 93]. Intrusion detection system (IDS) and other traditional Internet security mechanisms are used to tackle the software attacks [124].

Data security is of paramount importance for IoT security. Security of data mostly refers to the protection of the confidentiality, integrity, non-repudiation, and availability of data.

- Confidentiality: it refers to the property that data is only visible to authorized users. The confidentiality is further extended to protections on private information, such as the transmission path, transmission time, and identities of senders/receivers. Encryption, anonymity, and access control mechanisms are utilized to guarantee confidentiality.

- Integrity: it confirms that packets in transmission have not be modified by intermediate nodes. The integrity of IoT data and devices, e.g., sensor readings and actuator commands, is the fundamental guarantee for securing IoT operations. Signatures, message authentication codes and hash values are applied to confirm integrity.

- Non-repudiation: Nodes cannot deny their actions of transmitted data. Non-repudiation is a security requirement of node reputation and incentive mechanisms [17]. The asymmetric cryptography offers possible solutions for preserving non-repudiation.

- Availability: it denotes that the authorized users can immediately get access to their required resources even in disastrous conditions [44]. In other words, the network should be robust to failures of nodes and links, which may be caused by congestions or attacks.

Effective mechanisms need to be designed to protect IoT networks for confidentiality, integrity, non-repudiation, and availability of information flows [20]. Cryptographic solutions have abilities to provide confidentiality, integrity, non-repudiation, and availability [57]. However, traditional cryptographic solutions proposed for the Internet are not suitable for IoT, due to the specific characteristics of IoT. Symmetric cryptographic solutions face security challenges from captured keys. Once a symmetric key is captured, data secured by that key is insecure as well. The pairwise cryptographic solution, which assigns a unique key for each pair of nodes, provides the highest level of security against capture attacks. However, the key memory of such a solution grows linearly to the size of the network [34]. In other words, it cannot be used in a large-scale IoT network. Furthermore, symmetric cryptographic solutions cannot guarantee non-repudiation. Asymmetric cryptographic solutions, on the contrary, can be applied to guarantee all the security properties, i.e., confidentiality, integrity, non-repudiation, and availability. As asymmetric cryptographic algorithms have higher computational complexity, they should be specifically designed to work in the resource-constrained IoT networks [50]. For example, public key infrastructure (PKI), a typical mechanism based on asymmetric keys, consumes storage, computational and communication resources to exchange public keys and their certificates among nodes. Also, the management and revocation of certificates is a complex job, especially in an IoT network with a large scale number of nodes [46]. In consequence, both symmetric and asymmetric cryptographic solutions should reduce extra communications to alleviate the traffic burden in IoT, due to the limited bandwidth in IoT networks.

2.2 Related Works

2.2.1 Source Location Privacy

Location privacy is a major concern in IoT security. In an event-triggered IoT network, once nodes sense a predefined event happening in their monitoring areas, they send messages about the event to a collecting node, e.g., the sink in WSN or RSU in VANET. As sensory nodes communicate in a wireless manner, they are vulnerable to many attacks including passive eavesdropping, bogus messages injection and pollution attacks [135]. In [55], the protection of source location privacy (SLP) was discussed for the first time. Adversaries could analyse the eavesdropped traffic to trace back to the source node, by performing content-correlation, time-correlation and size-correlation analysis on traffic [32].

According to the view of the network at passive eavesdroppers, they can be classified as global eavesdroppers and local eavesdroppers. Here, global eavesdroppers have capabilities to hear all traffic in the network, whilst local eavesdroppers just obtain traffic in a limited area. Routing based solutions [88, 7] were proposed to prevent local eavesdroppers. Ozturk et al. proposed a flooding protocol in [88], where each node diffuses data at a specific probability. However, the protocol provides location privacy against local adversaries at the sacrifice of transmission efficiency. The phantom routing scheme (PRS) was later proposed in [88]. The packet sent by the source node follows a random route until it arrives at a phantom source node, which will later flood packets towards the destination node. In [55], the authors proposed the phantom single-path routing scheme (PSRS), where the phantom source node uses single path routing instead of the flooding in PRS to route the message towards the destination node. A geographic routing protocol based on geographic information was proposed in [7]. It combines geographic routing together with other security techniques, such as encryption and trust management, to provide source location privacy in the network level. However, previous routing based solutions can only resist against local adversaries. In other words, these solutions cannot defend against global adversaries.

Dummy packets [108, 10] were then introduced to obfuscate the real traffic and prevent global adversaries from performing time-correlation analysis. ConstRate [108] introduces dummy traffic to achieve a constant sending rate at each node. If one node has no packets to send in a predetermined sending slot, it sends dummy packets instead. However, ConstRate incurs significant latency and energy consumption. To balance privacy and network performance, FitProRate proposed in [108] sends real or dummy packets at a fitted probabilistic rate. In [10], the notion of *interval indistinguish ability* was introduced as a fundamental property to further ensure source-location privacy. To provide indistinguishable intervals, a new kind of packets, fake packets, are randomly generated and sent in [10] in addition to real and dummy packets. All the above solutions generate a large amount of dummy traffic, which consumes network energy and degrades the network lifetime.

To tackle the burst of dummy packets, [133] and [118] were proposed to filter out dummy traffic in SLP solutions. In [133], a network is divided into cells and each cell selects a proxy to filter out dummy traffic in this cell. Each node sends real or dummy packets to the proxy in its cell and the proxy waits to send only real packets to the sink at predefined time slots. However proxies may become the bottleneck of the network. Inappropriate proxy selections or transmission rate may lead to high latency or packet-loss. A proxy selection algorithm, optimal filtering scheme(OFS), was proposed in [118] to maximize the network lifetime. However all these solutions cannot filter out dummy packets until they arrive at a specific proxy node.

Most dummy traffic based solutions [108, 10, 133, 118] assumed that adversaries cannot distinguish dummy packets from real packets based on the packet content. Network coding has the potential to guarantee such an assumption. In [43], it was proved that network coding based solutions have the potential to resist against content-correlation analysis due to its inherent coding/mixing nature. The homomorphic encryption is employed on Global Encoding Vectors (GEVs) to enhance both traffic untraceability and packet confidentiality in network coding based solutions. In [42], a specific kind of dummy packets satisfying *dummy nullity* was designed. Such dummy packets can be absorbed at intermediate nodes before arriving at the destination. [39] incorporated network coding and opportunistic routing to enhance the source-destination pairwise privacy.

Previous works were proposed to defend against passive attackers, while their resistance against active attackers is ignored. In fact, active attackers can insert or pollute packets to degrade network performance. Especially, it has been proved that network coding based solutions are vulnerable to pollution attacks [62]. Polluted packets can propagate further downstream and finally infect the entire network. Although there are detection and authentication mechanisms proposed to tackle pollution attacks [142], the concealed traffic patterns and content in location privacy preserving solutions degrade the detection and authentication performance [78]. The balance between location privacy and authentication remains an open question.

2.2.2 Secure Data Transmission

Cryptographic keys have been extensively employed to enforce confidentiality and integrity of data transmitted in decentralized wireless IoT networks. For example, IEEE 1609.2 specifies the process flow for the security processing services for secure data exchange, including signing and encrypting data. This standard supports the Elliptic Curve Digital Signature Algorithm (ECDSA) to generate signatures and the Advanced Encryption Standard (AES) as the symmetric encryption algorithm to encrypt data [4].

Sharing a common pair of keys between the communication partners is a prerequisite for establishing cryptographic communication. Key predistribution has been considered to share keys among neighbours in distributed IoT networks with stable topologies. A well-known Eschenauer-Gligor (EG) scheme, originally developed for encryption [38], predistributes symmetric keys and can potentially eliminate the need for real-time delivery of keys and certificates in authentication. Unfortunately, handshaking would be required between neighbouring nodes to decide on their common keys, resulting in excessive delay and communication overhead. This is unsuitable for decentralized wireless networks where communication overhead is a major concern. In [23], the authors proposed a q-composite key predistribution scheme based on the EG scheme, where only the nodes, predistributed no less than q keys in common, can communicate with each other using a new key obtained by hashing the common keys. In spite of improved security, the q-composite key predistribution would dramatically degrade connectivity. The authors also proposed to predistribute keys on a pairwise basis, i.e., every symmetric key is predistributed to a unique pair of devices. The EG scheme was later adopted to VANET in [9]. When a vehicle enters a new area, the Road Side Unit sends the shared keys of possible neighbours of the vehicle to the vehicle. In this way neighbours know the shared keys without handshaking in a stable topology. However, all the above EG based schemes require handshaking between a pair of nodes to decide on their common keys, each time the topology changes [38, 23, 9], resulting in excessive delay and communication overhead. The overhead further deteriorates in distributed networks such as the one specified by IEEE 802.11p [3], as collisions among uncoordinated transmissions that grow with the network density can increasingly defer key delivery or penalize the effective capacity of the networks [37]. For this reason, these schemes are unsuitable for decentralized mobile IoT networks with fast-changing topologies and intensive transmission collisions.

μTESLA is another popular protocol to authenticate messages using symmetric keys, where short MACs of ten to twenty bytes can be generated and verified using the same key [92]. Either a delayed publication of the key by the transmitter after the key expires, or a three-way negotiation via a trusted third-party was specified to make the key available at both the transmitter and receiver [92]. However, neither of these are suitable for mobile environments, where topologies keep changing and in most cases no trusted third party is available.

A straightforward approach to authentication in decentralized mobile networks is based on the trust secured by digital certificates [100] in a piggybacked manner, where every node uses a unique private key to sign its messages and sends the messages and signatures along with the corresponding public key and its certificate. The messages can be authenticated by validating the public key through the certificate and verifying the signatures through the public key. However, the real-time transmissions of public keys and certificates (around 100 bytes in total [64]) would substantially increase communication overhead, Additionally, the piggyback scheme [100] requires each node to maintain an up-to-date certificate revocation list, which is difficult in large scale networks.

Using the idea of piggyback [100], μTESLA was recently extended to mobile networks, named TSVC [70]. Apart from a message, a node transmits a MAC, the symmetric key generating the MAC, a signature signed on the symmetric key by using a unique private key of the node, the corresponding public key to verify the signature, and the digital certificate of the public key. The digital certificate is used to validate the legitimacy of the public key; the signature is to validate the symmetric key. After the transmission, the node aborts the symmetric key immediately (i.e., makes the key expire). It was also proposed in [70] that the public key, signature and certificate are only sent once until the topology changes, before which a chain of symmetric keys generated through one-way hash can be used and sent (together with MACs) in reverse order. The last symmetric key in the chain is used first and signed. Those generated earlier but used later can be verified by simply hashing and comparing with the one received earlier at a receiver. However, the communication overhead of TSVC is still quite high, especially in the case where the network topologies change frequently.

In [119], nodes in the same area form various logic groups and a virtual key tree model was proposed to provide keys within each group. When a node leaves the group, selected supernodes in the group generate and distribute a salt value to the whole group to start rekeying operations. However, a highly mobile topology requires frequent rekeying operations, leading to wireless channel congestion. Another approach is to use group signature [111], where a large number of private keys can match a single public key using bilinear mapping techniques. It can be straightforwardly used for authentication, provided every node has the public key and one of the corresponding private key. The approach has the merit of strong tolerance to topology changes, whereas group signature is computationally intensive, typically requires over 40 ms for authentication operations [21], and is unsuitable for real-time applications [94].

2.2.3 Analysis Model

Considering the distributed nature of IoT, unsuccessful authentication and encryption can be caused by either unmatched keys or transmission collisions. The complicated factors in failure secure data transmissions also make the security analysis of decentralized IoT challenging. Markov models were developed to evaluate IEEE 802.11a/e [16, 72], and later relay assisted mobile networks [85] and VANETs [121]. The models are superior in the sense that they can precisely model the protocol behaviors of individual nodes and rigorously infer the collision probabilities. However, none of the existing Markov models [16, 72, 85, 121] take security into account.

Analyses of different key predistribution schemes in terms of network connectivity were conducted in [131, 139, 140], exploiting random graph theory in static networks. However, these works [131, 139, 140] analysed the security of key distribution schemes based on simplified transmission channels. In [131], 1-connectivity was analysed for the EG scheme, where independent and identically distributed (i.i.d.) on/off channels were assumed between every pair of static nodes. A random intersection graph (e.g., random key graph) captures the key predistribution by placing an edge between any two nodes that share a key. An Erdős-Rényi (ER) graph captures static network topology by connecting any two nodes with an "on" channel. The analysis was later extended to the q-composite scheme [139], where a disk model was used to specify the transmission range of every node and modeled as a random geometric graph. The 1-connectivity was studied through the intersection of the random intersection graph and the ER graph [131] or random geometric graph [139]. A zero-one law was established with conditions identified to guarantee the networks to be asymptotically almost surely connected, as the number of nodes becomes large. Recently, this analysis has been applied to k-connectivity of the EG scheme [140]. However, the oversimplifications on transmission channels degrade the accuracy of the models.

2.3 Summary

This chapter provided a survey on IoT security. IoT security were analysed from the top layer to the bottom layer of IoT networks, followed by the introductions to IoT secure assurances and an overview of the cryptographic based solutions. Furthermore, related works on IoT location privacy, secure data transmission protocols, and analysis models were provided.

Chapter 3

Anti-Pollution Source-Location Privacy Protection in Multi-hop IoT Networks

3.1 Introduction

The threat to the source-location privacy is one of the critical security issues in IoT networks, such as wireless sensor networks and vehicular ad hoc networks, where adversaries may reversely trace along the traffic to the source node and further hunt targets around the source location. Source-location privacy solutions are required to defend against traffic analysis, including time-correlation, content-correlation and size-correlation analysis.

Dummy and fake packets were introduced in [80, 10] to eliminate time-correlation. However, the large amount of dummy traffic shortens the network lifetime. To counteract the dummy traffic, [133] and [118] divided the network into cells and designated proxies to filter out dummy packets for each cell. Nodes send packets to the corresponding proxy and the proxy forwards only real packets to the destination. However, these solutions cannot filter out dummy packets until they arrive at a specific proxy node. Network coding provides another solution to protect location privacy. In [43], it was proved that network coding based solutions have the potential to prevent content-correlation analysis due to its inherent coding/mixing nature. However, all the above solutions, especially the network coding based solutions, are vulnerable to pollution attacks [62]. Although authentication mechanisms were proposed to tackle pollution attacks [142], they should be specially designed in location-privacy preserving solutions as concealing traffic patterns and content in location-privacy preserving solutions conflicts with authentication mechanisms.

This chapter proposes an Anti-Pollution Source-Location Privacy scheme (AP-SLP) to filter out dummy and polluted packets during transmission. AP-SLP uses a triple-type homomorphic signature algorithm (TTHS) to verify signatures and recognize the corresponding packet type in terms of *Real*, *Dummy* and *Polluted*, without knowing the packet content. Opportunistic key predistribution schemes provide keys for TTHS. Each node is predistributed with two kinds of key rings to collaboratively provide privacy and integrity. The simulation results demonstrate that the proposed AP-SLP improves the message delivery rate and saves energy as compared with previous works.

The rest of the chapter is organized as follows. Section 3.2 presents network and attack models. Section 3.3 introduces the proposed scheme AP-SLP from two aspects, the opportunistic key predistribution schemes and the triple-type homomorphic signature algorithm. The security analysis and simulation results are given in Section 3.4 and Section 3.5, respectively, followed by the summary in Section 3.6.

3.2 Network Model

3.2.1 Target Network and Attack Model

This chapter is interested in a typical event-triggered IoT network, which is normally deployed for tracking and monitoring applications. It consists of one sink node and a large number of sensory nodes. Sensory nodes are widely deployed into an area to inspect a certain phenomenon. Once the certain phenomenon appears, the IoT node deployed around transmits a packet as the source node to inform the sink about the event. Here the sink is always trustful and powerful, while IoT sensory nodes are energy limited and vulnerable to attacks. Adversaries aim at revealing the source location and degrading network performance without being detected. Both external and internal adversaries coexist in the network. They perform active and passive attacks to achieve their targets.

External eavesdroppers have the ability to overhear the network traffic. If the transmitted packets are not encrypted, eavesdroppers can inspect events, including the source location, directly from the packets. If the transmitted packets are encrypted, external eavesdroppers use the hop-by-hop-trace attack to deduce the forwarding path and finally trace back to the source location. These attackers correlate outgoing with incoming packets at intermediate nodes in terms of packets content, size, and time. It is also assumed that internal adversaries, which capture some legal nodes and obtain secret keys held by those compromised nodes, can decode packets with captured keys.

Active adversaries perform pollution attacks, i.e., insert false packets or modify transmitted packets. These malicious packets inserted or modified by adversaries are named as polluted packets. Pollution attack severely degrades network performance, especially in network coding based networks [29]. The propagation of polluted packets not only wastes the network energy and bandwidth but also reduces the decode success rate in network coding based networks.

3.2.2 Network Coding based Transmission Model

To prevent adversaries from tracing back to the source location by performing time-correlation analysis, all nodes use slotted transmission and send packets in the predetermined time slots [10] with anonymous source addresses. If a node has no real packets $\hat{\mathbf{o}}_r$ to transmit, it generates dummy packets $\hat{\mathbf{o}}_d$ with the same size instead. The set of packets, $\{\hat{\mathbf{o}}_r\}$ or $\{\hat{\mathbf{o}}_d\}$, generated by the source at a particular time slot, is gathered as a generation of packets [95]. The type of a single generation, T_g, is "dummy" or "real". Here, real packets contain meaningful information about real events, while dummy packets have meaningless content but just to obfuscate the

Table 3.1 : Notation Interpretation

Notation	Explanation
n	the number of original packets
m	the dimension of original packets
$\mathbf{p}_i$	network coding packets
α_i/β_i	Global/Local Encoding Vector
PK	Public key used to verify packets type
$SK := \alpha$	Secret key used to sign real packets
Π_V	End-to-end asymmetric key pool
Π_{EK}	Hop-by-hop encryption key pool
T_g	Type of the generation
id_g	Generation identity
id_V	End-to-end signature key identity
id_E	Hop-by-hop encryption key identity

real traffic. Tab. 3.1 summarizes the notations used in this chapter.

To counter content-correlation analysis, we perform random network coding [54] on n original packets $\{\hat{\mathbf{o}}_i\}_{i=1}^{n} \in \mathbb{F}_q^m$, which are within a single generation. $\hat{\mathbf{o}}_i$ is the generalized denotation of $\hat{\mathbf{o}}_r$ and $\hat{\mathbf{o}}_d$. Each generation has a unique generation identity, id_g. The source node forms a $(n+m)$-dimensional packet, $\mathbf{o}_i = (\mathbf{e}_i, \hat{\mathbf{o}}) \in \mathbb{F}_q^{m+n}$, by augmenting each m-dimensional vector $\hat{\mathbf{o}}_i$ with a n-dimensional unit vector $\mathbf{e}_i \in \mathbb{F}_q^n$. Then the source sends out packets $\{\mathbf{p}_i\}$, which are random linear combinations of $\{\mathbf{o}_i\}$. When an intermediate node $\mathbf{M}$ receives packets $\{\mathbf{p}_i\}$ from the upstream neighbour, the node $\mathbf{M}$ performs random network coding on $\{\mathbf{p}_i\}$ with randomly chosen *Local Encoded Vectors* (LEV) $\beta_i \in \mathbb{F}_q$ to generate a new packet $\mathbf{p} = \sum \beta_i \mathbf{p}_i$. $\mathbf{M}$ further forwards encoded packets $\{\mathbf{p}\}$ to its downstream nodes until reaching the destination.

If the encoded packets sent in the network are not modified by adversaries, these packets are linear combinations of the original packets $\{\mathbf{o}_i\}$ in the following form,

$$\begin{aligned}
\mathbf{p}_i &= (p_{i,1}, \cdots, p_{i,n}, p_{i,n+1}, \cdots, p_{i,m+n}) \\
&= (\underbrace{-\alpha_i-}_{n}, \underbrace{-\hat{p}_i-}_{m}) \in \mathbb{F}_q^{m+n}, \\
\alpha_i &= (p_{i,1}, \cdots, p_{i,n}) \in \mathbb{F}_q^{n}, \\
\hat{p}_i &= (p_{i,n+1}, \cdots, p_{i,m+n}) \in \mathbb{F}_q^{m},
\end{aligned}$$

where the first m symbols of $\mathbf{p}_i$ compose the *Global Encoding Vector* (GEV), denoted as α_i. Once the destination receives n linear independent and unmodified packets, $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$, it decodes and obtains original packets, $\{\hat{\mathbf{o}}_i\}$, by performing Gaussian elimination on $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$. In [54], it is proved that if $\mathbb{F}_q$ is sufficiently large, any n encoded packets are linear independent and can be decoded at a high probability.

3.3 Proposed Anti-Pollution Source-Location Privacy Preserving Scheme

This section presents the proposed anti-pollution source-location privacy preserving scheme (AP-SLP), as shown in Fig. 3.1. In AP-SLP, nodes send messages in predetermined time slots to prevent time-correlation analysis. To be specific, if nodes have no real messages to send in predetermined time slots, nodes will send dummy messages instead. Nodes also perform random network coding on packets to prevent content- and size-correlation analysis, as explained in Section 3.2.

As the transmission of dummy and polluted packets costs bandwidth and energy, a triple-type homomorphic signature (TTHS) algorithm is proposed in AP-SLP to filter out dummy and polluted packets at the intermediate nodes during transmis-

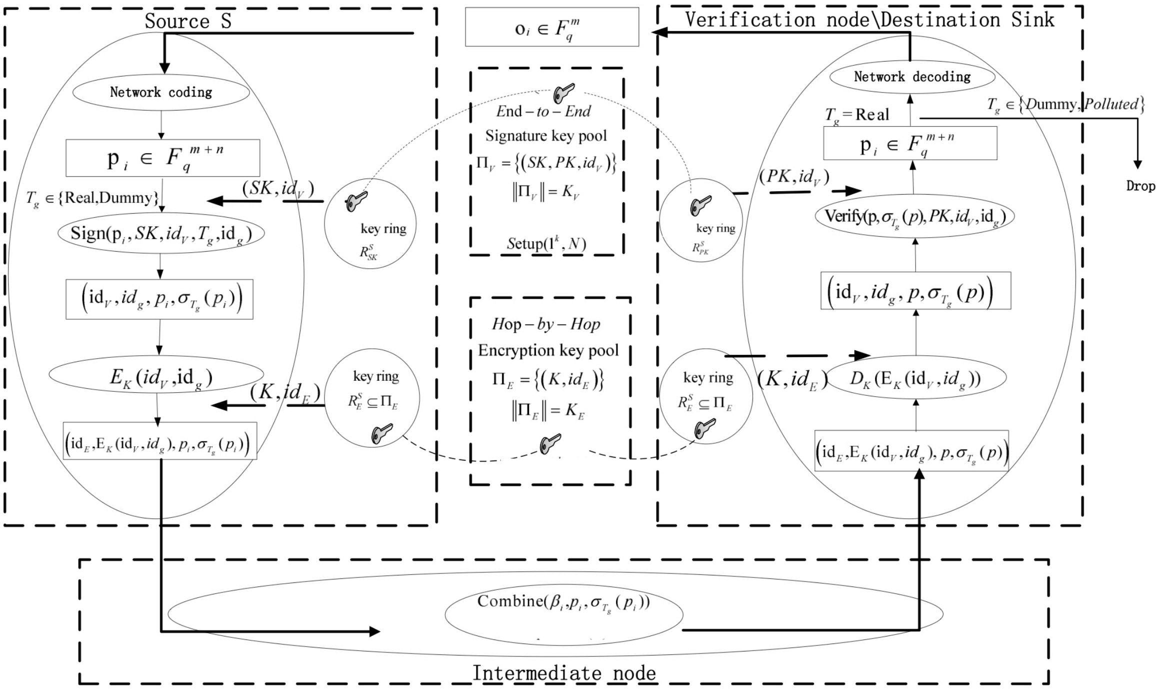


Figure 3.1 : Flowchart of proposed Anti-Pollution Source-Location Privacy Preserving scheme (AP-SLP)

sion. Details will be given in Section 3.3.2. The source node produces a homomorphic signature according to the packet type with a private key. Intermediate nodes with the corresponding public key are able to verify the packet integrity and recognize the packet type without decrypting the packet content. Only real packets are further forwarded to the destination, while dummy and polluted packets are filtered out to reduce transmission cost.

The aforementioned triple-type homomorphic signature algorithm requires that asymmetric keys for authentication are preloaded in sensory nodes prior to deployment. It is impractical to preload a single master key at all nodes due to its vulnerability to key capture, or assign each pair of nodes with a unique pairwise key due to the demand of huge key memory [24]. In AP-SLP, opportunistic key predistribution schemes are proposed to ensure that intermediate nodes can verify signatures with certain possibilities. Each node is preloaded with two kinds of key rings. One is the asymmetric key ring, consisting of a private key ring and a public key ring. All these asymmetric keys are randomly chosen from an asymmetric key pool, which is shared by the whole network, as shown in Fig. 3.1. The TTHS algorithm uses private keys to sign packets. Signatures and key identities of the used private keys are attached to packets before transmissions. Intermediate nodes which store the corresponding public keys in their public key rings can verify packets. Each node also preloads an encryption key ring to encrypt asymmetric key identities in packets. The ciphertexts of asymmetric key identities change hop-by-hop to avoid content-correlation on the asymmetric key identities. Details will be given in Section 3.3.1.

3.3.1 Key Predistribution Mechanism

This section illustrates two opportunistic key predistribution schemes used in AP-SLP. They cooperate to guarantee that an intermediate node has a possibility to verify the signature while adversaries cannot correlate key identities.

End-to-End Signature Key Predistribution

The system generates a asymmetric key pool $\Pi_V = \{(SK, PK, id_V)\}_{K_V}$, which comprise K_V pairs of end-to-end asymmetric keys. Each pair of asymmetric keys has a unique asymmetric key identity id_V. The sink uses Π_V to verify received packets. For each sensory node $\mathbf{N}$, k_{SK} private keys are randomly chosen from Π_V to form a private key ring, $R^{\mathbf{N}}_{SK} = \{(SK, id_V)\}_{k_{SK}}$, which is preloaded in the node before deployment. When the node $\mathbf{N}$ is to send a new generation of packets, it randomly chooses a private key from its private key ring, $(SK, id_V) \in R^{\mathbf{N}}_{SK}$, to sign packets in this generation. id_V is also attached to packets.

k_{PK} public keys are also randomly chosen from Π_V to form a public key ring, $R^{\mathbf{N}}_{PK} = \{(PK, id_V)\}_{k_{PK}}$, for each node. $R^{\mathbf{N}}_{PK}$ is preloaded in the node to verify signatures and recognize the type of received packets. When $\mathbf{N}$ receives a packet attached with id_V, $\mathbf{N}$ checks whether it has the corresponding public key in its public key ring, i.e., $(PK, id_V) \in R^{\mathbf{N}}_{PK}$. If so, $\mathbf{N}$ verifies the signature with the corresponding public key, forwards real packets, and drops meaningless packets (dummy and polluted packets). In this way, meaningless packets can be prevented from further propagating in the network. The possibility that the node N possesses (PK, id_V) can be given by,

$$p_V = \frac{\binom{K_V-1}{k_{PK}-1}}{\binom{K_V}{k_{PK}}} = \frac{k_{PK}}{K_V}. \tag{3.1}$$

Hop-by-Hop Encryption Key Predistribution

The asymmetric key identity and generation identity are attached to packets. However, they are unchanged during the transmission. To prevent adversaries from performing content-correlation analysis on asymmetric key identities and generation identities, AP-SLP encrypts asymmetric key identities and generation identities hop-by-hop.

The hop-by-hop encryption key pool $\Pi_{EK} = \{(K, id_E)\}_{K_E}$ consists of K_E symmetric keys, each of which has a unique identity id_E. k_E symmetric keys are randomly chosen from Π_{EK} to form an encryption key ring, $R^{\mathbf{N}}_E \subset \Pi_{EK}$, for each node. Before a node $\mathbf{N}$ sends or forwards packets attached with (id_V, id_g), $\mathbf{N}$ randomly chooses a symmetric key $(K, id_E) \in R^{\mathbf{N}}_E$ and encrypts (id_V, id_g) to $\mathbb{E}_K(id_V, id_g)$ by using (K, id_E), where $\mathbb{E}$ is a light-weight symmetric encryption algorithm. Then $\big(id_E, \mathbb{E}_K(id_V, id_g)\big)$ is attached to packets instead of (id_V, id_g). When the node $\mathbf{N}$ receives a packet attached with $\big(id_E, \mathbb{E}_K(id_V, id_g)\big)$ from the upstream neighbour, $\mathbf{N}$ decodes $\mathbb{E}_K(id_V, id_g)$ to get (id_V, id_g) as follows if $\mathbf{N}$ has (K, id_E) in its encryption key ring $R^{\mathbf{N}}_E$.

$$(id_V, id_g) = \mathbb{D}_K\big(\mathbb{E}_K(id_V, id_g)\big),$$

where $\mathbb{D}_K$ denotes the corresponding decryption algorithm of $\mathbb{E}$ by using key K.

p_E, which denotes the possibility that a node has (id_E, K) in its encryption key ring, can be given by,

$$p_E = \frac{\binom{K_E-1}{k_E-1}}{\binom{K_E}{k_E}} = \frac{k_E}{K_E}. \tag{3.2}$$

Then **N** checks whether it possess (PK, id_V) in its public key ring as aforementioned. (PK, id_V) is finally used to verify packets. Therefore, a node **N** can verify a specific packet if and only if **N** has both the hop-by-hop symmetric key in its encryption key ring and the end-to-end public key in its public key ring. The possibility can be given by,

$$p_S = p_V \times p_E. \tag{3.3}$$

3.3.2 Homomorphic Signature Algorithm

This section introduces the proposed triple-type homomorphic signature algorithm, which filters out meaningless packets without exposing privacy in a network coding based network. Intermediate nodes can distinguish three types of packets, i.e., real packets, dummy packets and polluted packets, by using keys predistributed in Section 3.3.1.

Key Setup: Setup$(1^k, N)$

Given a security parameter 1^k and a positive integer N, the system generates a bilinear group tuple $\Im = (\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_T, e, \varphi)$ [18], where cyclic groups $\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_T$ have the same prime order $q > 2^k$, and $e : \mathbb{C}_1 \times \mathbb{C}_2 \to \mathbb{C}_T$ satisfies bilinearity and non-degeneracy. Here bilinearity denotes that for any $g \in \mathbb{C}_1, h \in \mathbb{C}_2$, and $a, b \in Z$, $e(g^a, h^b) = e(g, h)^{ab}$, and non-degeneracy denotes that if g generates $\mathbb{C}_1$ and h generates $\mathbb{C}_2$, then $e(g, h)$ generates $\mathbb{C}_T$ [18]. $\varphi : \mathbb{C}_2 \to \mathbb{C}_1$ is an efficiently computable isomorphism. Asymmetric keys are set up as given in Algorithm 1.

Each pair of asymmetric keys (SK, PK) is assigned with a unique asymmetric key identity id_V. Asymmetric keys are used to sign and verify packets. There are

Algorithm 1: Setup$(1^k, N)$[18]

```
Input: $1^k$,$N$
Output: $SK, PK$
1: The private key $SK := \alpha$ is randomly chosen from $\mathbb{F}_q$.
2: The corresponding public key $PK := (\Im, H, \{g_i\}_{i=1}^N, h, \mu)$ satisfies following conditions:
   $\Im = (\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_T, e, \varphi)$.
   $H : \mathbb{Z} \times \mathbb{Z} \to \mathbb{C}_1$ is a collision-resistant cryptographic hash function.
   Randomly choose generators $g_1, g_2, ..., g_N$ of $\mathbb{C}_1$ and generator $h$ of $\mathbb{C}_2$.
   $\mu = h^\alpha$, here $\alpha$ is the private key.
```

also symmetric keys in networks to encrypt asymmetric key identities hop-by-hop. The generation of symmetric keys depends on the encryption algorithm, which is not novelly designed in this chapter. Readers can refer to [109] for details. Keys are distributed among nodes as present in Section 3.3.1.

Source Signs: Sign(*$\boldsymbol{p}, SK^S, id_V, T_g, id_g$*)

When the source is to send an network coding based packet $\mathbf{p}$ (explained in Section 3.2.2), it randomly chooses a private key (SK^S, id_V^S) from its private key ring to sign $\mathbf{p}$. As $SK^S = \alpha \in \mathbb{F}_q$ is an element from the finite field $\mathbb{F}_q$, there exists $-\alpha$ satisfying $\alpha + (-\alpha) = 0$ for any α. Given (SK^S, id_V^S), $SK^S = \alpha$ is used if the packet type is real, else $-\alpha$ is used instead to sign the dummy packet $\mathbf{p}$. The signature $\sigma(\mathbf{p})$ is given by Algorithm 2.

Send Packet: Send$\{id_E||\mathbb{E}_K(id_V^S, id_g)||\boldsymbol{p}||\sigma(\boldsymbol{p})\}$

Intermediate and the destination nodes need the key identity id_V^S and the generation identity id_g to verify signatures, hence the source attaches id_V^S and id_g to the

Algorithm 2: Sign$(\mathbf{p}, SK^S, id_V, T_g, id_g)$

```
Input:
    Packet $\mathbf{p} = (p_1, p_2, \ldots, p_n, p_{n+1}, \ldots, p_{n+m})$,
    $SK^S := \alpha$, Generation type $T_g^S$, Generation identity $id_g$.
Output: Signature $\sigma(\mathbf{p})$
if $T_g^S = Real$, then
    $\sigma(\mathbf{p}) = \sigma_R(\mathbf{p}) = \left(\prod_{i=1}^{m} H(id_g, i)^{p_{n+i}} \prod_{j=1}^{n} g_j^{p_j}\right)^{\alpha}$
else if $T_g^S = Dummy$, then
    $\sigma(\mathbf{p}) = \sigma_D(\mathbf{p}) = \left(\prod_{i=1}^{m} H(id_g, i)^{p_{n+i}} \prod_{j=1}^{n} g_j^{p_j}\right)^{-\alpha}$
end if
```

packet and the signature $\{\mathbf{p}, \sigma(\mathbf{p})\}$. However, the key identity and the generation identity are not changed during the transmission if they are transmitted in plaintext. Adversaries can trace back to the source location by following packets with the same key identity and the generation identity. To conceal id_V^S and id_g, the source encodes id_V^S and id_g with a symmetric key (K, id_E) randomly chosen from its encryption key ring. Here K is the symmetric key and id_E is its unique key identity. id_E and the ciphertext of (id_V^S, id_g) are attached to the packet instead of (id_V^S, id_g). id_E does not reveal the packet trajectory, because the symmetric key is changed hop-by-hop during the transmission.

Finally, the source sends out packets in the form of $\{id_E||\mathbb{E}_K(id_V^S, id_g)||\mathbf{p}||\sigma(\mathbf{p})\}$, as given by Algorithm 3. Here $||$ means attaching. $\mathbf{p}$ is the packet to be sent. $\sigma(\mathbf{p})$ is the signature of $\mathbf{p}$ given by Algorithm 2. id_V^S is the key identity of the used asymmetric key. $\mathbb{E}_K(id_V^S, id_g)$ is the ciphertext of (id_V^S, id_g) using the symmetric key K. id_E is the key identity of K.

```
Algorithm 3: Send($\mathbf{p}, \sigma(\mathbf{p}), id_V^S, id_g, R_E$)
Input:
    Packet $\mathbf{p}$, Signature $\sigma(\mathbf{p})$,
    Asymmetric key identity $id_V^S$, Generation identity $id_g$,
    Symmetric key ring $R_E$
Output:    $\{id_E||\mathbb{E}_K(id_V^S, id_g)||\mathbf{p}||\sigma(\mathbf{p})\}$
    Randomly choose key $(K, id_E) \in R_E$
    $E_K(id_V^S, id_g)$: Encode $(id_V^S, id_g)$ with $K$
    Send $\{id_E||\mathbb{E}_K(id_V^S, id_g)||\mathbf{p}||\sigma(\mathbf{p})\}$
```

Verification: Verify$(PK^S, \boldsymbol{p}, \sigma(\boldsymbol{p}))$

Upon receiving a packet $\{id_E||\mathbb{E}_K(id_V^S, id_g)||\mathbf{p}||\sigma(\mathbf{p})\}$ from the upstream neighbour, a node first searches the key K with identity id_E in its encryption key ring. If it has K, the node decodes $\mathbb{E}_K(id_V^S, id_g)$ with the key K. After that, the node searches the public key PK^S with the identity id_V^S in its public key ring. If it has PK^S, it outputs the generation type T_g with Algorithm 4.

The correctness of real packet signatures has been proved in [18]. Here, we prove that dummy packet signatures are verified correctly in Algorithm 4. According to Algorithm 2, the valid signature of a dummy packet $\mathbf{p} \in \Pi_D$ can be given as $\sigma_D(\mathbf{p}) = \delta^{-\alpha}$. Note that the Setup phase requires that e satisfies bilinearity properties, which means that $e(g^a, h^b) = e(g, h)^{ab} \in \mathbb{C}_T, \forall g \in \mathbb{C}_1, h \in \mathbb{C}_2, a, b \in \mathbb{F}_q$. Therefore we have

$$\gamma_1(PK^S, \sigma(\mathbf{p})) = e(\sigma_D(\mathbf{p}), h) = e(\delta^{-\alpha}, h) = [e(\delta, h)]^{-\alpha},$$

$$\gamma_2(PK^S, id_g, m, \mathbf{p}) = e(\delta, \mu) = e(\delta, h^{\alpha}) = [e(\delta, h)]^{\alpha}.$$

$\gamma_1(PK^S, \sigma(\mathbf{p})) \cdot \gamma_2(PK^S, id_g, m, \mathbf{p}) = 1$ establishes for dummy packet signatures, as given in Algorithm 4.

Only packets with $T_g = Real$, i.e., real packets, are kept in intermediate nodes for

Algorithm 4: Verify$(PK^S, \mathbf{p}, \sigma(\mathbf{p}))$

```
Input:
    $\mathbf{p} = (p_1, p_2, \ldots, p_n, p_{n+1}, \ldots, p_{n+m})$, $\sigma(\mathbf{p})$, $PK^S$
Output:   $T_g$
Compute $\gamma_1$ and $\gamma_2$
    $\gamma_1(PK^S, \sigma(\mathbf{p})) = e(\sigma(\mathbf{p}), h)$
    $\gamma_2(PK^S, id_g, m, \mathbf{p}) = e(\delta, \mu)$                    (3.4)
    $here, \delta := \prod_{i=1}^{m} H(id_g, i)^{p_{n+i}} \prod_{j=1}^{n} g_j^{p_j}$
if $\gamma_1(PK^S, \sigma(\mathbf{p})) = \gamma_2(PK^S, id_g, m, \mathbf{p})$ then
    $T_g = Real$
else if $\gamma_1(PK^S, \sigma(\mathbf{p})) \cdot \gamma_2(PK^S, id_g, m, \mathbf{p}) = 1$ then
    $T_g = Dummy$
else
    $T_g = Polluted$
end if
```

further combination, or in the destination/sink for further decoding [54]. Dummy and polluted packets are dropped once detected. Here the type “Polluted” denotes that the packet is not the linear combination of the packets sent by the source, i.e., the packet is modified or inserted by pollution attackers.

Combination: Combination$\{\boldsymbol{p}_i,\sigma(\boldsymbol{p}_i)\}$

After an intermediate node verifies the packet type and keeps the real packets $\{\mathbf{p}_i,\sigma(\mathbf{p}_i)\}$, it performs random network coding on $\{\mathbf{p}_i,\sigma(\mathbf{p}_i)\}$ to generate a new packet $\mathbf{p} = \sum \beta_i \mathbf{p}_i$. Here β_i is a randomly selected *LEV*. The intermediate node also generates the valid signature $\sigma(\mathbf{p})$ for $\mathbf{p}$ based on $\{\sigma(\mathbf{p}_i)\}$, as given in Algorithm 5. Then the intermediate node randomly chooses a symmetric key (K, id_E) from its encryption key ring and forwards a packet in the form of $\{id_E||\mathbb{E}_K(id_V^S, id_g)||\mathbf{p}||\sigma(\mathbf{p})\}$, as given in Algorithm 3.

Algorithm 5: Combination$\{\mathbf{p}_i,\sigma(\mathbf{p}_i)\}$

```
Input:
    $\mathbf{p}_i,\sigma(\mathbf{p}_i)$
Output:  $\mathbf{p}$, $\sigma(\mathbf{p})$
Randomly choose local encoded vector (LEV) $\beta_i \in \mathbb{F}_q$
Perform linear network coding on $\mathbf{p}_i$ with $\beta_i$:
    $\mathbf{p} = \sum \beta_i \mathbf{p}_i$
Generate signature for $\mathbf{p}$:
    $\sigma(\mathbf{p}) = \prod \sigma(\mathbf{p}_i)^{\beta_i}$    (3.5)
```

Proposition 1: The combination algorithm in Algorithm 5 outputs a valid signature, i.e., the signature algorithm is homomorphic.

Proof 3.1: Given a set of packets and valid signatures $(\mathbf{p}_i, \sigma(\mathbf{p}_i))$, $\mathbf{p} = \sum \beta_i \mathbf{p}_i$, the random linear combination of $\mathbf{p}_i$ with LEV β_i, can be given by,

$$\begin{aligned}\mathbf{p} =& \sum \beta_i \mathbf{p}_i \\ =& \sum \beta_i \cdot (p_{i,1}, \cdots, p_{i,n}, p_{i,n+1}, \cdots, p_{i,n+m}) \\ =& (\sum \beta_i p_{i,1}, \cdots, \sum \beta_i p_{i,n}, \sum \beta_i p_{i,n+1}, \cdots, \sum \beta_i p_{i,n+m}),\end{aligned} \tag{3.6}$$

where $\mathbf{p}_i = (p_{i,1}, \cdots, p_{i,n}, p_{i,n+1}, \cdots, p_{i,n+m})$ is a $(n+m)$-dimension vector, as explained in 3.2.2. The combination correctness in the case of real packets is proved in [18]. We only prove the correctness in the case of dummy packets, i.e., $\mathbf{p}_i$ are dummy packets. Here, $\mathbf{p}$, which is the linear combination of $\mathbf{p}_i$, is also a dummy packet.

$\prod \sigma(\mathbf{p}_i)^{\beta_i}$ is the signature of $\mathbf{p}$ given by Algorithm 5. It can be further derived as follows,

$$\begin{aligned}\prod_i \sigma(\mathbf{p}_i)^{\beta_i} &= \prod_i \Big[\big(\prod_{k=1}^{m} H(id_g, k)^{p_{i,k+n}} \prod_{j=1}^{n} g_j^{p_{i,j}}\big)^{-\alpha}\Big]^{\beta_i} \\ &= \Big[\prod_i \prod_{k=1}^{m} H(id_g, k)^{\beta_i p_{i,k+n}} \cdot \prod_i \prod_{j=1}^{n} g_j^{\beta_i p_{i,j}}\Big]^{-\alpha} \\ &= \Big[\prod_{k=1}^{m} \big(H(id_g, k)\big)^{\sum_i \beta_i p_{i,k+n}} \prod_{j=1}^{n} (g_j)^{\sum_i \beta_i p_{i,j}}\Big]^{-\alpha} \\ &= \sigma(\mathbf{p})\end{aligned}$$

where the first equality is obtained by substituting $\sigma(\mathbf{p}_i)$ in Algorithm 2 into the right-hand side (RHS) of the equality. The last equality is based on (3.6) and Algorithm 2, which defines the valid signature of the dummy packet $\mathbf{p}$. So (3.5) establishes for dummy and real packets. This concludes the proof.

3.4 Security Analysis

This section provides security analysis on the proposed AP-SLP scheme in terms of protecting source-location privacy against traffic analysis, including time-correlation analysis, content-correlation analysis, and size-correlation analysis.

Time-correlation analysis. In the proposed AP-SLP scheme, a node sends real packets, or generates and sends dummy packets, in predetermined time slots to eliminate time-correlation at the node, provided that adversaries are unable to distinguish dummy packets from real packets based on the packet content [32].

There are two kinds of adversaries in the network.

One kind of adversaries are external ones which do not have any keys. As AP-SLP performs linear network coding on packets before transmitting, adversaries need to collect sufficient packets from the same source and the same generation to decode packets. However the source uses anonymous node identity and the generation identity is encrypted hop-by-hop. The possibility that adversaries can collect enough packets in a same generation for network decoding is negligible, as proved in [43]. In other words, external adversaries can hardly distinguish packet types from packet content.

The other adversaries are internal ones which capture some asymmetric and symmetric keys. In the case that internal adversaries capture both asymmetric and symmetric keys used in the packet, adversaries can verify the type of the packet based on the signature. Even if adversaries recognize the real packet and obtain the end-to-end asymmetric key identity, they can not recognize the real source based on the end-to-end asymmetric key identity. It is because AP-SLP uses opportunistic key predistribution schemes (see Section 3.3.1), where an asymmetric key identity maps to an asymmetric key and the asymmetric key can be predistributed to a large number of nodes.

Content-correlation analysis. Adversaries perform content-correlation analysis on two types of content in the proposed AP-SLP.

One is the traditional packet content. Network coding inherently mixes packets at each hop, which guarantees that correlating content of network-coded packets is computationally impossible, as proved in [43].

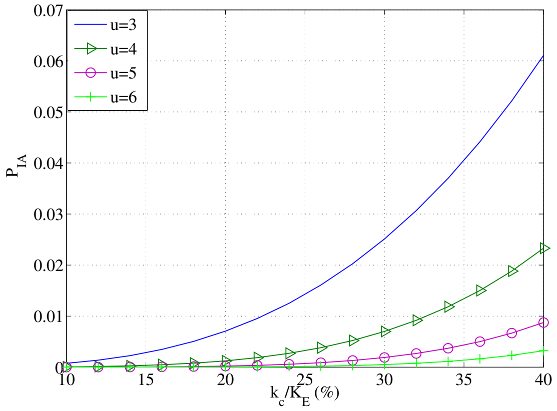


Figure 3.2 : The possibility that adversaries trace back to the source location based on asymmetric key identities.

The asymmetric key identity and the generation identity in the proposed scheme also reveal content-correlation in transmissions. The asymmetric key and generation identity are encrypted by symmetric keys and change hop-by-hop in content. It is impossible for external adversaries to decrypt the asymmetric key and generation identity, or directly perform content-correlation analysis on encrypted identities. Internal adversaries can decrypt identities only if the used symmetric keys are captured. In other words, the possibility that adversaries trace back to the source based on asymmetric key and generation identities is equal to the possibility that adversaries capture all symmetric keys used along the path from the source node to the destination. The possibility, denoted by $p_{\mathbf{IA}}$, can be given by,

$$p_{\mathbf{IA}} = \frac{\binom{K_E-u}{k_c-u}}{\binom{K_E}{k_c}}, \tag{3.7}$$

where u denotes the number of distinct symmetric keys used along the path from the source node to the destination. k_c is the number of captured symmetric keys. Fig. 3.2 plots the possibility that adversaries can trace back to the source based on identities. The x-axis is the percentage of captured symmetric keys. The size of the symmetric key pool is set to 100. The figure shows that $p_{\mathbf{IA}}$ decreases when the number of used symmetric keys u grows or the number of captured symmetric keys k_c decreases. The possibility $p_{\mathbf{IA}}$ is negligible, i.e., less than 10^{-3}, when u is larger than 5 and less than 16% symmetric keys are captured. The percentage of captured symmetric keys depends on the number of captured nodes and the size of encryption key ring. The more keys are stored in the encryption key ring per node, the more keys are captured with a given number of captured nodes. Therefore, the size of the encryption key ring should be kept small to enhance the location privacy.

Size-correlation analysis. Random network coding trims every packet to the same size, therefore the size-correlation is eliminated.

3.5 Performance Evaluation

This section evaluates the performance of the proposed AP-SLP scheme in terms of message delivery ratio and energy cost. ConstRate [80] and NC-ConstRate are compared as the baseline. NC-ConstRate integrates network coding with ConstRate [80]. Simulations were performed using Network Simulator NS2. In simulations, 20 sensory nodes with a transmission radius of 100 meters are randomly distributed in an area of $1100m \times 300m$. The sending time slots are predetermined as in ConstRate [80]. Tab. 3.2 provides simulation parameters. Each set of parameters runs for 100 times. The average performance results of the proposed AP-SLP, as well as ConstRate and NC-ConstRate, are plotted.

Fig. 3.3 compares the message delivery ratio of ConstRate, NC-ConstRate, and the proposed AP-SLP, where internal attackers launch pollution attacks. The x-axis denotes the percentage of internal attackers in the network. In the case that no internal attackers exist in the network, AP-SLP provides the highest delivery ratio. It is because only real packets are further forwarded in AP-SLP, saving bandwidth and energy to improve the transmission efficiency of real packets. When the per-

Table 3.2 : Notation Interpretation

Notation	Value
Simulation network area	$1100m \times 200m$
Number of nodes	20
Captured node ratio	$[0, 0.3]$
Transmission range	100m
Verify possibility	$p_{\mathbf{S}} = 0.1, 0.2, 0.3$
Receiving/Sending energy	0.6
Computing energy	0.01

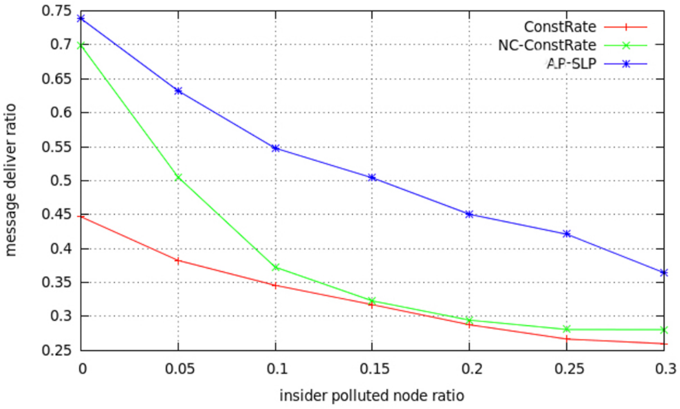


Figure 3.3 : Comparison of the message delivery ratio

centage of internal attackers increases, delivery ratios of all three solutions decrease. NC-ConstRate and AP-SLP decrease more quickly than ConstRate. It is because network coding used in NC-ConstRate and AP-SLP is especially sensitive to pollution attacks. However, the proposed AP-SLP still obtains the highest delivery ratio, as it prevents the propagation of polluted and dummy packets by filtering out these packets during transmissions.

Fig. 3.4 presents how the message delivery ratio of AP-SLP varies with the verification key possibility $p_{\mathbf{S}}$. $p_{\mathbf{S}}$, defined in (3.3), ranges from 0.1 to 0.3. When $p_{\mathbf{S}}$ grows, message delivery ratio increases as well. It is because more intermediate nodes have keys to verify packet types and meaningless packets are dropped much earlier.

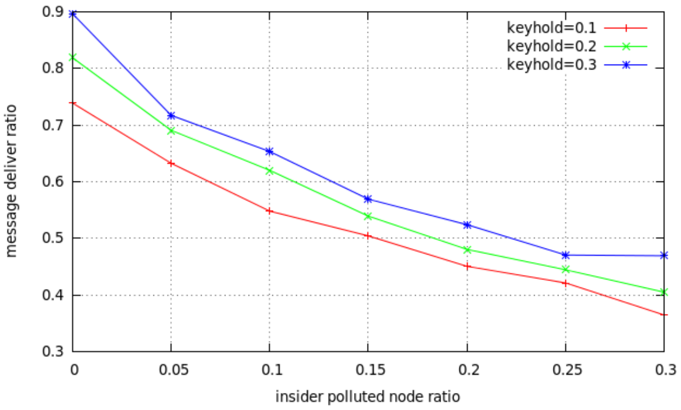


Figure 3.4 : Message delivery ratio of AP-SLP

Fig. 3.5 shows how AP-SLP saves energy with different $p_{\mathbf{S}}$. Energy related parameters are given in Tab. 3.2. It is shown that NC-ConstRate consumes more energy than AP-SLP. It is because NC-ConstRate transmits meaningless packets and decodes polluted packets, which all consume energy. In contrast, AP-SLP filters out dummy and polluted packets in transmission. Another observation is that the gap between the consumed energy of NC-ConstRate and AP-SLP enlarges when the insider polluted nodes ratio grows. It is also due to the recognition of polluted packets in the proposed AP-SLP.

Fig. 3.5 also shows that a growing $p_{\mathbf{S}}$ contributes to saving energy. It is because when $p_{\mathbf{S}}$ increases, more polluted packets are detected and more transmission energy is saved. In other words, the network gains a better message delivery ratio and saves

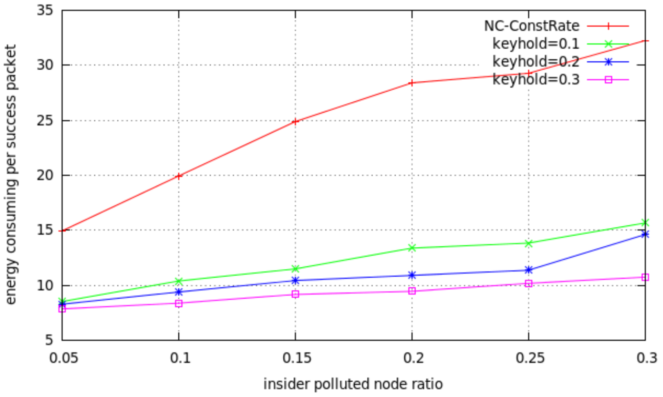


Figure 3.5 : Energy consumed per successful packet

more energy when $p_{\mathbf{S}} = p_V \times p_E$ grows. However, $p_{\mathbf{S}}$ should be kept in a limited range. One reason is the limited storage space of IoT nodes. Another reason is that adversaries capture several nodes and collect keys of all these captured nodes to reveal location privacy. Fig. 3.2 shows that a higher p_E increases content-correlation and degrades source location privacy. To balance privacy and network performance, AP-SLP should decrease p_E and increase p_V to balance privacy and authentication performance.

3.6 Summary

This chapter proposes an anti-pollution source-location privacy solution, AP-SLP, to defend against pollution attacks while protecting source-location privacy in event-triggered IoT networks. AP-SLP provides the triple-type homomorphic signature algorithm (TTHS) in network coding based IoT networks to filter out polluted and dummy packets during transmission. Keys required in TTHS are predistributed according to the opportunistic key distribution schemes, which preload each node with an end-to-end asymmetric key ring and a hop-by-hop encryption key ring. Nodes use the asymmetric key ring to sign and verify packets in an opportunistic way, and the encryption key ring to ensure source location privacy. The simulation results demonstrate that the proposed AP-SLP improves the message delivery rate and prolongs network life compared with previous works.

Chapter 4

Design and Analysis of Encrypted Data Transmission Protocol in Distributed IoT

4.1 Introduction

IoT networks communicate in an open manner, where adversaries may eavesdrop on the IoT communications and obtain sensitive messages directly, e.g., hackers may steal the information and violate privacy of patients in the smart healthcare application. Therefore protecting the data confidentiality is crucial to IoT applications. Data encryption is the basic method to protecting data confidentiality in transmission. For example, IEEE Standard 802.11p has recommended Elliptic Curve Cryptography (ECC) to encrypt data in VANET.

However, critical challenges of unstable topologies and the collisions of uncoordinated data transmissions arise, due to the mobile and distributed nature of IoT. Previously, data confidentiality has been enforced by providing a pair of nodes that intend to communicate with a pre-agreed security key, so that malicious attacks, such as eavesdropping, can be prevented. One method of providing security keys is to assign each pair of nodes with a specific pair of communication keys and pre-store all keys in each node, which requires a quite huge memory linearly to the size of the network. Another method is that each node acquires symmetric keys from a common key pool, and only those with identical keys can communicate, thereby enforcing security [38, 23]. The identical keys are found by shaking hands between two nodes. The topology needs to be stable to allow handshaking processes to happen. However, in a dynamic IoT network, such as VANET, with high mobility, the network topology can constantly change, and a relay node may quickly move out of the relaying position and be replaced by another node. Another challenge arises due to the decentralized nature of IoT networks, where nodes within each other's transmission ranges may incur collided transmissions. As a result, an unsuccessful (re)transmission attempt can be caused by either unmatched security keys or a transmission collision. This would confuse the transmitter whether a new key should be used for the next retransmission. None of existing key designs, e.g., [38, 23, 9], have taken this into account, and no practical solution avails.

Security analysis in wireless dynamic IoT networks is also a challenging task. The reason is that some wireless IoT networks exploit CSMA/CA and backed-off retransmissions to combat (re)transmission collisions. The backoff process can interact with the key selection, making the security analysis challenging. Markov chain models can evaluate the collision effects in IEEE 802.11a/e [72, 16] and IEEE 802.11p VANETs [120, 121]. However, the models did not take data security into consideration. Some other models [131, 139, 140] that did analyse data security were based on significantly simplified transmission channels. However, the oversimplifications on transmission channels degrade the accuracy of the models.

This chapter proposes a new encrypted data transmission protocol, which provides data confidentiality to dynamic IoT networks with constantly changing topologies. The key idea is that a transmitter adaptively switches between backing off transmissions and changing keys to increase success rate with matched keys. New acknowledgement (ACK) messages are designed to distinguish the cause of a failed (re)transmission between a packet collision and a mismatched key, such that the proposed switching can be implemented. A new 3-dimensional (3D) Markov chain model is also proposed, which captures the interactions among collisions and key selections. The 3D Markov model characterizes the proposed protocol and derives key performance metrics, the transmission success rate. Security analyses are also carried out using the 3D Markov model with collusion attacks considered. The secure connectivity of the proposed protocol is evaluated.

The rest of this chapter is organized as follows. In Section 4.2, the proposed protocol is elaborated on. In Section 4.3, the proposed 3D Markov chain model is described. In Section 4.4, simulation and numerical results are provided, followed by summaries in Section 4.5.

4.2 Proposed Encrypted Data Transmission Protocol

A new encrypted data transmission protocol is proposed where both the key selection and the transmission backoff due to transmission collisions are incorporated. This is based on the idea of probabilistic key distribution, i.e., EG [38], which was originally proposed for stable networks with requirements of handshaking. The proposed protocol extends the idea to highly mobile environments, where the frequent handshaking and collisions of uncoordinated packet transmissions can have a critical impact on the network connectivity and security.

We pre-assign each node with a ring of k keys randomly chosen from a pool of K_{P} keys ($k \leq K_{\mathrm{P}}$). Here, the K_{P} keys are off-line generated and k keys predistributed to specific nodes through secure channels [38, 23]. Each key has a unique key identity. The secure channels between nodes and the key pool can be available when the nodes are under manufacturing, maintenance and repair. Secure wireless channels may be required occasionally to revoke compromised keys [53]. This can be achieved, since every node typically has a dedicated pair of private and public keys, and so does the security authority maintaining the key pool. Each time the security authority is to revoke a list of keys, it uses its private key to sign the identities of revoked keys. Then, it broadcasts the revoked key identities and the signature as a revocation notification to the entire network in secure wireless channels. Once a node receives the revocation notification, it revokes all the listed keys from its key pool after verifying the signature of the revocation notification. This consideration has been adopted widely in existing schemes, such as EG [38], q-composite [23], and piggyback [100].

Consider a transmitter with N neighbours within its communication range, including the receiver which can be chosen by any routing protocol. The transmitter and the receiver can establish secure communication on-the-fly if both of them host the used encryption key. The proposed encrypted data transmission protocol is general to routing protocols, including the designated routing protocols and opportunistic routing protocol. Here, designated routing protocols [15, 84] designate receiver based on the shortest path or other specific metrics, while the opportunistic routing protocol [51] designates the neighbour which has the used encryption key as the receiver. To simplify the illustration, the descriptions in this chapter are applied to any routing protocol unless a specific routing protocol is indicated. It is assumed the considered network is saturated, i.e., the nodes always have messages to send. The assumption is reasonable in the presence of a large number of nodes which have to compete for access to the limited channel resources. Transmit queues build up at every individual node [113, 59, 128, 112]. When the transmitter is to send a new message, the transmitter randomly chooses a key, denoted by κ_1, from its key ring to encode the message. The subscript "$_1$" of κ_1 indicates the key is the first randomly selected to transmit the message. The transmitter also randomly selects a backoff timer from the initial backoff window $[0, W_0 - 1]$, where W_0 is the initial backoff window size. Then the transmitter starts to count down the backoff timer. Meanwhile, it keeps sensing the availability of the channel. If a transmission is sensed, the countdown is interrupted, and will not be resumed until the channel is sensed to be free again. Once the timer becomes zero, the transmitter starts to transmit the encrypted message.

There are two possible causes if the transmission attempt is unsuccessful – a transmission collision or the unavailability of κ_1 at the receiver(s)*. Two new types of ACK, which the receive can return, are designed to distinguish these two causes. In the case where the receiver receives and successfully decodes the message with the matched key, it will return an ACK1. In the case where the receiver receives the message but fails to decode it, which means the receiver is not equipped with κ_1. The receiver will return an ACK2. In the remaining cases the receiver fails to receive the message, due to the collision between the co-current transmissions of the transmitter and other nodes (including the receiver). No ACK will be returned.

The transmitter will retransmit the message if ACK1 is not responded, i.e., either only receipt of an ACK2 or in the absence of any ACK. In response to an ACK2, the transmitter will choose another key, κ_2, from its key ring to encrypt the message and a new backoff timer from the initial backoff window $[0, W_0 - 1]$, and retransmit the message. In the absence of ACK, the transmitter does not change the key which is in use. It will double the backoff window $[0, W_i - 1]$, randomly pick up a timer from the window, and retransmit the encrypted message. $W_i = 2^i W_0$, where i indicates the i-th retransmission attempt using a given key. Such backoff procedure follows the CSMA/CA.

The maximum number of retransmission attempts, denoted by M, is set under one single key. When M is reached and still no ACK has been returned, the transmitter will change the key and start to transmit a new message with the initial backoff window $[0, W_0 - 1]$. We also set the maximum number of keys to be tried for a message, denoted by $r \leq k$. When r keys have been tried to encrypt the message and the receiver has returned ACK2 for all the keys, the transmitter will drop the current message and start to send a new message.

*The unavailability of κ_1 at the receiver indicates that the designated receiver does not stores κ_1 in the case of designated routing protocols, and none of the neighbours stores κ_1 in the case of the opportunistic protocol, respectively.

4.3 3D Markov Chain Modelling and Performance Metrics

In this section, a new 3D Markov model is proposed to characterize the proposed encrypted data transmission protocol. Based on the 3D Markov Model, network performance metrics, including the collision possibility p_c, transmission possibility τ, transmission success rate P_{ST}, transmissions delay D and secure transmission success rate P_{STSR}, are derived.

4.3.1 3D Markov Chain Modelling

A new 3-dimensional (3D) Markov chain model is developed. In the proposed model, each state of the Markov process is denoted by a 3-tuple $(i, j, t), 0 \leq i \leq M, 0 \leq j \leq W_i - 1, 1 \leq t \leq r$. Each state (i, j, t) indicates that current message at the transmitter has been retransmitted for i times with the t-th chosen key and the current backoff timer is j. For illustration convenience, we assume that the considered networks are saturated. In other words, transmission queues remain non-empty at the nodes.

The states of 3D Markov model can be collected by totally r parallel planes. The t-th parallel collects the Markov states corresponding to the t-th selected key at the transmitter, κ_t. Each horizontal chain/row of states on a plane, say row i, lists all the backoff timer for a given (re)transmission attempt i. The backoff timer j in the i-th chain ranges from 0 to $(W_i - 1)$. A transition can happen between two adjacent state on the same row, from j to $(j - 1)$, describing the countdown process of the backoff timer. A transition can take place between two adjacent rows on the same plane, from top to bottom, caused by a transmission collision. The input state of such a transition can only be the one on the left end of the upper row/chain where $j = 0$, since that state is where a (re)transmission occurs.

Transitions can also take place between planes, where the input states are those leftmost on the planes for the aforementioned reason. The transitions from top to bottom can only take place between the adjacent planes. They are caused by the unavailability of the keys, which the transmitter used to transmit the current message, at the receiver(s). In this case, the receiver fails to decode the current message, and a new key is to be selected at the transmitter for a retransmission. There are also transitions from any planes to the top plane, which are the results of successfully decoding the previous message at the receiver. In this case, the receiver and the transmitter have a common key. The transmitter will start to transmit a new message, as captured in the top plane. The new message can have a different receiver, and a randomly selected key from the key ring of the transmitter will be used for the new transmission. One exception is the transition from the bottom line of any plane to the top plane. In addition to the successful decoding of the previous message, the transition can also be triggered by reaching the maximum number of retransmissions of the current encrypted message, M. Another exception is the transition from the bottom plane to the top plane. In addition to the successful decoding of the previous message, the transition can also be triggered by a reached maximum number of keys that the transmitter can try – r. All the above transitions are based on the design of the proposed encrypted data transmission protocol; see Section 4.2.

4.3.2 Transition Probability and Stationary Probability

The transition probability between each pair of states in the proposed Markov model can be explicitly calculated in the following cases.

In the case of (i, j, t) with $j > 0$, the transmitter continues to count down its backoff timer until $j = 0$. Therefore, the transition probability can be given by,

$$\Pr[(i, j-1, t)|(i, j, t)] = 1, \qquad j > 0. \tag{4.1}$$

In the case of $(i, 0, t)$, there are three possible outcomes states following the state $(i, 0, t)$, depending on the respond that transmitter receives.

For the transition from $(i, 0, t)$ to $(i + 1, 0, t)$, it happens when transmitter receives no ACKs and $i < M$. In this case, the transmission collides. The transition probability can be given by

$$\Pr[(i + 1, j, t)|(i, 0, t)] = \frac{1}{W_{i+1}} p_c, \qquad i < M, \tag{4.2}$$

where p_c is the collision probability and can be given by,

$$p_c = 1 - (1 - \tau)^N, \tag{4.3}$$

where τ is the transmission probability per node per slot, and will be discussed in detail later. In this case, the transmitter still uses current key and increases stage index to $(i+1)$. A backoff counter j is chosen from $[0, W_{i+1} - 1]$. $\frac{1}{W_{i+1}}$ is due to the uniformly selected new backoff counter on stage $(i + 1)$.

For the transition from $(i, 0, t)$ to $(0, j, t + 1)$, it happens when the transmitter receives ACK2 and $t < r$. In this case, key is proved unmatched. The transition probability can be given by

$$\Pr[(0, j, t + 1)|(i, 0, t)] = \frac{1}{W_0}(1 - p_c)p_{k_t}, \qquad t < r, \tag{4.4}$$

where p_{k_t} is conditional probability that the κ_t is not preloaded in receiver(s) on condition that $(t - 1)$ keys have already been used to encrypt the current message and also unmatched. $p_{k_t} = \binom{K_P - t}{k} / \binom{K_P - t+1}{k}$ in the case of the designated routing protocols, and $p_{k_t} = \left[\binom{K_P - t}{k} / \binom{K_P - t+1}{k}\right]^N$ in the case of the opportunistic protocol. $(1 - p_c)p_{k_t}$ is the probability that no collision happens and κ_t is unmatched. In this case, a new key is chosen and stage index is initialized from stage-0 on Plane-$(t + 1)$. $\frac{1}{W_0}$ is due to the uniformly selected new backoff counter on stage 0.

For the transition from $(i, 0, t)$ to $(0, j, 1)$. If $i < M$ and $t < r$, such transition happens when the transmitter receives an ACK1. In this case, the transmission succeeds, i.e., the transmission is collision-free and the key matches at the receiver(s). The transition probability can be given by

$$\Pr[(0,j,1)|(i,0,t)] = \frac{1}{W_0}(1-p_c)(1-p_{k_t}),\ i < M, t < r, \tag{4.5}$$

where $(1-p_c)(1-p_{k_t})$ is the probability that no collision happens and key matches.

In the case of $i = M$ and $t < r$, transition from the state $(M,0,t)$ to $(0,j,1)$ happens when transmission succeeds or collision happens. The transition probability can be given by

$$\Pr[(0,j,1)|(M,0,t)] = \frac{1}{W_0}[1-(1-p_c)p_{k_t}], \qquad t < r, \tag{4.6}$$

where $[1 - (1 - p_c)p_{k_t}]$ is the probability that transmitter receives ACK1 or no response. If transmitter receives ACK1, transmission ends successfully and the transmitter starts to transmit a new message with a new selected key. Model is initialized from stage-0 on the first plane. Else no response answers, which denotes collision happens for the state $(M,0,t)$. As the stage index reaches the maximize stage index threshold M, the transmitter drops current message and initializes model from stage-0 on the first plane.

In the case of $i < M, t = r$, transition from $(i,0,r)$ to $(0,j,1)$ happens when transmission succeeds or the key is unmatched. The transition probability can be given by

$$\Pr[(0,j,1)|(i,0,r)] = \frac{1}{W_0}(1-p_c), \qquad i < M, \tag{4.7}$$

where $(1-p_c)$ is the probability that no collision happens. If the key proves matched at the receiver, transmission succeeds and the transmitter starts to transmit a new message. In this case, model is initialized from stage-0 on the first plane. Else key is unmatched, as the key index reaches the maximum key index threshold, r, the transmitter drops the current message and initializes model from stage-0 on Plane-1.

In the case of $r = M, t = r$, the transition probability from $(M,0,r)$ to $(0,j,1)$

can be given by

$$\Pr[(0,j,1)|(M,0,r)] = \frac{1}{W_0}, \tag{4.8}$$

as $(M,0,r)$ is the last optional state in model, the transmitter starts to transmit a new message and initializes model from stage-0 on the first plane no matter what responds.

Given the transition probabilities of (4.1), (4.2), and (4.4) to (4.8) as above, we are able to calculate the stationary probability of state (i,j,t), denoted by $\Pr(i,j,t)$, of the new 3D Markov chain model.

The states $(i,j,t), i > 0$ can be written as given by

$$\begin{aligned}\Pr(i,j,t) &= \frac{1}{W_i} p_c \Pr(i-1,0,t) + \Pr(i,j+1,t)\\ &= \frac{W_i - j - 1}{W_i} p_c \Pr(i-1,0,t) + \Pr(i,W_i - 1,t)\\ &= \frac{W_i - j}{W_i} p_c \Pr(i-1,0,t), i > 0\end{aligned} \tag{4.9}$$

where the first equality is obtained by noting state (i,j,t) can only be transited from $(i,j+1,t)$ or $(i-1,0,t)$ and therefore can be written using (4.1) and (4.2). The second equality can be obtained by repeating this recursively to $\Pr(i,j+1,t),\cdots,\Pr(i,W_i-2,t)$. The third equality is because (i,W_i-1,t) can only be transited from $(i-1,0,t)$ and can be obtained using (4.2).

Note that $\Pr(i,0,t) = p_c \Pr(i-1,0,t) = p_c^i \Pr(0,0,t)$ by letting $j = 0$ in (4.9). As a result, (4.9) can be rewritten as

$$\Pr(i,j,t) = \frac{W_i - j}{W_i} p_c^i \Pr(0,0,t). \tag{4.10}$$

The state $(0,j,t), t > 1$ can be given by

$$\begin{aligned}\Pr(0,j,t) &= \frac{W_0 - j}{W_0}(1-p_c)p_{k_{t-1}} \sum_{i=0}^{M} \Pr(i,0,t-1)\\ &= \frac{W_0 - j}{W_0}(1-p_c^{M+1})p_{k_{t-1}} \Pr(0,0,t-1), t > 1\end{aligned} \tag{4.11}$$

where the first equality is obtained by recursively decomposing $\Pr(0,j,t) = \frac{1}{W_0}\sum_{i=0}^{M}(1-p_c)p_{k_{t-1}}\Pr(i,0,t-1)+\Pr(0,j+1,t)$ based on (4.1) and (4.4), because state $(0,j,t)$ can only transit from $(i,0,t-1)$ or from $(0,j+1,t)$ for $j < W_0$. The second equality is because $\Pr(i,0,t-1) = p_c^i\Pr(0,0,t-1)$, as noted earlier, and $\sum_{i=0}^{M}(1-p_c)p_c^i = 1-p_c^{M+1}$.

Let $j = 0$ in (4.11). $\Pr(0,0,t) = (1-p_c^{M+1})p_{k_{t-1}}\Pr(0,0,t-1)$. Recursively substituting this into (4.11), we have

$$\Pr(0,j,t) = \frac{W_0-j}{W_0}(1-p_c^{M+1})^{t-1}\Pr(0,0,1)\prod_{s=1}^{t-1}p_{k_s}. \tag{4.12}$$

Let $j = 0$ in (4.12) and substitute the result back to (4.12). We finally obtain

$$\Pr(0,j,t) = \frac{W_0-j}{W_0}\Pr(0,0,t), \qquad t > 1, \tag{4.13}$$

which is also captured by (4.10).

The remaining state $(0,j,1)$ can be given by

$$\begin{aligned}\Pr(0,j,1) =& \frac{W_0-j}{W_0}\Big[\Pr(M,0,r)+\sum_{t=1}^{r-1}\sum_{i=0}^{M-1}(1-p_c)(1-p_{k_t})\Pr(i,0,t)\\ &+\sum_{t=1}^{r-1}[1-(1-p_c)p_{k_t}]\Pr(M,0,t)+\sum_{i=0}^{M-1}(1-p_c)\Pr(i,0,r)\Big]\\ =&\frac{W_0-j}{W_0}\Pr(0,0,1),\end{aligned} \tag{4.14}$$

where the first two equalities can be derived similarly as the first equality of (4.11), based on (4.1), (4.5), (4.6), (4.7) and (4.8). The last equality is obtained by substituting $j = 0$ in the first equality and substitute that back into the first equality. Obsiviously, $\Pr(0,j,1)$ also satisfies (4.10). Based on (4.10), (4.13) and (4.14), (4.10) is the unified expression for any (i,j,k).

4.3.3 Collision Probability

Note that

$$\sum_{t=1}^{r}\sum_{i=0}^{M}\sum_{j=0}^{W_i-1}\Pr(i,j,t) = 1. \tag{4.15}$$

Substituting (4.10) into the left-hand side of (4.15) and performing mathematical manipulations, we have

$$\sum_{t=1}^{r}\sum_{i=0}^{M}\sum_{j=0}^{W_i-1}\Pr(i,j,k) = \sum_{t=1}^{r}\Big[\sum_{i=0}^{M}\sum_{j=0}^{W_i-1}\frac{W_i-j}{W_i}p_c^i\Pr(0,0,t)\Big]$$
$$=\sum_{t=1}^{r}\frac{1}{2}\Big[W_0\frac{1-(2p_c)^{M+1}}{1-2p_c}+\frac{1-p_c^{M+1}}{1-p_c}\Big]\Pr(0,0,t).$$

As a result,

$$\sum_{t=1}^{r}\Pr(0,0,t) = \frac{2}{[W_0\frac{1-(2p_c)^{M+1}}{1-2p_c}+\frac{1-p_c^{M+1}}{1-p_c}]}. \tag{4.16}$$

The transmission probability τ can be written, as given by

$$\tau = \sum_{t=1}^{r}\sum_{i=0}^{M}\Pr(i,0,t) = \frac{(1-p_c^{M+1})}{1-p_c}\sum_{t=1}^{r}\Pr(0,0,t), \tag{4.17}$$

since a transmission takes place when the backoff timer $j = 0$, and $\Pr(i,0,t) = p_c^i\Pr(0,0,t)$, as noted earlier.

p_c and τ can be obtained by jointly solving (4.3), (4.16) and (4.17).

4.3.4 Transmission Success Rate

$P_s(i,0,t)$, denoting the probability that a message is successfully transmitted in the state $(i,0,t)$, can be given by

$$P_s(i,0,t) = (1-p_c)(1-p_{k_t})\frac{\Pr(i,0,t)}{\Pr(0,0,1)} = (1-p_c)(1-p_{k_t})p_c^i\prod_{s=1}^{t-1}\Big[p_{k_s}(1-p_c^{M+1})\Big],$$

where $(1-p_c)(1-p_{k_t})$ is the probability that a receiver receives and decodes a given (re)transmission with the key κ_t. $\frac{\Pr(i,0,t)}{\Pr(0,0,1)}$ is the possibility that the message can be transmitted in the state $(i,0,k)$, given the condition that at least one transmission happens at state $(0,0,1)$. The second equality can be obtained by substituting (4.10) and (4.12).

Let P_{ST} be the transmission success rate per message. It can be calculated using $P_s(i,0,t)$, as given by

$$P_{ST} = \sum_{t=1}^{r}\sum_{i=0}^{M}P_s(i,0,t) = \sum_{t=1}^{r}(1-p_c^{M+1})^t(1-p_{k_t})\prod_{s=1}^{t-1}p_{k_s}.$$

4.3.5 Secure Transmission Success Rate

Collusion attack, as a type of cyber-security breach, poses severe security threats to decentralized wireless networks [98]. In collusion attacks, adversaries physically capture legitimate nodes and turn the nodes to be internal adversaries [56]. The attack model in this chapter is that collusion attackers physically capture N_C nodes in a network of N_T nodes. The keys of the N_C captured nodes are gathered to derive the keys of the remaining $(N_T - N_C)$ non-captured nodes for eavesdropping or other attacks. Secure transmission success rate (STSR), which quantifies the secure transmission ability in hostile environments, can be defined as the probability that a message is successfully transmitted from a non-captured node to another non-captured node, and the message content keeps confidential to the captured nodes [49]. In other words, the key that non-captured nodes use is unavailable to all the N_C captured nodes.

The STSR can be given by

$$P_{STSR} = \left(1 - \frac{N_C}{N_T}\right)^2 \sum_{t=1}^{r} \Big[\sum_{i=0}^{M} P_s(i, 0, t) \sum_{x=k}^{\min\{K_P - t, kN_C\}} P_{ck}(x|N_C) \frac{\binom{K_P - x}{t}}{\binom{K_P}{t}} \Big], \quad (4.18)$$

where $(1 - \frac{N_C}{N_T})^2$ is the probability that both the transmitter and receiver of the successful transmission are non-captured. $\sum_{i=0}^{M} P_s(i, 0, t)$ is the probability that the transmitter transmits the message successfully with the key κ_t; see Section 4.3.4. In this case, the non-captured transmitter has tried t different keys, and the t keys should all be unavailable to captured nodes. The probability of this is given in the last summation of the equation. $P_{ck}(x|n)$ is the probability that n nodes stores x distinct keys, $k \leq x \leq \min\{nk, K_P\}$. $P_{ck}(x|n)$ can be calculated recursively by

$$P_{ck}(x|n) = \sum_{\forall x' \leq x} \left(P_{ck}(x'|n-1) \frac{\binom{x'}{k - x + x'} \binom{K_P - x'}{x - x'}}{\binom{K_P}{k}} \right), \quad (4.19)$$

where $\binom{x'}{k-x+x'}\binom{K_P-x'}{x-x'} \Big/ \binom{K_P}{k}$ is the probability that the n-th captured node exposes $(x - x')$ new keys apart from the x' keys exposed earlier by the $(n - 1)$ previously captured nodes. The recursive calculation can be initialized by $P_{ck}(k|1) = 1$ and $P(k'|1) = 0$ for any $k' \neq k$, since every node stores exactly k keys.

A special case of P_{STSR} is when $N_C = 0$, i.e., the network is not under the collusion attack. According to Section 4.3.4,

$$P_{STSR} = \sum_{t=1}^{r} \sum_{i=0}^{M} P_s(i, 0, t) = P_{ST}.$$

The validity of (4.18) can be confirmed.

4.4 Numerical Result

In this section, simulation and numeral results are provided to evaluate the proposed protocol. The parameters used in this chapter are summarized in Table 4.1. A thousand independent runs are carried out, and averaged for every simulation result.

Table 4.1 : Simulation Parameters

Parameter	Explanation	Value
M	backoff stage threshold	6
W_0	initial backoff window size	16
K_P	size of key pool	200,1000
k	size of key ring	[1%K_P,30%K_P]
N	number of neighbours per node	$10, 20$
N_T	number of nodes in whole area	$100(N + 1)$
N_C	number of captured nodes in whole area	$[0, 20]$

We first evaluate the proposed encrypted data transmission protocol in the case of the designated routing protocol. Fig. 4.1 compares the analytical and simulation

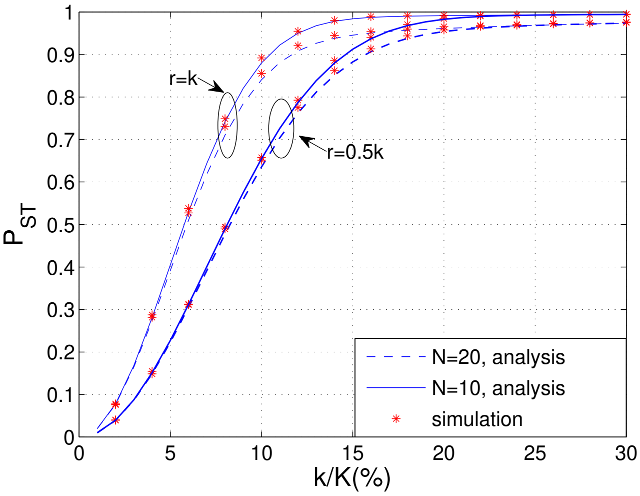


Figure 4.1 : Transmission success rate in the case of designated routing protocol, where the key ring size ranges from $1\%K_P$ to $30\%K_P$, $N = 5$ and 10, and $r = 0.5k$ and k.

results of the transmission success rate for the proposed protocol, where the x-axis is the percentage of the key ring of every node with respect to the entire key pool. It is shown that the analytical results of P_{ST} coincide with the simulations. The validity of the proposed 3D Markov model is confirmed. It is also revealed in the figure that the maximum number of keys that a transmitter tries for sending a message, i.e., r, stops from making difference, when the key ring size is larger than $k = 23\%K_P$. This is due to the increased likelihood of having keys shared between the transmitter and receiver. A small number of tries can therefore identify a shared key.

Fig. 4.2 plots the secure transmission success rate of the proposed scheme. In Fig. 4.2, we see that P_{STSR} decreases with the growth of N_C, due to an increased number of exposed keys. Another observation is that, for every given N_C, the P_{STSR} curve is concave with a peak. At the peak, the improved connectivity among non-captured nodes and the increased keys (carried by the N_C captured nodes) exposed to the adversaries, both of which are due to the enlarged key ring size k, are balanced.

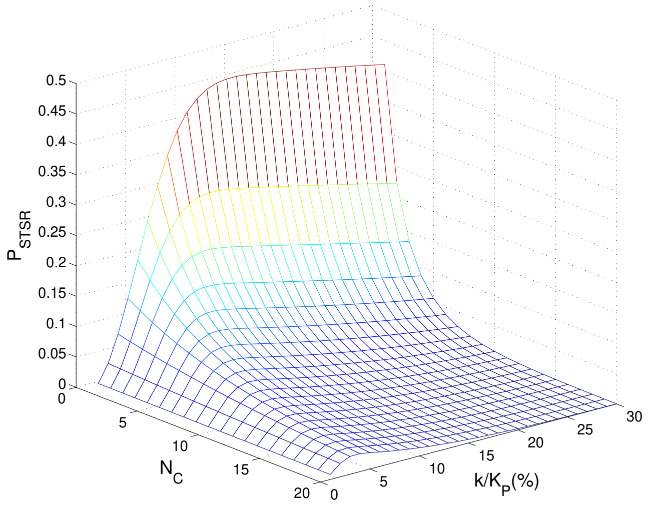


Figure 4.2 : Secure transmission success rate, where k ranges from $1\%K_P$ to $30\%K_P$, N_C ranges from 1 to 20.

The k value corresponding to the peak is referred to as the "optimal key ring size". Before the peak, the connectivity has a dominating effect, and P_{STSR} increases with k; after that, the adverse effect of the increased exposed keys becomes dominant, and P_{STSR} decreases as k grows.

It is shown in the figure that the optimal key ring size decreases, as N_C increases. This is because the number of exposed keys increases with N_C; whereas the connectivity only depends on k and is independent of N_C. The adverse effect of the increased exposed keys becomes dominant over the connectivity. To offset the adverse effect, the optimal key ring size needs to reduce, thereby reducing the number of exposed keys.

In Fig. 4.3, it is shown that P_{STSR} keeps stable against r in the case of large k, i.e., $k \geq 20\%K_P$, since fewer keys need to be tried before a successful transmission due to a large number of shared keys at the transmiter and receiver, as discussed earlier. We also see that there is an optimal key ring size, for every given $\frac{r}{k}$ value,

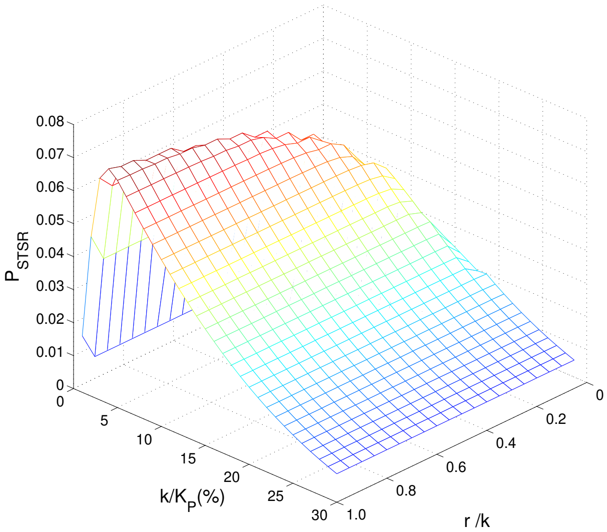


Figure 4.3 : Secure transmission success rate, where k ranges from $1\%K_P$ to $30\%K_P$, r ranges from $0.05k$ to k, $N_C = 5$.

and the optimal key ring size decreases with the increase of $\frac{r}{k}$. The optimal key ring size achieves the balance, between the adverse effect of the increased number of exposed keys during the transmission of a message, and the increased connectivity, both due to the increase of k. Therefore, the optimal key ring size decreases with r (which is proportional to k in the figure).

Fig. 4.4 compares the transmission success rate of the proposed protocol based on the opportunistic routing with that of the EG scheme. When the network is dense, e.g., $N = 40$, the proposed opportunistic routing based protocol requires only one key per message, i.e., $r = 1$, and $k/K_P > 16\%$ to achieve the same transmission success rate as the EG scheme. In practice, EG scheme requires at least two extra collision-free transmissions for handshaking before transmitting messages. Therefore the proposed protocol based on opportunistic routing requires a fewer number of transmissions to provide the same transmission success rate, i.e., it transmits more efficiently than the practical EG scheme. However, Fig. 4.1 shows that

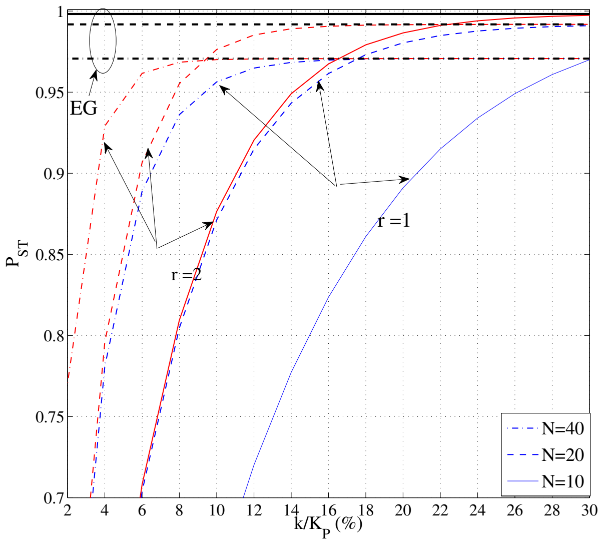


Figure 4.4 : Transmission success rate in the case of opportunistic routing protocol, where the key ring size ranges from $1\%K_P$ to $30\%K_P$, $N = 10, 20, 40$, and $r = 1, 2$

the proposed protocol based on designated routing requires a number of collision-free transmissions before finding the matched key. It indicates that EG requires fewer transmissions than the proposed protocol based on the designated routing to achieve the same transmission success rate. The reason for the better performance of the opportunistic routing based protocol is that all neighbours of the transmitter contribute to finding the matched key. The key matching possibility between the transmitter and any neighbour is absolutely much higher than that between the transmitter and one designated receiver. In sum, the key to releasing the potential of the proposed on-the-fly encrypted data transmission protocol is to increase the key matched possibility, which can be achieved by using the opportunistic routing protocol.

4.5 Summary

In this chapter, a new encrypted transmission protocol is proposed to transmit encrypted messages on-the-fly in dynamic IoT networks, where the transmitter is enabled to switch between backing off transmissions and changing keys, adapting to the different causes of a failed (re)transmission attempt. As a result, packet collisions and mismatched keys that result in failed (re)transmissions can be adequately addressed. We also developed a new 3D Markov chain model, which characterizes the proposed protocol and analyses the secure transmission ability of the protocol. Interesting insights and useful guidelines to releasing the potential of the on-the-fly encrypted data transmission protocol are also provided.

Chapter 5

Design and Analysis of Opportunistic Authentication Protocol with Key Predistribution

5.1 Introduction

This chapter provides a new protocol of joint transmission and authentication to protect data integrity. It speeds up authentication and embraces opportunistic routing in a decentralized mobile IoT networks, such as VANET. Motivated to reduce communication overhead, key pairs are randomly predistributed across the network. Nodes in proximity, predistributed the same pair of keys, can verify and instantly route messages for each other in an opportunistic and cooperative fashion combating the fast-changing topology of the network. Another important aspect of the protocol is that a node is designed to increasingly combine collision-free yet unauthenticated messages and a new message for digital signature or message authentication code (MAC) generation, while trying different keys on-the-fly. Once a key is matched, the unauthenticated messages and the new message can be verified altogether and forwarded. In this sense, the communication overhead for authentication can be reduced to be independent of the number of keys tried.

This chapter also conducts a comprehensive analysis on the proposed protocol. The interactions between the uncoordinated transmissions and key selection of a node are explicitly modelled as a three-dimensional (3D) Markov chain. Analytical results based on the 3D Markov chain reveal that the key selection does not intensify the transmission collisions, but it does affect the authentication rate, delay and throughput, especially under collusion attacks with the keys being continually revoked. Validated by Monte-Carlo simulations, analytical results also confirm the tolerance of the proposed protocol against changing topologies, as well as the substantially improved resistance against collusion attacks, as compared to the prior art. The improvement enlarges, as the network gets denser and/or the nodes move faster.

We note that the proposed protocol and Markov model are general and not limited to specific keys types. In the case of symmetric keys, a short MAC can be generated at a transmitter and verified at a neighbouring node predistributed the same key. In the case of asymmetric key pairs, a signature can be generated using a predistributed private key and verified at a node predistributed the corresponding public key. Of course, only the keys from the same key pool should be predistributed and used within a network. Moreover, this chapter provides performance comparisons between symmetric and asymmetric keys. On one hand, the symmetric keys generating MACs incur substantially lower computational costs than asymmetric keys which generating signatures. Typically, the authentication time is about 1 μs for a MAC [125] and about 4 ms for a signature, e.g., using ECDSA [21], both of which, however, are shorter by orders of magnitude than the average per-message authentication delay caused by collided (re)transmissions, as will be shown in this chapter. On the other hand, asymmetric keys are shown to be much more robust against collusion attacks than symmetric keys in the proposed protocol. This is because the number of asymmetric key pairs exposed is far less than that of symmetric keys, provided the same number of nodes compromised.

The rest of this chapter is organized as follows. In Section 5.2, the network model and proposed authentication protocol are elaborated on. In Section 5.3, the comprehensive modelling and analysis of the proposed protocol are conducted, based on which the resistance of the protocol against collusion attacks is quantified in Section 5.4. In Section 5.5, numerical results are provided, followed by summaries in Section 5.6.

5.2 Key Predistribution and Authentication Protocol

In this section, opportunistic route discovery for decentralized mobile networks is described, where nodes cooperatively relay messages to cope with fast-changing topologies and unreliable uncoordinated transmissions of the networks. A new authentication process is designed, which enables opportunistic routing, or in other words, cooperative relaying.

5.2.1 Network Setup

We consider a scenario, where a transmitter, denoted by Node 0, sends messages to an intended destination with the assistance of its neighbouring (relay) nodes, denoted by Nodes $1, \cdots, N$, in a decentralized mobile IoT network. N is the number of neighbours within the transmission range of the transmitter at an instant. Consider the mobility of the nodes, the N nodes can change from one instant to another. Only authenticated messages are further forwarded. Opportunistic routing [51, 5] is adopted, such that one of the neighbours correctly receiving an authentic message can instantly forward the message. In this sense, routing (or route discovery) is opportunistic and is accomplished on-the-fly while the message is being forwarded.

5.2.2 Key Predistribution

Assume that a pool of K_P key pairs are off-line generated and predistributed in a secure way to the nodes in the decentralized mobile IoT network. As aforementioned, the proposed authentication protocol is general to symmetric and asymmetric keys. However, only key pairs from the same key pool, i.e., symmetric or asymmetric, should be predistributed and used within one single network. The predistribution of asymmetric and symmetric keys are illustrated as follows.

Asymmetric Key Predistribution

In the case that asymmetric keys are predistributed, every node in the network is predistributed K_1 private keys and K_2 public keys, randomly chosen from the K_P key pairs ($K_1 \leq K_P$, $K_2 \leq K_P$). Consider a large-scale decentralized mobile network, K_P is set to be large, e.g., $10^4 \sim 10^5$. K_1 is typically around 10 to 20, preserving robustness against the collusion attacks, as will be discussed in Section 5.5. K_2 is typically up to $5\% \sim 10\%$ of K_P, thereby leveraging between connectivity and the memory requirement per node.

A node uses the private keys to sign its own messages and produce signatures, and uses the public keys to verify the others' messages. The private key that signs a message, and the public key that authenticates the message must be from the same key pair. The proposed protocol does not need to decide on the common key through handshaking. As mentioned in Section 5.2.1, the proposed protocol authenticates messages in an opportunistic manner. Any neighbour of the transmitter, predistributed the same key, can authenticate the message by verifying the signature.

Symmetric Key Predistribution

In the case that symmetric keys are predistributed, every node in the network is predistributed $K_2(= K_1)$ keys, randomly chosen from the K_P keys in the key pool. As the captured secret keys degrade the network security, will be discussed in Section 5.5, more keys predistributed per node bring to less security. On the other hand, the number of keys predistributed per node is in proportion to the successful authentication rate. Therefore K_2 is typically taken around a few hundred, balancing connectivity and resilience against collusion attacks. A node uses a symmetric key to produce a MAC, and transmits the MAC and its associated message together. A neighbour of the node, predistributed the same key, can authenticate the message by verifying the MAC.

5.2.3 Transmission and Authentication

It is assumed the considered network is saturated, i.e., the nodes always have messages to send [113, 59, 128, 112]. This assumption is reasonable as explained in Section 4.2. The transmitter attaches authentication information to each message before transmission. Consider CSMA/CA is applied for wireless transmissions, which is extensively employed in decentralized IoT networks, e.g., WSN and VANET [3]. Receivers, which can verify the integrity of incollided received messages with the authentication information, will further forward the messages. Fig. 5.1 provides the flowchart of the proposed authentication protocol at the transmitter. For illustration purpose, we consider the case that asymmetric keys are predistributed in Fig. 5.1 (however, the same process also applies to the case that symmetric keys are predistributed). In Fig. 5.1, index i indicates the i-th (re)transmission of a message and the associated signature, j is the random backoff timer, k is the number of messages hashed and signed to produce the signature, and $\pi(k)$ is the k-th selected private key to produce the signature.

The transmitter starts to transmit a message by generating a signature with one of its private keys in the case where asymmetric, or a MAC with one of its symmetric keys in the case where symmetric keys are predistributed. The transmitter also sets a uniformly random timer within the initial backoff window W_0, and counts down the timer by one per timeslot. Meanwhile, the transmitter keeps sensing the channel. It freezes counting if the channel is busy, and resumes only after the channel is free again. The transmission of the message and signature/MAC is triggered, once the timer becomes zero.

In the case that the transmission is collision-free and some neighours store the matched key, acknowledgments (ACKs) are returned by neighbours which suc-

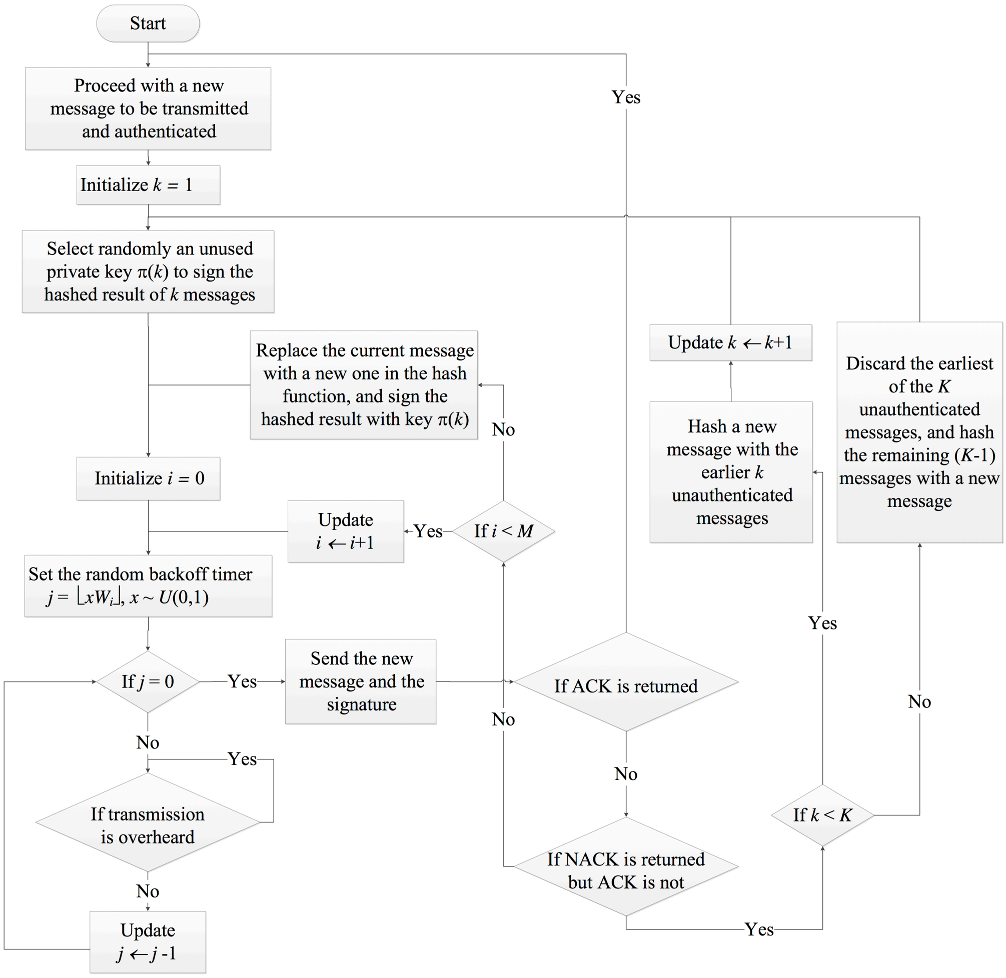

Start
Proceed with a new message to be transmitted and authenticated
Yes
Initialize k = 1
Select randomly an unused private key $\pi(k)$ to sign the hashed result of k messages
Replace the current message with a new one in the hash function, and sign the hashed result with key $\pi(k)$
No
Update $k \leftarrow k$+1
Discard the earliest of the K unauthenticated messages, and hash the remaining (K-1) messages with a new message
Hash a new message with the earlier k unauthenticated messages
Initialize i = 0
Update $i \leftarrow i$+1
Yes
If $i < M$
Set the random backoff timer $j = \lfloor xW_i \rfloor$, $x \sim U(0,1)$
Yes
If $j = 0$
Yes
Send the new message and the signature
If ACK is returned
No
No
Yes
No
No
If transmission is overheard
No
If $k < K$
If NACK is returned but ACK is not
Update $j \leftarrow j$ -1
Yes

Figure 5.1 : Flowchart of the proposed authentication operations at the transmitter.

cessfully authenticate the message with the matched key. These nodes can cooperatively relay the authenticated message towards the destination, as described in [51]. Other neighbouring nodes which do not have the matched key, return non-acknowledgments (NACKs)*. Upon the receipt of the ACKs, the transmit-

*We can design two new consecutive time periods following every (re)transmission to accommodate multiple ACKs/NACKs. The neighbouring nodes which successfully authenticate the message return ACKs in the first period. Those which fail to do so return NACKs in the second period. Contention-based transmissions can be adopted in each of the two periods to relieve communication errors, i.e., the nodes randomly but uniformly send ACKs or NACKs in the corresponding periods.

ter discards the authenticated message and proceeds with another new message, as described in the left-hand side (LHS) of Fig. 5.1. In this case, the message is successfully authenticated. Here, we assume that channel conditions are ideal, i.e., there are a finite number of nodes but no hidden nodes [113]. This is due to the fact that transmission collisions are the dominating reason for failed (re)transmissions in distributed wireless networks.

In the case that the transmission collides with those of other nodes, neither ACK nor NACK is returned. The transmitter retransmits the same message and signature/MAC for up to another M times. Each time, it doubles the backoff window $W_n = 2W_{n-1}$ ($n = 1, \cdots, M$), sets a new timer within $[0, W_n)$, and counts down the timer to trigger the retransmission. After M unsuccessful retransmission attempts, the transmitter discards the message and proceeds with a new one, as described in the middle of Fig. 5.1.

It is possible that ACKs are not returned while NACKs are. In other words, the message is received collision-free but no nodes are predistributed the matched key to authenticate it. In this case, the transmitter is designed to try up to $K \leq K_1$ different keys $\pi(1), \cdots, \pi(K)$, for the message. Every time the key is changed, the transmitter resets the backoff window to be W_0, and conducts up to $(M + 1)$ (re)transmissions, as described earlier. Here, $\pi(k)$ is the k-th selected key of the transmitter (more specifically, the k-th selected key in the case where symmetric keys are predistributed, or the k-th selected private key in the case where asymmetric keys are predistributed), provided that the counterparts of the previous $(k-1)$ keys $\pi(1), \cdots, \pi(k-1)$ are not predistributed at any neighbours.

A special design is proposed to reduce the communication cost of trying different keys. Specifically, a new message is sent each time the key is changed, e.g., to $\pi(k)$. The signature/MAC sent along with the message is produced by applying $\pi(k)$ to a combined message of the previous $(k-1)$ collision-free yet unauthenticated messages and the new message, as described in the middle of Fig. 5.1. A hash function can be used to combine the messages, due to its one-wayness and collision resistance [81][†]. The length of the signature remains unchanged, due to the fact that the output of a hash function has a consistent length. For example, the output of the hash function SHA-256 is 256 bits (or 32 bytes), provided the input to the function is no greater than $(2^{64}-1)/8 \approx 2 \times 10^{18}$ bytes. This input is far longer than the total length of unexpired messages, which is up to K times the size of a single message.

If one of the neighbours is predistributed the matched key and the (re)transmission is also collision-free, both the new and the previous $(k-1)$ unauthenticated messages can be successfully verified. (The previous $(k-1)$ messages have already been received collision-free at the neighbours.) Otherwise, if K keys are tried, K messages are delivered, but none is verified, the transmitter discards the head-of-line unauthenticated message, combines the rest of the unauthenticated messages with a new one, applies a new key, and starts to transmit the new message and the new signature/MAC, as described in the right-hand side (RHS) of Fig. 5.1. (This new key is the K-th in regards of the new head-of-line unauthenticated message.) In this sense, the different keys tried, $\pi(1)$ to $\pi(k)$, are applied to k increasingly combined messages. The communication overhead per message does not grow with the increase of the keys tried.

Occasionally, a message and its attached signature/MAC under $\pi(k)$ exhaust $(M+1)$ (re)transmissions with collisions. In that case, the transmitter discards the message, combines the $(k-1)$ unauthenticated messages with another new message, generates a new signature/MAC still using $\pi(k)$, and continues to transmit the new

[†] The collision resistance here means that no two different messages can have the same hashed outcomes, when using the same hash function [81]. This collision is conceptually different from the transmission collisions discussed in this thesis.

message and signature/MAC. The reason is that the message exhausting $(M+1)$ collided (re)transmissions is not received at any nodes, and should not be part of the combined message on which the signature/MAC is generated.

5.3 Modelling and Authentication Analysis

In this section, the new joint transmission and authentication protocol is analysed by a new 3D Markov chain model, which characterizes the behaviour of an individual node, including collisions, retransmissions, and changes of keys.

The new 3D Markov model is depicted in Fig. 5.2, where detailed transitions on the top and bottom planes and transitions from the bottom to the top plane are provided. Other transitions are suppressed for the readability of the figure. In each state of the model, $(i, j, k) : 0 \leq i \leq M; 0 \leq j \leq W_i - 1; 1 \leq k \leq K$, "$k$" indicates the k-th key $\pi(k)$ in regards of the head-of-line unauthenticated message of the transmitter. The previous $(k-1)$ keys fail to be verified, as their counterparts are not predistributed among the neighbours of the node. As designed in Section 5.2.3, $\pi(k)$ is used to produce the signature/MAC upon the combined message of $(k-1)$ received yet unauthenticated messages and a new message. "i" indicates the i-th (re)transmission attempt of the signature/MAC (after i collided attempts if $i > 0$), and the corresponding backoff window size is W_i. "j" indicates the number of timeslots that remains until the i-th (re)transmission. Note that the 3D Markov model is different from that proposed in Chapter 4. Here, k not only denotes the index of tried keys but also indicates the number of messages authenticated by the current signature/MAC. Furthermore, state transitions of this 3D Markov Model are different from transitions in Chapter 4, which will be explained later.

Following the proposed protocol, the states of the Markov model transit towards decreasing j till zero, as the backoff timer counts down for (re)transmissions. After the (re)transmissions, the states transit by incrementing i and keeping k unchanged

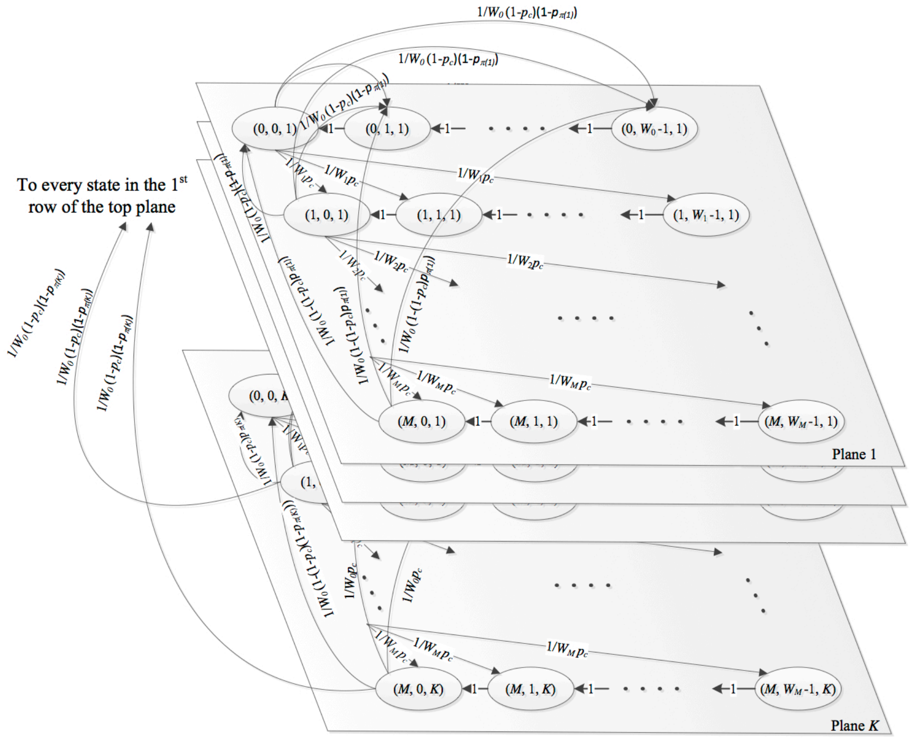


Figure 5.2 : Illustration on the proposed 3D Markov model.

due to collisions; or incrementing k and resetting $i = 0$ due to the lack of matched keys to verify collision-free (re)transmissions; or reseting both $i = 0$ and $k = 1$ after collision-free and successful authentications. In the case that i cannot be increased, i.e., collisions happen at $i = M$, the states transit by resetting $i = 0$ while keeping k unchanged. This is because the latest message exhausts $(M + 1)$ (re)transmissions with collisions, and new (re)transmissions replace this message with a new one and update the signature/MAC still using $\pi(k)$, as described at the end of Section 5.2.3.

As extensively assumed [59, 128], the collision possibility of the nodes is time invariant in the proposed 3D Markov model. This is reasonable in the presence of a large number of nodes, since a node can be randomly at different stages of retransmissions, while the retransmission of a stage is also randomly delayed. At

$$\begin{cases} \Pr[(i,j-1,k)|(i,j,k)] = 1, & j \neq 0; \quad (5.1a) \\ \Pr[(0,j,1)|(i,0,k)] = \dfrac{1}{W_0}(1-p_c)(1-p_{\pi(k)}); & (i,0,k) \neq (M,0,1); \quad (5.1b) \\ \Pr[(i+1,j,k)|(i,0,k)] = \dfrac{1}{W_{i+1}}p_c, & i < M; \quad (5.1c) \\ \Pr[(0,j,k)|(M,0,k)] = \dfrac{1}{W_0}p_c, & 1 < k < K; \quad (5.1d) \\ \Pr[(0,j,k+1)|(i,0,k)] = \dfrac{1}{W_0}(1-p_c)p_{\pi(k)}, & k < K; \quad (5.1e) \\ \Pr[(0,j,K)|(i,0,K)] = \dfrac{1}{W_0}(1-p_c)p_{\pi(K)}, & i < M; \quad (5.1f) \\ \Pr[(0,j,K)|(M,0,K)] = \dfrac{1}{W_0}[1-(1-p_c)(1-p_{\pi(K)})]; & \quad (5.1g) \\ \Pr[(0,j,1)|(M,0,1)] = \dfrac{1}{W_0}[1-(1-p_c)p_{\pi(1)}]. & \quad (5.1h) \end{cases}$$

any instant, a node is in one of the states in the 3D Markov model. Every action that the node can take is accounted for by a transition between the states.

The transition probabilities of the model are given in (5.1). Specifically, all states corresponding to the k-th key $\pi(k)$ are placed on a plane. States horizontally chained in the i-th row on the k-th plane (both from top) describe the backoff for the i-th (re)transmission under $\pi(k)$. These states certainly transit leftwards; see (5.1a), until state $(i,0,k)$ in which the (re)transmission is triggered.

In the case that the (re)transmission is collision-free and authenticated, state $(i,0,k)$ transits to any state in the first row on the top plane, i.e., $(0,j,1)$, to proceed with a new message at the probability of $\frac{1}{W_0}(1-p_c)(1-p_{\pi(k)})$; see (5.1b). Here, $(1-p_c)$ and $\frac{1}{W_0}$ are the probabilities of a collision-free (re)transmission and that a state in the first row is selected, respectively. $p_{\pi(k)}$ is the probability that no neighbours have the counterpart of $\pi(k)$ provided that the counterparts of $\pi(1),\cdots,\pi(k-1)$

are not predistributed among the neighbours, and can be given by

$$p_{\pi(k)} = \left[\frac{\binom{K_P-k}{K_2}}{\binom{K_P-k+1}{K_2}} \right]^N . \tag{5.2}$$

Therefore, $(1-p_{\pi(k)})$ is the probability that the counterpart of $\pi(k)$ is predistributed among the neighbours.

In the case that the (re)transmission is collided, state $(i,0,k)$ can transit to any state in the row for $(i+1)$ on the same plane, i.e., $(i+1,j,k)$, at the probability of $\frac{1}{W_{i+1}}p_c$ if $i < M$; or any state in the first row on the plane, i.e., $(0,j,k)$, at the probability of $\frac{1}{W_0}p_c$ if $i = M$; see (5.1c) and (5.1d).

In the case that the (re)transmission is collision-free but fails to be authenticated due to the lack of the counterpart key, state $(i,0,k)$ can transit to any state in the first row of the next plane, i.e., $(0,j,k+1)$, or remain on the last plane, i.e., $(0,j,K)$, $\forall j$, if the transmission happens on the last plane, both at the probability of $\frac{1}{W_0}(1-p_c)p_{\pi(k)}$; see (5.1e) and (5.1f).

Note that state $(M,0,K)$ is a special case and is not included in (5.1d) or (5.1f), because the failed authentication can be caused by either a collision or the lack of matched keys. It can transit to any state in the first row of the last plane, i.e., $(0,j,K)$, at the probability of $\frac{1}{W_0}[1-(1-p_c)(1-p_{\pi(K)})]$; see (5.1g).

The only remaining case is that state $(M,0,1)$ can transit to any state in the first row on the top plane, i.e., $(0,j,1)$, at the probability of $\frac{1}{W_0}[1-(1-p_c)p_{\pi(1)}]$, after either a collided (re)transmission or a successful authentication; see (5.1h).

The above cases exhaust all possible transitions between the states of the proposed Markov model. The transition probabilities, provided in (5.1), are complete. The performance of the authentication protocol is analysed with the following new theorem.

Theorem 1: The proposed joint transmission and authentication protocol (see

Sec. 5.2.3) does not intensify collisions. The transmission and collision probabilities per timeslot, τ and p_c, are independent of the key pool size K_P, the key ring sizes K_1 and K_2, and the key selection $\pi(k), k = 1, \cdots, K$.

Proof 5.1: The key idea of the proof is based on the 3D Markov chain model. The transition probabilities of the model are evaluated and collapsed to derive the steady probability distribution of the Markov model and in turn, the transmission probability of a node. Collision probabilities, capturing the interaction among all nodes, are used to connect the transmission probabilities of individual nodes. Both the transmission probability and collision probability can be shown to be independent of the key selection, validating in the theorem.

The steady probabilities of states $(i, j, k), i > 0$, can be written as

$$\begin{aligned}
\Pr(i, j, k) &= \frac{1}{W_i} p_c \Pr(i-1, 0, k) + \Pr(i, j+1, k) && \text{(5.3a)} \\
&= \frac{W_i - j - 1}{W_i} p_c \Pr(i-1, 0, k) + \Pr(i, W_i - 1, k) && \text{(5.3b)} \\
&= \frac{W_i - j}{W_i} p_c \Pr(i-1, 0, k) && \text{(5.3c)} \\
&= \frac{W_i - j}{W_i} p_c^i \Pr(0, 0, k), && \text{(5.3d)}
\end{aligned}$$

where (5.3a) is due to the fact that state (i, j, k) can only transit from $(i, j+1, k)$ or $(i-1, 0, k)$, and therefore can be written using (5.1a) and (5.1c); (5.3b) is obtained by recursively incrementing j and substituting the updated (5.3a) into the RHS of (5.3a); (5.3c) is because $(i, W_i - 1, k)$ can only transit from $(i-1, 0, k)$, and can therefore be given by $\Pr(i, W_i - 1, k) = \frac{1}{W_i} p_c \Pr(i-1, 0, k)$ based on (5.1c). (5.3d) is because $\Pr(i, 0, k) = p_c \Pr(i-1, 0, k) = p_c^i \Pr(0, 0, k)$ by letting $j = 0$ in (5.3).

The steady probabilities of states $(0, j, k), 1 < k < K$, can be given by

$$
\begin{aligned}
\Pr(0, j, k) =& \Pr(0, j+1, k) + \frac{1}{W_0}(1-p_c)p_{\pi(k-1)} \sum_{i=0}^{M} \Pr(i, 0, k-1) \\
&+ \frac{1}{W_0} p_c \Pr(M, 0, k) && \text{(5.4a)} \\
=& \frac{W_0 - j}{W_0} \Big[(1-p_c)p_{\pi(k-1)} \sum_{i=0}^{M} \Pr(i, 0, k-1) + p_c \Pr(M, 0, k)\Big] && \text{(5.4b)} \\
=& \frac{W_0 - j}{W_0} \Big[(1-p_c^{M+1})p_{\pi(k-1)} \Pr(0, 0, k-1) + p_c^{M+1} \Pr(0, 0, k)\Big] && \text{(5.4c)}
\end{aligned}
$$

where (5.4a) is due to the fact that state $(0, j, k)$ can only transit from states $(0, j+1, k)$, $(i, 0, k-1) \forall i = 0, \cdots, M$, and $(M, 0, k)$, and therefore can be written based on (5.1a), (5.1e) and (5.1d); (5.4b) is obtained by recursively substituting (5.4a) with incrementally increased j into the RHS of (5.4a); (5.4c) is because $\Pr(i, 0, k) = p_c^i \Pr(0, 0, k)$, as noted, and also $\sum_{i=0}^{M}(1-p_c)p_c^i = 1 - p_c^{M+1}$.

Let $j = 0$ in (5.4), we have

$$\Pr(0, 0, k) = (1-p_c^{M+1})p_{\pi(k-1)} \Pr(0, 0, k-1) + p_c^{M+1} \Pr(0, 0, k),$$

which can be reorganized as

.
$$\Pr(0, 0, k) = p_{\pi(k-1)} \Pr(0, 0, k-1), \text{ if } 1 < k < K \tag{5.5}$$

Substitute (5.5) into (5.4c). (5.4) can be rewritten as

$$\Pr(0, j, k) = \frac{W_0 - j}{W_0} \Pr(0, 0, k), \qquad 1 < k < K \tag{5.6}$$

Likewise, the steady probabilities of states $(0, j, K)$ can be given by

$$\begin{aligned}
\Pr(0, j, K) = &\Pr(0, j+1, K) + \frac{1}{W_0}\Big[(1-p_c)p_{\pi(K)}\sum_{i=0}^{M-1}\Pr(i,0,K) + (1-p_c)p_{\pi(K-1)}\times \\
&\sum_{i=0}^{M}\Pr(i,0,K-1) + \Big(1-(1-p_c)(1-p_{\pi(K)})\Big)\Pr(M,0,K)\Big] \qquad (5.7a)\\
=&\frac{W_0-j}{W_0}\Big((1-p_c)p_{\pi(K)}\sum_{i=0}^{M-1}\Pr(i,0,K) + \Big(1-(1-p_c)(1-p_{\pi(K)})\Big)\Pr(M,0,K) \\
&+(1-p_c)p_{\pi(K-1)}\sum_{i=0}^{M}\Pr(i,0,K-1)\Big) \qquad (5.7b)\\
=&\frac{W_0-j}{W_0}\Big((1-p_c^{M+1})p_{\pi(K-1)}\Pr(0,0,K-1)+ \\
&(p_{\pi(K)}+p_c^{M+1}-p_c^{M+1}p_{\pi(K)})\Pr(0,0,K)\Big), \qquad (5.7c)
\end{aligned}$$

where (5.7a) is due to the fact that state $(0, j, K)$ can only transit from states $(0, j+1, K)$, $(i, 0, K)$ for $i < M$, $(M, 0, K)$, and $(i, 0, K-1)$, $\forall i = 0, 1, \cdots, M$, and therefore can be obtained by using (5.1a), (5.1e), (5.1f), and (5.1g); (5.7b) can be obtained in the same way as (5.3c) and (5.4b); (5.7c) is obtained by using $\Pr(i, 0, k) = p_c^i \Pr(0, 0, k)$, as noted earlier.

Let $j = 0$ in (5.7), we have

$$\Pr(0,0,K) = \frac{p_{\pi(K-1)}}{1-p_{\pi(K)}}\Pr(0,0,K-1). \qquad (5.8)$$

Substitute (5.8) into (5.7c). (5.7) can be rewritten as

$$\Pr(0,j,K) = \frac{W_0-j}{W_0}\Pr(0,0,K). \qquad (5.9)$$

Furthermore, the steady probabilities of the remaining states $(0, j, 1)$ can be given by

$$
\begin{aligned}
\Pr(0,j,1) =& \Pr(0,j+1,1) + \frac{1}{W_0}\Bigg(\sum_{k=2}^{K}\sum_{i=0}^{M}\Pr(i,0,k)(1-p_c)(1-p_{\pi(k)}) \\
&+ \sum_{i=0}^{M-1}\Pr(i,0,1)(1-p_c)(1-p_{\pi(1)}) + \Pr(M,0,1)\Big(1-(1-p_c)p_{\pi(1)}\Big)\Bigg) \quad (5.10a)\\
=& \frac{W_0-j}{W_0}\Bigg(\sum_{k=2}^{K}\sum_{i=0}^{M}\Pr(i,0,k)(1-p_c)(1-p_{\pi(k)}) + \sum_{i=0}^{M-1}\Pr(i,0,1)(1-p_c)(1-p_{\pi(1)}) \\
&+ \Pr(M,0,1)\Big(1-(1-p_c)p_{\pi(1)}\Big)\Bigg) \quad (5.10b)\\
=& \frac{W_0-j}{W_0}\Bigg((1-p_c^{M+1})\sum_{k=2}^{K}\Pr(0,0,k)(1-p_{\pi(k)}) \\
&+ \Big(1-p_{\pi(1)}+p_c^{M+1}p_{\pi(1)}\Big)\Pr(0,0,1)\Bigg) \quad (5.10c)
\end{aligned}
$$

which, by referring to (5.5) and (5.8), can be further rewritten as

$$
\Pr(0,j,1) = \frac{W_0-j}{W_0}\Pr(0,0,1). \quad (5.11)
$$

here, (5.10a) is due the fact that state $(0,j,1)$ can only transit from states $(0,j+1,1)$, $(i,0,k)\forall i, k \neq 1$, $(i,0,1), i = 1,2,\cdots,M-1$, and $(M,0,1)$ at the probabilities given in (5.1a), (5.1b) and (5.1h); (5.10b) can be obtained in the same way as (5.3c) and (5.4b); (5.10c) is obtained by substituting $\Pr(i,0,k) = p_c^i \Pr(0,0,k)$ into the RHS of (5.10b).

According to the above derivation, the general expression for the steady probability of any state (i,j,k) can be given by

$$
\Pr(i,j,k) = \frac{W_i-j}{W_i}p_c^i\Pr(0,0,k), \quad (5.12)
$$

where $\Pr(0,0,k), k > 1$ can be recursively computed by

$$
\Pr(0,0,k) = \begin{cases} p_{\pi(k-1)}\Pr(0,0,k-1), \text{ if } k \neq K; & (5.13a)\\ \dfrac{p_{\pi(K-1)}}{1-p_{\pi(K)}}\Pr(0,0,K-1), k = K & (5.13b)\end{cases}
$$

Using (5.12), we have

$$\begin{aligned}\sum_{k=1}^{K}\sum_{i=0}^{M}\sum_{j=0}^{W_i-1}\Pr(i,j,k) =&\sum_{k=1}^{K}\sum_{i=0}^{M}\sum_{j=0}^{W_i-1}\frac{W_i-j}{W_i}p_c^i\Pr(0,0,k)\\ =&\sum_{k=1}^{K}\sum_{i=0}^{M}\frac{W_i+1}{2}p_c^i\Pr(0,0,k)\\ =&\sum_{k=1}^{K}\sum_{i=0}^{M}\frac{2^iW_0+1}{2}p_c^i\Pr(0,0,k)\\ =&\frac{1}{2}\Big(W_0\frac{1-(2p_c)^{M+1}}{1-2p_c}+\frac{1-p_c^{M+1}}{1-p_c}\Big)\sum_{k=1}^{K}\Pr(0,0,k).\end{aligned}$$

Note that $\sum_{k=1}^{K}\sum_{i=0}^{M}\sum_{j=0}^{W_i-1}\Pr(i,j,k)=1$. Therefore,

$$\sum_{k=1}^{K}\Pr(0,0,k)=\frac{2}{W_0\frac{1-(2p_c)^{M+1}}{1-2p_c}+\frac{1-p_c^{M+1}}{1-p_c}}. \tag{5.14}$$

The transmission probability τ is given by

$$\begin{aligned}\tau =&\sum_{k=1}^{K}\sum_{i=0}^{M}\Pr(i,0,k) &(5.15a)\\ =&\frac{(1-p_c^{M+1})}{1-p_c}\sum_{k=1}^{K}\Pr(0,0,k) &(5.15b)\\ =&\frac{2(1-p_c^{M+1})}{W_0(1-p_c)\sum_{i=0}^{M}(2p_c)^i-p_c^{M+1}+1}, &(5.15c)\end{aligned}$$

where (5.15a) is due to the fact that a (re)transmission takes place if and only if $j=0$; (5.15b) is because $\Pr(i,0,k)=p_c^i\Pr(0,0,k)$; and (5.15c) is by substituting (5.14) into (5.15b).

Also note that the collision probability p_c is given by

$$p_c=1-(1-\tau)^N. \tag{5.16}$$

As a result, p_c and τ can be obtained by jointly solving (5.15c) and (5.16). Both p_c and τ are independent of the key pool size K_P, the key ring sizes K_1 and K_2, as well as the key selection $\pi(1),\cdots,\pi(K)$. This concludes the proof of the theorem.

Remark: Theorem 1 can also be justified intuitively, since the message and signature/MAC are different under different keys. From a (re)transmission perspective, the different planes in Fig. 5.2 are identical, and can be compressed to be one plane (as developed in [16] without consideration on security). Even though τ and p_c are independent of K_P, K_1, K_2 and $\pi(k)$, $k = 1, \cdots, K$, and can be evaluated using the results of [16], nevertheless, the key metrics of our authentication design, such as authentication rate, delay and throughput, do depend on $\pi(1), \cdots, \pi(K)$. The proposed Markov model is important to evaluate the metrics, as follows.

5.3.1 Authentication Success Rate

The authentication success rate, denoted by P_A, defines the ratio of successfully authenticated messages to the total messages transmitted. P_A can be readily inferred using the proposed 3D Markov model, as given by

$$P_A = \sum_{k=1}^{K} \left(\frac{k}{\tau} \sum_{i=0}^{M} \Pr(i, 0, k)(1 - p_c^{M+1})(1 - p_{\pi(k)}) \right) \tag{5.17}$$

where $\frac{1}{\tau} \sum_{i=0}^{M} \Pr(i, 0, k)$ is the transmission probability per timeslot under $\pi(k)$, given τ. $(1 - p_c^{M+1})$ is the probability that at least one (re)transmission under $\pi(k)$ is not collided. $(1 - p_{\pi(k)})$ is the probability that the collision-free (re)transmission is authenticated with the matched key (i.e., k messages are all authenticated).

5.3.2 Authentication Delay

The average per-message delay of successfully authenticated messages, denoted by υ_A, can be given by

$$\upsilon_A = \frac{1}{P_A} \sum_{k=1}^{K} \left(k \sum_{i=0}^{M} \frac{\upsilon_{k,i} + \sum_{k'=1}^{k-1} \upsilon_{k'}}{k} \times \right. \tag{5.18a}$$

$$\left. \frac{1}{\tau} \Pr(i, 0, k) \big(1 - p_c^{M+1}\big) \big(1 - p_{\pi(k)}\big) \right), \tag{5.18b}$$

where $\frac{\upsilon_{k,i} + \sum_{k'=1}^{k-1} \upsilon_{k'}}{k}$ is the average authentication delay of k messages. Specifically, $\upsilon_{k'}$ $(k' = 1, \cdots, k-1)$ is the average delay resulting from the failed verification of $\pi(k')$, i.e., the average delay on the k'-th plane of the proposed 3D Markov model. $\upsilon_{k,i}$ is the average delay of the successful verification of $\pi(k)$ till the i-th (re)transmission under $\pi(k)$. $\frac{1}{P_A}$ normalizes the delay over successfully authenticated messages only. (5.18b) gives the probability that $\pi(k)$ is successfully verified at the i-th (re)transmission, after failed verifications of the previous $(k-1)$ keys $\pi(k')$, $k' = 1, \cdots, k-1$; see (5.17).

Two types of timeslot with different durations are meticulously considered to evaluate υ_k: mini-slot with duration of υ_m (during which no nodes transmit), and transmission slot with duration of υ_T (during which at least one node transmits). $0 < \upsilon_m \ll \upsilon_T$.

Exploiting the proposed 3D Markov model, $\upsilon_{k,i}$ and $\upsilon_{k'}$ ($k' = 1, \cdots, k-1$) can be given by

$$\upsilon_{k,i} = \sum_{i'=0}^{i} \left(\frac{W_{i'} - 1}{2} \Big(\upsilon_m (1-\tau)^N + \upsilon_T \big(1 - (1-\tau)^N\big) \Big) + \upsilon_T \right) + \upsilon_c \tag{5.19a}$$

$$\upsilon_{k'} = \sum_{i'=0}^{M} \left(\frac{p_c^{i'}(1-p_c)}{1-p_c^{M+1}} \upsilon_{k',i'} \right) \tag{5.19b}$$

where $\frac{W_{i'}-1}{2}$ is the average number of timeslots backed off before the i'-th retransmission. $\Big(\upsilon_m(1-\tau)^N + \upsilon_T\big(1-(1-\tau)^N\big)\Big)$ is the average duration of the timeslots. Specifically, $(1-\tau)^N$ is the probability of a mini-slot, during which no node (incl. the transmitter) transmits; $\big(1-(1-\tau)^N\big)$ is the probability of a transmission slot, during which at least one node, other than the transmitter, transmits. Clearly, the average duration of a timeslot and the average number of timeslots are independent, and can be multiplied for the average delay before the i'-th (re)transmission under $\pi(k')$. $\frac{p_c^{i'}(1-p_c)}{1-p_c^{M+1}}$ is the probability that the i'-th retransmission is collision-free while the earlier (re)transmissions have all been collided.

In (5.19), υ_c is the average delay in the case where all the $(M+1)$ (re)transmissions under a key are collided. In this case, the proposed 3D Markov model remains on the same plane until a (re)transmission is collision-free. Then, the model proceeds to the first plane, if the (re)transmission is authenticated; or the next plane, otherwise. υ_c is given by

$$\begin{aligned}\upsilon_c =&\sum_{c'=0}^{\infty}(p_c^{M+1})^{c'}c'\sum_{i'=0}^{M}\left(\frac{W_{i'}-1}{2}\Big(\upsilon_m(1-\tau)^N+\upsilon_T\big(1-(1-\tau)^N\big)\Big)+\upsilon_T\right)\\ =&\frac{p_c^{M+1}}{(1-p_c^{M+1})^2}\sum_{i'=0}^{M}\left(\frac{W_{i'}-1}{2}\Big(\upsilon_m(1-\tau)^N+\upsilon_T\big(1-(1-\tau)^N\big)\Big)+\upsilon_T\right)\end{aligned} \tag{5.20}$$

where $c' = 0, 1, \cdots$ indicates the number of the times that the Markov model repeats on a plane, due to collisions; $(p_c^{M+1})^{c'}$ is the possibility of c' repetitions on the plane; $\sum_{i'=0}^{M}\left(\frac{W_{i'}-1}{2}\Big(\upsilon_m(1-\tau)^N+\upsilon_T\big(1-(1-\tau)^N\big)\Big)+\upsilon_T\right)$ is the average delay of a repetition, as discussed in (5.19); $\sum_{c'=0}^{\infty}(p_c^{M+1})^{c'}c' = \frac{p_c^{M+1}}{(1-p_c^{M+1})^2}$.

5.3.3 Authenticated Throughput

The average throughput of successfully authenticated messages, denoted by T_A, can be written as

$$\begin{aligned}T_A =& \frac{\sum_{k=1}^{K}\Big(k\sum_{i=0}^{M}\Pr(i,0,k)(1-p_c)(1-p_{\pi(k)})\Big)L_{pkt}}{\upsilon_m(1-\tau)^{N+1}+\upsilon_T\big(1-(1-\tau)^{N+1}\big)}\\ =& \frac{\tau P_A(1-p_c)/(1-p_c^{M+1})L_{pkt}}{\upsilon_m(1-\tau)^{N+1}+\upsilon_T\big(1-(1-\tau)^{N+1}\big)},\end{aligned} \tag{5.21}$$

where L_{pkt} is the number of bits per message; $\sum_{k=1}^{K}\Big(k\sum_{i=0}^{M}\Pr(i,0,k)(1-p_c)(1-p_{\pi(k)})\Big)$ provides the expected number of messages successfully authenticated per timeslot; and $\upsilon_m(1-\tau)^{N+1}+\upsilon_T\big(1-(1-\tau)^{N+1}\big)$ provides the average duration of a timeslot.

5.4 Resistance Analysis against Collusion Attacks

In this section, the resistance or robustness of the proposed protocol is evaluated, when the key pairs are increasingly revoked under a series of collusion attacks. In collusion attacks, adversaries physically capture legitimate nodes. The captured nodes (or internal adversaries) use their predistributed legitimate keys to sign or encrypt falsified messages, and send the messages to other legitimate nodes. Key revocation is widely used to defend against collusion attacks [53]. The keys, used to sign or encrypt falsified messages, are revoked and removed from all the nodes, provided that the falsification of messages can be detected using false information detection techniques [47].

To evaluate the resistance, we first derive the probability mass function (PMF) of the number of revoked keys, κ, on condition of n captured and compromised nodes. For illustration convenience, the discussion here is on asymmetric keys, particularly on the numbers of revoked private and public keys under collusion attacks. The discussion applies to symmetric keys though, where we can evaluate the number of revoked symmetric keys in the same way as we evaluate that of revoked private or public keys.

In the case of asymmetric keys, the conditional PMF of the number of revoked private keys can be recursively given by

$$\Pr(\kappa|n) = \sum_{\forall \kappa' \leq \kappa} \left(\Pr(\kappa'|n-1) \frac{\binom{\kappa'}{K_1-\kappa+\kappa'}\binom{K_P-\kappa'}{\kappa-\kappa'}}{\binom{K_P}{K_1}} \right), \tag{5.22}$$

which has been explained in (4.19) in Section 4.3.5. Note that $\Pr(K_1|n=1)=1$ and $\Pr(\kappa|n=1)=0$ for any $\kappa \neq K_1$, since every node is predistributed K_1 private keys. (5.22) can be calculated in recurrence.

Given κ, the PMF of the number of the remaining, non-compromised private keys per legitimate node, denoted by γ_1, can be given by

$$\Pr(\gamma_1|\kappa) = \binom{K_P-\kappa}{\gamma_1}\binom{\kappa}{K_1-\gamma_1} \Big/ \binom{K_P}{K_1}, \tag{5.23}$$

where K_1 can be replaced by K_2 to achieve the PMF of γ_2, i.e., the number of the remaining, non-compromised public keys per legitimate node.

Provided n, the PMF of γ_i $(i=1,2)$, can be obtained by substituting (5.22) and

(5.23) into the following.

$$\Pr(\gamma_i|n) = \sum_{\forall\kappa\leq nK_1,K_P} \Pr(\gamma_i|\kappa)\Pr(\kappa|n). \tag{5.24}$$

To this end, γ_1 and γ_2 are the respective effective key ring sizes of private and public keys per legitimate node, after n nodes are compromised under collusion attacks

As dictated in Theorem 1, τ and p_c are independent of the key ring sizes and the key selection at individual nodes. In other words, they are unaffected by the shrinking key rings. However, the shrinking key rings do affect the authentication performance – authentication success rate, delay and throughput, because $p_{\pi(k)}$ in (5.17), (5.18) and (5.21) depends on the key ring sizes of the public keys, as shown in (5.2).

Given that n nodes are captured under collusion attacks, we can rewrite (5.2) as

$$p_{\pi(k)}(\gamma_2^1,\cdots,\gamma_2^N|n) = \prod_{i=1}^{N} \frac{\binom{K_P-k}{\gamma_2^i}}{\binom{K_P-k+1}{\gamma_2^i}}, \tag{5.25}$$

where γ_2^i denotes the public key ring size at node $i = 0,1,\cdots,N$, and node 0 is the transmitter. The corresponding conditional probability $\Pr(\gamma_1^1,\cdots,\gamma_1^N|n)$ can be obtained by substituting (5.24), as given by

$$\Pr(\gamma_2^1,\cdots,\gamma_2^N|n) = \prod_{i=1}^{N} \Pr(\gamma_2^i|n). \tag{5.26}$$

Finally, the authentication success rate, delay and throughput, after a series of collusion attacks, can be obtained by substituting (5.25) into (5.17), (5.18) and (5.21), and taking means over all possible $(\gamma_2^1,\cdots,\gamma_2^N)$ using (5.26).

5.5 Numerical Result

In this section, Monte-Carlo simulations are carried out to evaluate the proposed authentication process and validate the analysis. Either symmetric or asymmetric keys can be used in the proposed protocol. In the case of symmetric keys, MACs are generated, as done in μTESLA [70]. Take HMAC-SHA1 for example, the length of a key is 14 bytes, and the length of a MAC is 20 bytes [70]. The number of keys predistributed per node is set to be from $0.5\% K_P$ to $15\% K_P$. In the case of asymmetric keys, a signature scheme is used. Take ECDSA [70] for example, the length of a public key is 28 bytes, the length of a private key is 28 bytes, and the length of a signature is 56 bytes. The number of private keys per node is set to be $K_1 = 20$, and the number of public keys per node ranges from $0.5\% K_P$ to $15\% K_P$. Other settings of the simulations are provided in Table 5.1.

For comparison purpose, this section simulates the piggyback approach proposed in [100], where a public key and its digital certificate are sent together with messages and signatures every time the topology changes. In the rest of the time, only messages and signatures are sent, since the neighbours can use most recently received public key to verify the signatures while the topology remains temporarily stable. The length of the digital certificate is 56 bytes [100], apart of the public key and signature specified above.

Table 5.1 : Simulation Parameters

	Description	Value
M	maximum number of retransmissions per message	6
K	maximum number of keys that can be tried	5
W_0	initial backoff window size	16
K_P	key pool size	$10^4, 3 \times 10^4$
R	channel bandwidth	6 Mbps
L_{pkt}	payload per message	$100 \sim 4000$ bytes
L_{id}	the length of key identity	2 bytes

TSVC [70] is also simulated. TSVC is based on μTESLA and uses symmetric keys to generate MACs. Particularly, TSVC can have a MAC and a symmetric key to transmit together with a message. The symmetric key is updated every message. Exploiting the idea of piggyback, every time the topology changes, a signature signed on the symmetric key by using a unique private key, the corresponding public key, and the certificate of the public key need to be transmitted apart from a message, MAC and symmetric key, such that the symmetric key can be validated. Other symmetric keys used later can be validated by hashing and comparing with this symmetric key.

The piggyback [100] and TSVC [70] approaches provide the upper bound for authentication success rate, since every collision-free message can be authenticated and the success rate only depends on the collision probability.

The EG scheme [38] is also applied straightforwardly for authentication, where it is assumed in favor of the scheme that handshaking between neighbouring nodes is accomplished offline and the communication overhead and delay of deciding on the common keys are overlooked. Particularly, both symmetric and asymmetric keys are considered.

Fig. 5.3 plots the authentication success rate of the proposed protocol P_A, and validates the proposed 3D Markov at fine accuracy. Both the symmetric and asymmetric keys provide the same results for the proposed protocol, since their difference lies in the communication overhead from a protocol point-of-view. We see that P_A exhibits concavity under the proposed protocol. Before the peaks, the limited (public) keys, predistributed among a small number of neighbours, has the dominating impact on P_A. The (public) keys and consequently P_A increase with N. After the peaks, intensive collisions caused by many neighbours dominate, and P_A declines as N grows. We also see that the proposed protocol converges to the upper bound

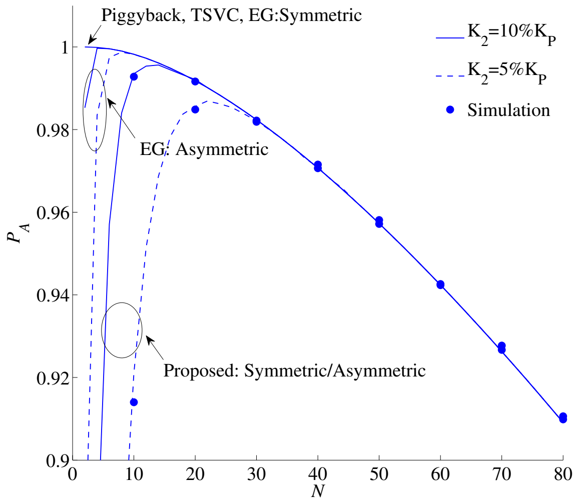
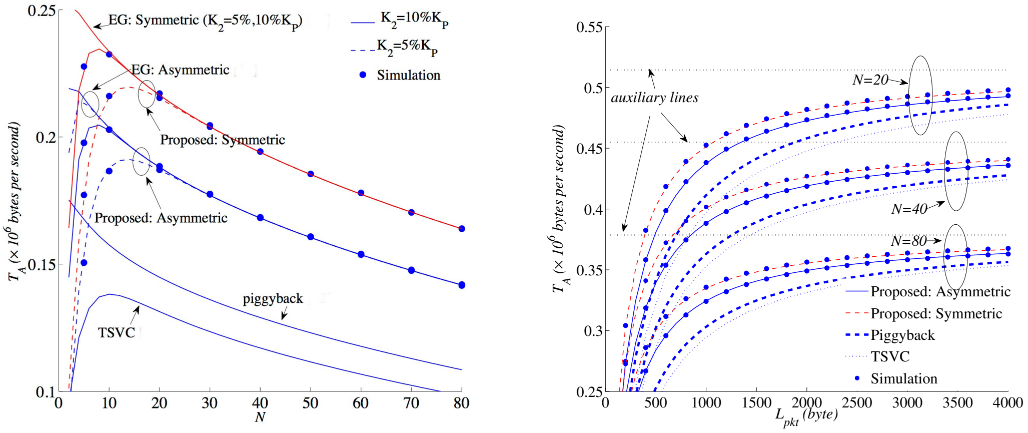


Figure 5.3 : Authentication success rate versus N, where $K_2 = 5\%K_P$ and $10\%K_P$, $K_P = 10^4$ and $K_1 = 20$.



(a) $K_1 = 20$, and $L_{\text{pkt}} = 100$ bytes

(b) $K_1 = 20$, and $K_2 = 5\%K_P$

Figure 5.4 : Comparison of the authenticated throughput, where $K_P = 10^4$.

provided by the piggyback and TSVC approaches. The conclusion drawn is that the authentication success is dominated by the collisions of uncoordinated transmissions in dense networks.

Keep in mind that in Fig. 5.3 the high communication overhead is unaccounted for in the existing piggyback and TSVC approaches, while the overhead of handshaking is overlooked in the EG scheme. The evaluation of authentication success rate cannot capture the requirement of communication overhead. Fig. 5.4 proceeds to compare the proposed protocol and these approaches in a fair fashion, from the aspect of authenticated throughput.

Fig. 5.4(a) shows that each of the protocols loses its throughput increasingly, when N is large. This is due to the intensifying collisions. The proposed protocol can consistently outperform the piggyback and TSVC approaches, despite the latter provides the upper bound for the authentication success rate. This is the result of the much higher overhead in the piggyback and TSVC approaches than in the proposed protocol, while the difference of the protocols in authentication success rate is indistinguishable (as shown in Fig. 5.3). For the same reason, the use of symmetric keys can outperform that of asymmetric keys, when both exploit the proposed protocol.

Fig. 5.4(b) displays that the throughput gain of the proposed protocol over the piggyback and TSVC approaches enlarges, until L_{pkt} is large (the communication overhead of the piggyback and TSVC approaches become relatively negligible) and the gain starts to diminish. The figure also reveals the monotonic increase of the authenticated throughput, as the payload L_{pkt} (or in other words, υ_T) grows. Particularly, as $\frac{\upsilon_m}{\upsilon_T} \to 0$, the throughput of the proposed protocol asymptotically converges to $\frac{\tau(N+1)P_A R}{(1-(1-\tau)^{N+1})} \frac{(1-p_c)}{(1-p_c^{M+1})}$ (plotted as auxiliary horizontal dashed lines); see (5.21).

Fig. 5.5 evaluates the impact of K_2 on the authenticated throughput of the proposed protocol, where only the case of asymmetric keys is plotted for clarity purpose. An observation is that the throughput grows and stabilizes quickly, as K_2 increases. Typically, less than $10\% K_P$ is sufficient to stabilize the throughput. The more the neighbours are, the smaller value of K_2 is needed to stabilize the

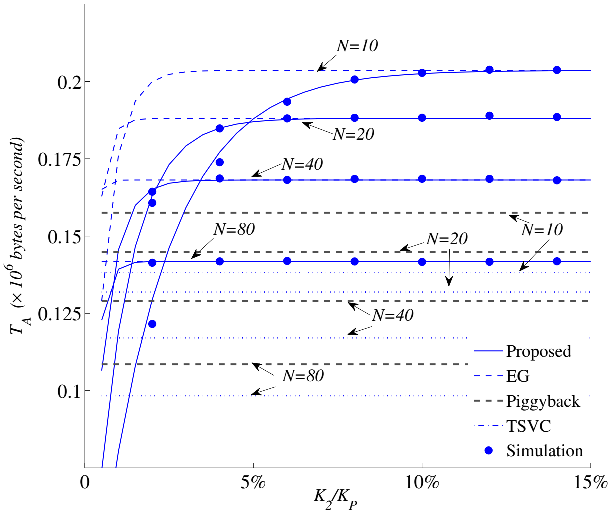


Figure 5.5 : Authenticated throughput versus the number of predistributed keys per node K_2, where $N = 10, 20, 40, 80$, $K_P = 10^4$ and $L_{\text{pkt}} = 100$ bytes.

throughput. This is due to the increased probability that a key is predistributed among the neighbours. Only a very small number of symmetric keys or asymmetric public keys is needed for the proposed protocol to outperform the piggyback and TSVC approaches, e.g., $K_2 = 2\% K_P$ for $N = 40$.

It is also seen in Fig. 5.5 that the proposed protocol achieves the same stabilized throughput as the EG scheme ideally could. With the keys agreed offline between neighbours, the idealized EG scheme here provides the highest throughput for the case where asymmetric keys are predistributed. The EG scheme is also shown to require less numbers of keys predistributed to stabilize the throughput. Both of these are due to the ideal assumption that the communication overhead of the handshaking in the EG scheme is overlooked here.

Fig. 5.6 compares the authenticated throughput of the proposed protocol, the EG scheme, the piggyback and TSVC approaches against the authentication delay per message in mobile environments in a fair manner, where each curve is plotted

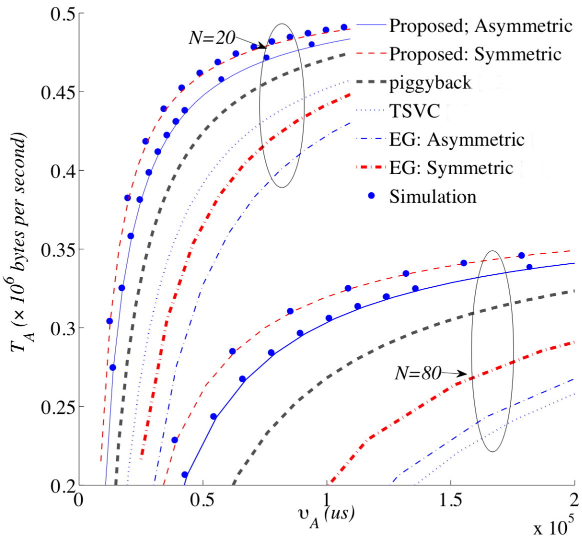


Figure 5.6 : Throughput versus per-message delay, where $K_P = 10^4$, $K_2 = 5\%K_P$, L_{pkt} increases from 100 to 4000 bytes.

by varying the payload size. For the EG scheme, the delay of handshaking between neighbouring nodes is taken into account. Now, the figure shows that the proposed protocol is superior to the EG scheme in practice. This is due to the handshaking delay of the EG scheme. Our protocol is also better than the piggyback and TSVC approaches. As aforementioned, the average per-message authentication delay caused by collided (re)transmissions and key mismatches is proved to be much higher than the time required to verify a MAC or a signature of typically 1 μs or 4 ms [21], especially in dense mobile networks.

Fig. 5.7 analyses the authenticated throughput, as the interval of the topology change increases. The figure is plotted by varying K from 1 to 30, since K, designed based on the topology change interval, is indicative of the mobility of the network. It is shown that the proposed protocol can tolerate far more frequent topology changes than the EG, the piggyback and TSVC approaches. For topology changes at an interval of 1 second, the proposed protocol can provide 20% higher throughput than

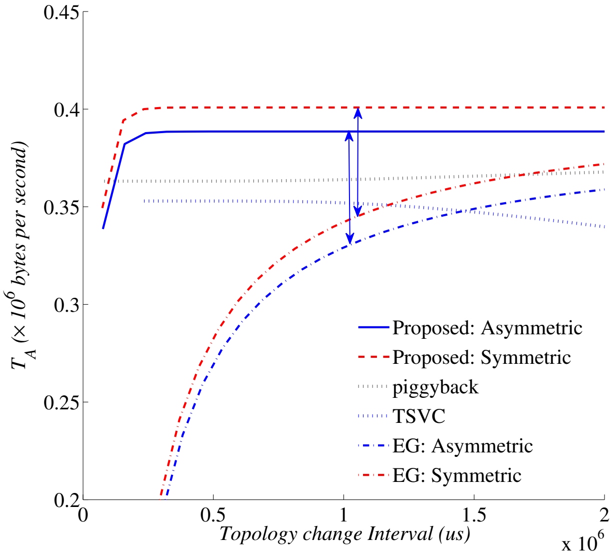
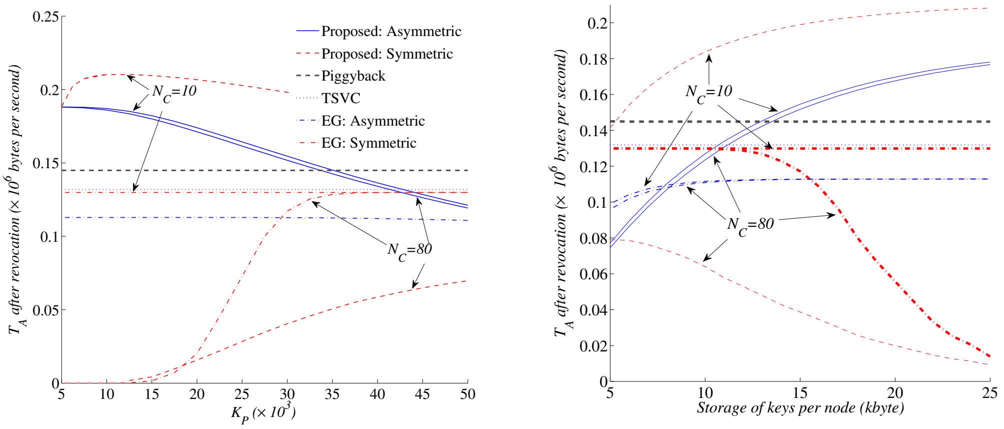


Figure 5.7 : Authenticated throughput versus topology interval, where $K_P = 10^4$, $K_2 = 5\%K_P$, $N = 40$, and $L_{\text{pkt}} = 1000$ bytes.

the EG scheme. In both Figs. 5.6 and 5.7, we see that symmetric keys can support higher throughput than asymmetric keys, under the proposed protocol. This is due to the lower communication overhead that a MAC requires than a signature.

Further, Figs. 5.8 and 5.9 assess the proposed protocol under collusion attacks. Fig. 5.8 analytically evaluates the impact of the entire key pool size K_P and the storage space per node on the robustness against collusion attacks, where N_C is the number of compromised nodes in the network. The analytical results of (5.21) and (5.25) are plotted in the figure. It is shown that given the storage space for the keys, the authenticated throughput of the proposed protocol first grows and then declines as K_P increases, as shown in Fig. 5.8(a). This is because when K_P is small, the proposed protocol is susceptible to collusion attacks. Increasing K_P helps reduce the ratio of the keys exposed and revoked after collusion attacks, resulting in a throughput increase. When K_P is large, increasing K_P leads to a reduction of the ratio of the keys predistributed per node against K_P. In turn, the probability



(a) The key storage per node is 15 kbytes.

(b) $K_P = 3 \times 10^4$.

Figure 5.8 : Authenticated throughput in the presence of collusion attacks, where $N = 20$, $N_C = 10, 80$, and $L_{\text{pkt}} = 100$ bytes.

of lack of matched keys between neighbours increases; in other words, the network connectivity decreases.

Symmetric keys are vulnerable to collusion attacks, as a large number of keys used to produce MACs can be exposed and revoked, compromising network connectivity. In contrast, asymmetric keys are robust against collusion attacks, since only a very small number of private keys used to produce signatures are exposed and revoked. The network connectivity and subsequently the throughput are little affected, e.g., when N_C increases from 10 to 80. For the same reason, symmetric keys are sensitive to the storage size for the keys; while asymmetric keys are far more tolerant, as shown in Fig. 5.8(b). Using asymmetric keys, the proposed protocol can outperform the EG, piggyback and TSVC approaches, when K_P is less than 5×10^4, 3.2×10^4 and 4.5×10^4, respectively, as shown in Fig. 5.8(a); or when the key storage is over 13 kbyptes per node, as shown in Fig. 5.8(b). Note here that the EG scheme performs handshaking to decide on the keys between neighbours and therefore depends little on K_P.

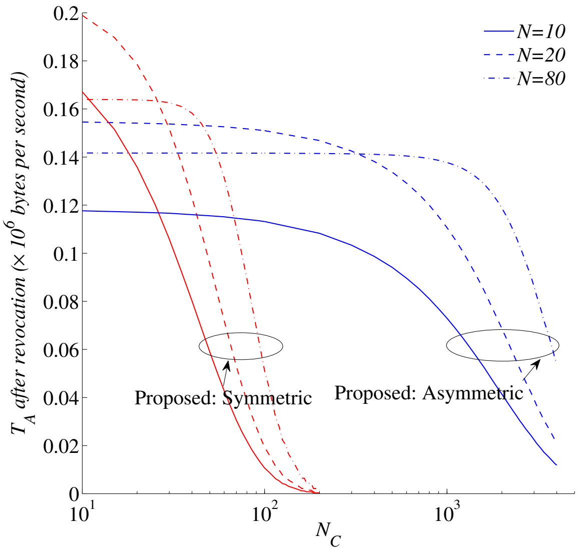


Figure 5.9 : Comparison between the uses of symmetric and asymmetric keys in terms of robustness against collusion attacks, where the storage per node for the keys is 15 kbytes, $K_P = 3 \times 10^4$, and $L_{\text{pkt}} = 100$ bytes.

Finally, Fig. 5.9 compares the uses of symmetric and asymmetric keys in the proposed protocol in terms of robustness against collusion attacks. It is confirmed that asymmetric keys can substantially improve the robustness of the proposed protocol against the collusion attacks, as discussed above. Nevertheless, symmetric keys can provide much higher throughput in the case that the attacks are not severe. This is due to the fact that MACs produced by symmetric keys require much less communication overhead than signatures produced by asymmetric keys.

5.6 Summary

In this chapter, an opportunistic authentication protocol is proposed for mobile IoT networks with fast changing topologies. The protocol, general to symmetric and asymmetric keys, is coupled with opportunistic routings to ensure data integrity in a collaborative manner. A new 3D Markov model was developed to characterize the interaction between transmission collisions and key mismatches in the networks. Closed-form expressions for authentication rate, delay and throughput were derived. Validated by simulations, analytical results corroborate the robustness of the proposed protocol against changing topologies, as well as substantially improved resistance to collusion attacks.

Chapter 6

Impact of Link Duration on the Integrity of Distributed Mobile Networks

6.1 Introduction

Focussing on the impact of short link duration on authentication performance in mobile IoT networks, this chapter develops a unified analytic framework, a new four-dimensional (4D) model, to characterize an on-the-fly authentication process, where a receiver can be valid for only a short duration and replaced frequently as the result of mobility. The first three dimensions capture a cycle of channel access and opportunistic authentication until either is the leading message of the cycle authenticated or exhausts the largest backoff window or the maximum number of keys allowed. The fourth dimension is proposed to characterize the transitions between cycles, unprecedentedly capturing any unexpired and unauthenticated messages carried on from earlier cycles.

Based on the new 4D model, three opportunistic authentication protocols are designed and analysed. For the opportunistic authentication protocols, keys are pre-distributed across the network and a general location-aware routing is considered, where a link remains valid only for a short period of time [52]. Delivered, unexpired but unauthenticated messages within a link duration can all be authenticated retrospectively at the receiver, if the receiver is preloaded with the key matching the one the transmitter has adopted. The three opportunistic authentication protocols couple on-the-fly authentication and channel access to different extents, for the purposes of (a) compliance with the standard channel access model, namely, distributed coordination function (DCF); (*b*) improving authenticated throughput; and (*c*) reducing authentication delay.

Validated by simulations, the unified modeling framework is able to compare a variety of different protocol designs of on-the-fly authentication. Interesting findings include that the authenticated throughput of the opportunistic protocols can asymptotically approach the unauthenticated throughput of DCF, while still complying with DCF. The cost of authentication diminishes with the growth of link duration. It is also found that the protocols are particularly sensitive to transmission collisions in distributed networks. Although on-the-fly authentication can significantly improve the authentication rate, reduce authentication delay and enhance scalability to dense mobile distributed networks, compared to the prior art, the analysis reveals that cross-layer consideration to jointly design retransmissions and rekeying is the key to achieve the significant gain of on-the-fly authentication in distributed mobile IoT networks.

Note that the analysis applies to any mobile models which study topology changes. The worst-case delay of successful authentication is used to evaluate the tolerance of authentication protocols against topology changes or network mobility. This is consistent with common mobility models, such as random walk and random waypoint, which typically derive link duration specifying the time that a network remains stable in terms of topology.

The rest of this chapter is organized as follows. In Section 6.2, a general description of the on-the-fly authentication is presented, followed by the proposed 4D Markov model in Section 6.3. Three representative authentication protocols are analysed in Section 6.4. In Section 6.5, numerical examples are provided, followed by the summary in Section 6.6.

6.2 On-the-fly Authentication Protocol

6.2.1 Network Setup

The scenario considered in this chapter is that a node sends messages towards another node at a specific location. Location-aware routing methods [15, 84] can be used to identify the receiver. By considering the mobility of the nodes, the desired receiver at the location of interest may remain valid for a limited period of time, after which a new node moves in and becomes the receiver. This period of time is referred as "link duration", denoted by τ_{link} [132]. It is crucial to get a message delivered to and verified at the receiver within τ_{link}, preserving the relevance and timeliness of the receiver. Without loss of generality, we encompass on a transmitter and an intended location, to which the closest node becomes the receiver, and assume N active nodes within the transmission range of the transmitter, including the receiver. It is also assumed that the network is saturated, i.e., the nodes always have messages to send. The assumption is reasonable in dense networks with limited bandwidths, as queues can build up at nodes [113, 59].

A pool of K_{P} key pairs are off-line generated and predistributed among all nodes through secure channels [38, 23]. Every node is preloaded with K_{pub} randomly selected public keys and K_{pri} randomly selected private keys. Cautious of collusion attacks, K_{pri} is typically set to be far smaller than K_{P}. The transmitter can randomly pick up a private key to sign a message before sending it. The receiver is able to authenticate the message instantly if equipped with the corresponding public key. As a result, there is no need for piggybacking the public key and the certificate in the transmission [100], or for handshaking between the transmitter and receiver to decide on common key pairs [38], thereby avoiding severe overhead and delays. Notations used in this chapter are explained in Tab. 6.1.

Table 6.1 : Notation and Definition

Notation	Definition
W_0, W_M	the minimum/maximum backoff windows
τ_{link}	link duration
σ_0	mini-slot time
K_{P}	the size of the key pool
$K_{\text{pub}}/K_{\text{pri}}$	the number of public/private keys per node
$\Pr(i, j, k)$	the steady-state probability of state (i, j, k) in the first three dimensions
K	the maximum number of different private keys used per cycle
λ_{s}	the steady-state probability of a successful cycle Ω_{s}
λ_{k}	the steady-state probability of an unsuccessful cycle resulting from key failures, denoted by Ω_{k}
λ_{c}	the steady-state probability of a collided cycle, i.e., an unsuccessful cycle resulting from unresolved collisions, denoted by Ω_{c}
p_c	the collision probability per slot
τ	the transmission probability of a node per slot
p_{π_k}	the probability that the k-th selected key of a cycle fails to be matched at the receiver
T_A	the authenticated throughput
$\overline{D}$	the average authentication delay
D^{wst}	the worst-case authentication delay
L_{pkt}	the size of packet payload

6.2.2 Communication and Authentication

As illustrated in Fig. 6.1, authentication can be carried out opportunistically in coupling with CSMA/CA based channel access, e.g., DCF, in distributed mobile IoT networks, given the predistributed keys. The different protocol designs shown in Fig. 6.1 lie in different responses to failed authentications, detailed descriptions of which will be provided in Section 6.4. Due to the cyclic networks of exponential backoff in CSMA/CA, all the authentication designs run on the basis of cycles. At the beginning of every cycle, the transmitter signs a new message with one of its preloaded private keys π_1, sets a random timer j within the initial backoff window W_0, and counts down the timer (at an interval of a mini-slot). Exploiting CSMA/CA, the transmitter keeps sensing the channel. It stops counting if the channel is busy, and resumes only after the channel is free again. The transmission of the message and signature is triggered, once the timer j becomes zero.

In the case that the message is received collision-free and verified with the matched public key, the receiver returns ACK and the transmitter proceeds with a new message to start a new cycle with an initial backoff window, W_0, and key selections. In the case that the message collides with those of other nodes, no ACK returns from the receiver. The transmitter backs off and retransmits the message and signature by doubling the backoff window, i.e., $W_i = 2W_{i-1}$ $(i = 1, \cdots, M)$, and resetting a new random timer j within W_i. W_M is the largest backoff window, after which the transmitter restarts a new cycle by resetting the backoff window and key selection, and proceeds with a new message.

In the case that the current message is received collision-free but unauthenticated due to the lack of the corresponding public key, the receiver returns a non-acknowledgement (NACK). The transmitter then chooses another private key, π_k, from its key ring, provided the previous $(k-1)$ keys, $\pi_{k'}, k' = 1, \cdots, k-1$ in this cycle all fail unmatched at the receiver.

We can design that the transmitter sends a new message each time the key is changed. The signature is updated by applying π_k to the hash of the new message and earlier collision-free yet unauthenticated messages which remain unexpired in the current link duration. If the receiver is predistributed with the corresponding public key of π_k and the (re)transmission is collision-free, the unexpired messages can be successfully authenticated altogether. Up to $K \leq K_{\text{pri}}$ private keys can be selected per cycle. The length of the signature remains unchanged, due to the fact that the output of a hash function has a consistent length. After K private keys fail unmatched at the receiver, the transmitter restarts a new cycle with W_0 and key selection. M and K can be meticulously selected to be accommodated with τ_{link}, as shown in Section 6.4.

6.3 Proposed 4D Markov Model

In this section, the new 4D model is developed to characterize an ongoing on-the-fly authentication process, where a receiver can be valid only for a short duration and replaced frequently as the result of mobility. The first three dimensions capture a cycle of channel access and opportunistic authentication. A cycle starts with a new message under the smallest backoff window and a newly selected private key. It allows the transmitter to change the key if a message transmission is collision-free but unauthenticated, or enlarge the backoff window if a transmission is collided; until either is the message authenticated, or exhausts the largest backoff window or the maximum number of keys allowed. The fourth dimension of the 4D model characterizes the transitions between cycles, capturing any unexpired and unauthenticated messages carried on from earlier cycles within a link duration τ_{link}. Fig. 6.2 presents the transitions of the proposed 4D Markov model, where the left-hand side captures the state transitions within an authentication cycle, and the overall state

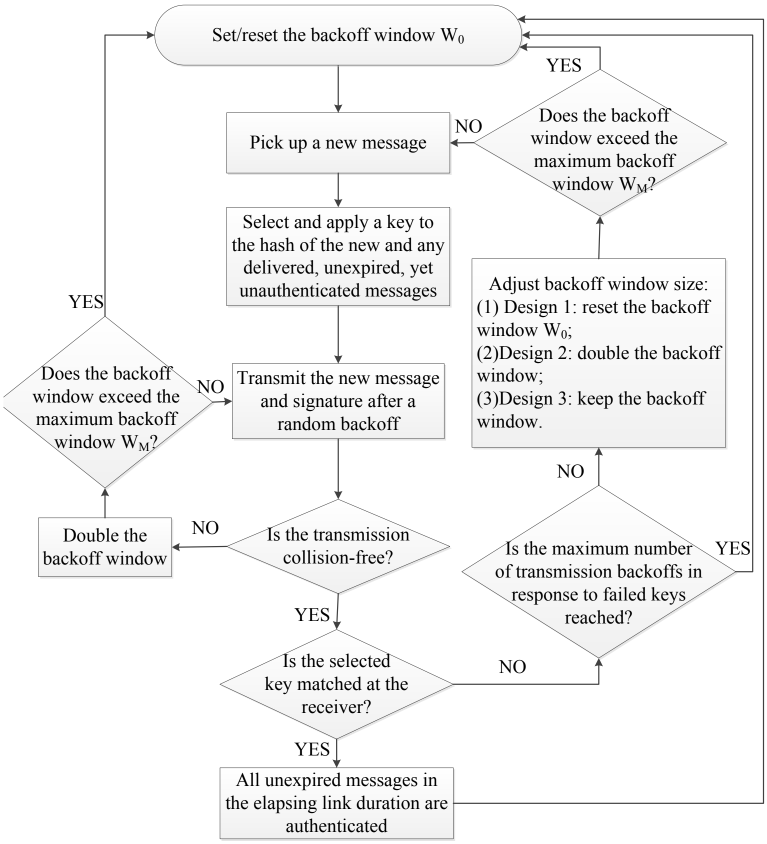

Set/reset the backoff window W0
Pick up a new message
Select and apply a key to the hash of the new and any delivered, unexpired, yet unauthenticated messages
Transmit the new message and signature after a random backoff
Is the transmission collision-free?
Double the backoff window
Does the backoff window exceed the maximum backoff window WM?
YES
NO
Is the selected key matched at the receiver?
All unexpired messages in the elapsing link duration are authenticated
Is the maximum number of transmission backoffs in response to failed keys reached?
Adjust backoff window size:
(1) Design 1: reset the backoff window W0;
(2)Design 2: double the backoff window;
(3)Design 3: keep the backoff window.
Does the backoff window exceed the maximum backoff window WM?

Figure 6.1 : The general flowchart of on-the-fly authentication protocols.

transitions between cycles are depicted on the right-hand side.

6.3.1 Modeling of an Authentication Cycle

The first three dimensions of the proposed model characterize the interactions between collisions, (re)transmissions, and key selections within an authentication cycle. An authentication cycle starts by selecting π_1 and setting the backoff window W_0, and ends if π_k, $k \leq K$ is matched at the receiver, referred to as a "successful cycle" or $\Omega_{\rm s}$; or the maximum number of K keys all fail unmatched, referred to as an "unsuccessful cycle" or $\Omega_{\rm k}$; or the maximum backoff window W_M is insufficient and a larger window would be required, referred to as a "collided cycle" or $\Omega_{\rm c}$.

Each state of the first three dimensions, denoted by triplet $(i, j, k), 0 \leq i \leq$

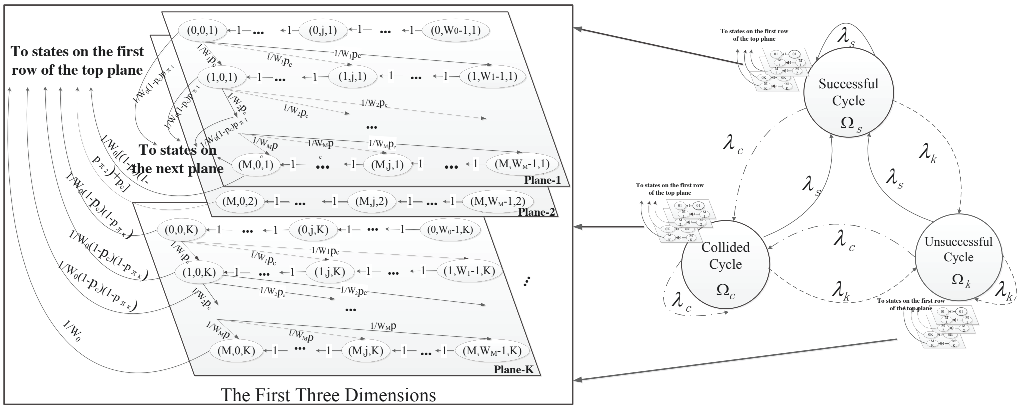


Figure 6.2 : Illustration on the proposed 4D Markov model

$M; 0 \leq j \leq W_i - 1; 1 \leq k \leq K$, indicates the j-th slot within W_i to elapse until the upcoming (re)transmission of the k-th unauthenticated message in this cycle, provided none of the preceding $(k-1)$ keys are matched at the receiver. Particularly, the k-th plane (from top) corresponds to the (re)transmissions of the k-th unauthenticated message in the current cycle and the signature. The signature is signed by π_k on the hash of the k messages in the current cycle and earlier unexpired (collision-free but unauthenticated) messages from the preceding cycles within the current link duration (see descriptions in Section 6.2.2). Horizontally connected states describe the countdown of the backoff timer, j. A (re)transmission of the designated transmitter is triggered, once the leftmost state, i.e., state $(i, 0, k)$, is reached.

If the (re)transmission is collision-free and π_k is matched at the receiver, the current cycle is successful, i.e., $\Omega_{\rm s}$, and the next state is state $(0, j, 1)$ in a new cycle. The probability of this transition is $\frac{1}{W_0}(1-p_c)(1-p_{\pi_k})$, where $p_{\pi_k} = \binom{K_{\rm P}-k}{K_{\rm pub}} / \binom{K_{\rm P}-k+1}{K_{\rm pub}}$ is the probability that π_k is not matched provided none of the preceding $(k-1)$ keys $\pi_{k'}$ $(k' = 1, \cdots, k-1)$ are matched, and p_c is the collision probability per slot, as given by

$$p_c = 1 - (1-\tau)^N. \tag{6.1}$$

Here, τ is the transmission probability of a node per slot.

If the (re)transmission is collision-free but unverified due to lack of matched keys, the next state is on the $(k+1)$-th plane. In other words, the transmitter sends the $(k+1)$-th new message together with the signature signed by π_{k+1} on the hash of the $(k+1)$ messages and other unexpired messages within the link duration. The probability of this transition is $\frac{1}{W_{i'}}(1-p_c)p_{\pi_k}$, where $W_{i'}$ is the backoff window of the next state on the $(k+1)$-th plane, and it depends on specific protocol designs, as will be discussed in Section 6.4.

If the (re)transmission is collided, the next state is state $(i+1, j, k)$, i.e., with the doubled backoff window W_{i+1} on the same plane. The probability of this transition is $\frac{1}{W_{i+1}}p_c$.

An exception is the case that a collision takes place at the leftmost state of the last chain on any plane, i.e., state $(M, 0, k)$. The cycle is a collided cycle, i.e., Ω_{c}, as the result of insufficiently large backoff window W_M. Another exception is state $(i, 0, K)$ on the last plane, where the (re)transmission is collision-free but π_K is unmatched. The cycle is unsuccessful, i.e., Ω_{k}, due to K unmatched keys. In both cases, the next state is state $(0, j, 1)$ starting a new cycle.

6.3.2 Unexpired Messages Between Cycles

The fourth dimension of the proposed model captures the transitions among the three different cycles with variable durations, namely, Ω_{s}, Ω_{k} and Ω_{c}, as illustrated on the right-hand side (RHS) of Fig. 6.2. Unexpired, collision-free but unauthenticated messages can be carried on along the transitions within a link duration τ_{link}. Let λ_{s}, λ_{k} and λ_{c} denote the transition possibilities from any cycle to the cycles Ω_{s}, Ω_{k}

and Ω_{c}, respectively. λ_{s}, λ_{k} and λ_{c} are also equal to the steady-state possibilities of Ω_{s}, Ω_{k} and Ω_{c}, respectively, since the cycles are independent provided the key selection is independent to preserve the Markov property; in other words, there is a possibility that the same, unmatched key is repeatedly selected in different cycles.

Here, λ_{s}, λ_{k} and λ_{c} can be given by

$$\lambda_{\mathrm{s}} = \sum_k \sum_i (1 - p_c)(1 - p_{\pi_k}) \Pr[i, k]; \tag{6.2a}$$

$$\lambda_{\mathrm{k}} = \sum_i (1 - p_c) p_{\pi_K} \Pr[i, K]; \tag{6.2b}$$

$$\lambda_{\mathrm{c}} = 1 - \Big(\lambda_{\mathrm{s}} + \lambda_{\mathrm{k}}\Big), \tag{6.2c}$$

where $(1-p_c)(1-p_{\pi_k})$ is the probability that the (re)transmission under W_i and π_k is successfully authenticated, and $(1-p_c)p_{\pi_K}$ is the probability that the (re)transmission is collision-free but unverified due to lack of matched key at the receiver. $\Pr[i,k]$ is the probability that there is a (re)transmission under W_i and π_k per cycle:

$$\Pr[i, k] = \Pr(i, 0, k)/\Pr(0, 0, 1), \tag{6.3}$$

where $\Pr(i,0,k)$ denotes the steady-state probability of the state $(i,0,k)$. Specifically, (6.2a) accounts for all possible cases in a cycle with successful authentication captured by the first three dimensions of the proposed Markov model; (6.2b) captures the cases using up to K private keys in the current cycle with K collision-free but unauthenticated messages; (6.2c) is self-explanatory. Capturing unexpired, collision-free but unverified messages from preceding cycles, we track back past unsuccessful and collided cycles within the current link duration once a private key is matched at the receiver.

6.4 Embodiment of the Proposed Model

In this section, three representative designs of on-the-fly authentication are discussed to illustrate the application of the proposed 4D Markov model, where the transmitter can take three different approaches to adjust the backoff window in response to a collision-free but unverified transmission stemming from an unmatched key. Closed-form analyses are inferred for authentication rate and delay. Particularly, we analyse the worst-case authentication delays of the designs, K and M of which need to be designed to be accommodated with a link duration, τ_{link}.

6.4.1 Design 1: IEEE 802.11 Compliant Retransmission and Rekeying

Design 1 complies with DCF (with little intrusion), where, every time a message is delivered collision-free, a confirmation is returned to the transmitter and consequently the backoff window of the transmitter is reset to the smallest W_0 for the next transmission (as done in DCF). In the case that a collision-free message is successfully authenticated, the receiver returns an ACK, as done under DCF; in the case that a collision-free message fails to be authenticated due to lack of the corresponding public key at the receiver, the receiver returns a NACK through the same slot reserved for the ACK, and accordingly the transmitter changes its key. In this sense, the backoff of the transmitter depend on collisions and are independent of rekeying, following a standard CSMA/CA and exponential backoff protocol.

For every cycle, the transition probabilities of Design 1 are given as follows:

$$\begin{cases} \Pr[(i,j-1,k)|(i,j,k)] = 1, & j \neq 0 \quad (6.4a) \\ \Pr[(0,j,1)|(i,0,k)] = \dfrac{1}{W_0}(1-p_c)(1-p_{\pi_k}), & i<M, k<K \quad (6.4b) \\ \Pr[(0,j,k+1)|(i,0,k)] = \dfrac{1}{W_0}(1-p_c)p_{\pi_k}, & k<K \quad (6.4c) \\ \Pr[(i+1,j,k)|(i,0,k)] = \dfrac{1}{W_{i+1}}p_c, & i<M \quad (6.4d) \\ \Pr[(0,j,1)|(M,0,k)] = \dfrac{1}{W_0}[1-(1-p_c)p_{\pi_k}], & k<K \quad (6.4e) \\ \Pr[(0,j,1)|(i,0,K)] = \dfrac{1}{W_0}(1-p_c), & i<M \quad (6.4f) \\ \Pr[(0,j,1)|(M,0,K)] = \dfrac{1}{W_0}, & (6.4g) \end{cases}$$

where (6.4a) captures the countdown of the backoff timer j; (6.4b) and (6.4c) account for collision-free (re)transmissions. Particularly, (6.4b) reflects the case that the key is matched at the receiver and a new cycle starts from $(0, j, 1)$. (6.4c) reflects the case that the key is not matched and a different key π_{k+1} is to be tried under a new transmission with W_0. (6.4d) accounts for failed authentications due to a (re)transmission collision. (6.4e), (6.4f) and (6.4g) describe the cases where the maximum window size or the largest number of keys to be tried in a cycle is reached, and the state transits to a new cycle starting with state $(0, j, 1)$.

Lemma 1: The steady-state probability of state triplet (i, j, k) of Design 1 is given by

$$\Pr(i, j, k) = \frac{W_i - j}{W_i} p_c^i \Pr(0, 0, k); \tag{6.5}$$

$$\Pr(0, 0, k) = (1 - p_c^{M+1})^{k-1} \prod_{s=1}^{k-1} p_{\pi_s} \Pr(0, 0, 1), \tag{6.6}$$

where both p_c and τ are independent of rekeying, i.e., key ring size, and can be evaluated by solving

$$\begin{cases} p_c = 1 - (1 - \tau)^N; \\ \tau = \dfrac{2(1 - p_c^{M+1})}{W_0(1 - p_c) \sum_{i=0}^{M} (2p_c)^i - p_c^{M+1} + 1}. \end{cases} \tag{6.7}$$

Proof 6.1: Note that (6.4a) and (6.4d) are exactly the same as (5.1a) and (5.1c) in Chapter 5. We can conclude (6.5), e.q., (5.3), withstands here with $i > 0$, since (5.3) was only dependent on (5.1a) and (5.1c).

Based on (6.4a) and (6.4c), the steady-state probabilities of triplet $(0, j, k), k > 1$ can be given by

$$\begin{aligned} \Pr(0, j, k) =& \frac{W_0 - j}{W_0} (1 - p_c) p_{\pi_{k-1}} \sum_{i=0}^{M} \Pr(i, 0, k - 1) && \text{(6.8a)} \\ =& \frac{W_0 - j}{W_0} (1 - p_c^{M+1}) p_{\pi_{k-1}} \Pr(0, 0, k - 1) && \text{(6.8b)} \\ =& \frac{W_0 - j}{W_0} (1 - p_c^{M+1})^{k-1} \Pr(0, 0, 1) \prod_{s=1}^{k-1} p_{\pi_s}, && \text{(6.8c)} \end{aligned}$$

where (6.8a) is obtained by recursively using (6.4a) and (6.4c). (6.8b) is based on $\Pr(i,0,k-1) = p_c^i \Pr(0,0,k-1)$, and $\sum_{i=0}^{M}(1-p_c)p_c^i = 1-p_c^{M+1}$. (6.8c) is obtained by letting $j=0$ in (6.8b), and recursively substituted in (6.8b). Substituting $j=0$ into (6.8c), (6.6) is proved.

From (6.4b), (6.4e), (6.4f) and (6.4g), the steady-state probabilities of the remaining states $(0,j,1)$ can be given by

$$\begin{aligned}\Pr(0,j,1) =& \frac{W_0-j}{W_0}\Bigg[\sum_{k=1}^{K-1}\sum_{i=0}^{M-1}(1-p_c)(1-p_{\pi_k})\Pr(i,0,k) + \sum_{i=0}^{M-1}(1-p_c)\Pr(i,0,K) \\ &+ \sum_{k=1}^{K-1}\left[1-(1-p_c)p_{\pi_k}\right]\Pr(M,0,k) + \Pr(M,0,K)\Bigg] \qquad (6.9a)\\ =& \frac{W_0-j}{W_0}\Pr(0,0,1), \qquad (6.9b)\end{aligned}$$

where the explanation of (6.9a) can be referred to the explanation of (6.8a) and (6.9b) is achieved by submitting $j=0$ into (6.9a). As a result, despite a number of different transition probabilities in (6.4) from those in (5.1), (5.12) still holds under this particular protocol for any i, i.e., (6.5) holds.

Since $\Pr(i,j,k) = \frac{W_i-j}{W_i}p_c^i\Pr(0,0,k)$ withstands in this model, as it did in (5.12), we can readily draw the same conclusion, i.e., (6.7), which is based on (5.12), as proved in Theorem 1 in Chapter 5. The details are therefore suppressed in this chapter.

Authentication Throughput

The protocol of Design 1 achieves the throughput, T_A, as given by

$$\begin{aligned}T_A = \frac{L_{\text{pkt}}}{\sigma}\sum_{k=1}^{K}\sum_{i=0}^{M}\Pr(i,0,k)(1-p_c)(1-p_{\pi_k})\times \qquad (6.10a)\\ \Big[k + \sum_{k'=1}^{K-k}k'P_{\text{pkt}}(k'|K-k)\Big], \qquad (6.10b)\end{aligned}$$

where L_{pkt} is the length of payload per packet; and σ is the average duration per slot given by

$$\sigma = (1-\tau)^{N+1}\sigma_0 + \big(1-(1-\tau)^{N+1}\big)\sigma_{\text{pkt}}, \tag{6.11}$$

since the possibility of a mini-slot with duration σ_0 is $(1-\tau)^{N+1}$, and the possibility of a transmission slot with duration σ_{pkt} is $\big[1-(1-\tau)^{N+1}\big]$; $\Pr(i,0,k)(1-p_c)(1-p_{\pi_k})$ gives the successful authentication possibility at state $(i,0,k)$; k is the number of authenticated messages contributed by the current successful cycle, and the other $k' \le (K-k)$ authenticated messages by $(K-k)$ successive messages in preceding cycles within the link duration τ_{link}. Following the fourth dimension of the proposed Markov model, $P_{\text{pkt}}(k'|K-k), 0 < k' \le (K-k)$ denotes the probability that the $(K-k)$ backtracked planes within τ_{link} contribute k' unexpired messages, and can be given by

$$P_{\text{pkt}}(K-k|K-k) = \lambda_{\text{k}}; \tag{6.12a}$$

$$P_{\text{pkt}}(K-k-1|K-k) = \sum_{x=K-k-1}^{K-1} P_c(x) + \sum_{x=0}^{K-k-2} P_{mc}\big(x|x+1\big)\lambda_{\text{k}}; \tag{6.12b}$$

$$P_{\text{pkt}}(k'|K-k) = \sum_{x=0}^{k'} P_{mc}\big(k'-x|K-k-x-1\big)\Big(\sum_{y=x}^{K-1} P_c(y)\Big) \tag{6.12c}$$

$$+\sum_{x=0}^{k'-1} P_{mc}\big(x|x+K-k-k'\big)\lambda_{\text{k}} \tag{6.12d}$$

$$+\sum_{x=k'+1}^{K-k-1} P_{mc}\big(k'|x\big)\lambda_{\text{s}}, 0 < k' \le K-k-2; \tag{6.12e}$$

where (6.12a) corresponds to the case that preceding the successful cycle is an unsuccessful cycle with all K unmatched keys. (6.12b) captures both the cases that a single preceding collided cycle alone contributes the $(K-k-1)$ backtracked collision-free messages, and that a collided cycle and a preceding unsuccessful cycle with K failed keys together contributes the $(K-k-1)$ collision-free messages. $P_c(x) = p_c \Pr[M, x+1], 0 \le x < K$, is the probability of a collided cycle with a total of $(x + 1)$ planes, in which case x messages are collision-free. $P_{mc}(k'|k), 0 \leq k' < k < K$, gives the conditional possibility of k' collision-free messages out of an integer number of complete collided cycles with a total of k planes. $P_{mc}(k'|k)$ can be obtained recursively given by

$$P_{mc}\big(k'|k\big) = \sum_{c_1+\cdots+c_{k-k'}=k'} \Big(P_c(c_1)\cdots P_c(c_{k-k'})\Big). \tag{6.13}$$

This is because any collided cycle only has the last message collided transmitted. The probability that $(k-k')$ collided cycles collectively contain a total of k' collision-free messages is used to calculate $P_{mc}\big(k'|k\big)$.

The rest of (6.12) collect all other possible cases for $k' = 1, \cdots, K - k - 2$. In particular, (6.12c) accounts for the case where the preceding $(K-k)$ planes all belong to collided cycles. (6.12d) accounts for the case where the $(K - k)$ planes consist of an integer number of successive collided cycles and a preceding unsuccessful cycle with all K unmatched keys. (6.12e) accounts for the case where the $(K - k)$ planes consist of an integer number of successive collided cycles and a preceding successful cycle. Based on (6.10) to (6.13), the authentication throughput of Design 1 can be calculated given p_c and τ.

It is clear that p_c and τ are the same as they are under the standard DCF which has no consideration on authentication. Therefore, they are independent of the key selection π_k in Design 1. The authenticated throughput of Design 1 can also be proved to asymptotically approach the unauthenticated throughput of DCF[*]. In other words, the cost of authentication on top of communication can diminish, as the link duration τ_{link} grows.

[*] The proof is not present in this chapter. For details, please refer to the Appendix E in our published paper J-1 listed in section *List of Publications*.

Authentication Delay and Link Duration

The average authentication delay of this protocol design, denoted by $\overline{D}$, is given by

$$\overline{D} = \frac{1}{\lambda_{\mathrm{s}}} \sum_{k=1}^{K} \sum_{i=0}^{M} \Pr[i,k](1-p_c)(1-p_{\pi(k)})\overline{D}(k);$$

where $\overline{D}(k)$ is the average authentication delay in the case that the last successful cycle consists of k planes, as given by

$$\overline{D}(K) = K\sigma_{\mathrm{nc}},$$

$$\begin{aligned}
\overline{D}(k) = {} & k\sigma_{\mathrm{nc}} + (K-k)\sigma_{\mathrm{nc}} P_{\mathrm{pkt}}(K-k|K-k) && \text{(6.14a)}\\
& + \Big((K-k-1)\sigma_{\mathrm{nc}} + \sigma_{\mathrm{c}}\Big) P_{\mathrm{pkt}}(K-k-1|K-k) && \text{(6.14b)}\\
& + \sum_{k'=1}^{K-k-2} \Big(k'\sigma_{\mathrm{nc}} + (K-k-k')\sigma_{\mathrm{c}}\Big)\Big(\sum_{x=0}^{k'} P_{mc}\big(k'-x|K-k-x-1\big) \sum_{y=x}^{K-1} P_c(y) \\
& + \sum_{x=0}^{k'-1} P_{mc}\big(x|x+K-k-k'\big)\lambda_{\mathrm{k}}\Big) && \text{(6.14c)}\\
& + \sum_{k'=1}^{K-k-2} \sum_{x=k'+1}^{K-k-1} \Big(k'\sigma_{\mathrm{nc}} + (x-k')\sigma_{\mathrm{c}}\Big) P_{mc}\big(k'|x\big)\lambda_{\mathrm{s}} && \text{(6.14d)}\\
& + \sum_{k'=0}^{K-k-1} \Big(k'\sigma_{\mathrm{c}}\Big)\big(P_c(0)\big)^{k'}\lambda_{\mathrm{s}} + \Big((K-k)\sigma_{\mathrm{c}}\Big)\big(P_c(0)\big)^{K-k}, k<K && \text{(6.14e)}
\end{aligned}$$

where $\sigma_{\mathrm{nc}} = \frac{1}{1-p_c^{M+1}} \sum_{i=0}^{M} \Big[p_c^i(1-p_c)\Big(\sum_{i'=0}^{i} d_{i'}\Big)\Big]$ and $\sigma_{\mathrm{c}} = \sum_{i=0}^{M} d_i$ are the average delays of an uncollided and a collided plane, respectively. $d_i = \frac{W_i-1}{2}\big((1-\tau)^N\sigma_0 + \big(1-(1-\tau)^N\big)\sigma_{\mathrm{pkt}}\big) + \sigma_{\mathrm{pkt}}$ is the average delay of the i-th backoff window. $(\sum_{i'=0}^{i} d_{i'})$ is the duration of a plane experiencing i (re)transmissions with possibility $p_c^i(1-p_c)$. $k\sigma_{\mathrm{nc}}$ is the average delay of the successful cycle with k planes, the other part on the RHS of (6.14a) corresponds to the case that the $(K-k)$ planes in preceding cycles all provide unexpired, collision-free but unauthenticated messages with the probability $P_{\mathrm{pkt}}(K-k|K-k)$. (6.14b) captures the case that $(K-k)$ planes provide $(K-k-1)$ unexpired messages with probability $P_{\mathrm{pkt}}(K-k-1|K-k)$, as in (6.12b).

(6.14c) accounts for the case that $(K-k)$ planes provide k' unexpired messages and $(K-k-k')$ collided messages; see (6.12c) and (6.12d). (6.14d) captures the case that $k' > 0$ unexpired messages are verified and the leading preceding cycle is a successful cycle; see (6.12e). (6.14e) captures the case that the $(K-k)$ planes do not contain unexpired messages.

It is important to evaluate the tolerance of the protocol design to τ_{link}. Particularly, the transmitter needs to identify K (which is also the maximum number of messages hashed in this design) in accordance with τ_{link}. For the sake of reliability, K can be identified as such that the worst-case authentication delay, D^{wst}, can be accommodated with a link duration τ_{link}. The worst-case delay here corresponds to the case that each (re)transmission is backed off for an entire backoff window (i.e., the delay is $W_i\sigma$ for the i-th retransmission) and each exponential backoff process reaches the maximum backoff window W_M, as given by

$$D^{\text{wst}} = K\sum_{i=0}^{M} W_i\sigma, \tag{6.15}$$

which should be shorter than τ_{link} by setting $K \leq \frac{\tau_{\text{link}}}{\sum_{i=0}^{M} W_i\sigma}$.

6.4.2 Design 2: IEEE 802.11 Compatible Joint Retransmission and Rekeying

Design 2 integrates opportunistic authentication in DCF. Specifically, the backoff window of the transmitter keeps doubling; until a message is delivered collision-free and authenticated (i.e., an ACK is returned), or W_M is reached. After that, the backoff window is reset. Different from Design 1, the transmitter of Design 2 doubles the backoff window on the receipt of NACKs. The exponential backoffs in response to not only transmission collisions, but also failed authentication attempts, can effectively enlarge the average backoff windows, alleviate collisions, and improve the throughput, but they can potentially increase the authentication delay.

The transition probabilities of triplet (i,j,k) of Design 2 are given by

$$\begin{cases} \Pr[(i,j-1,k)|(i,j,k)] = 1, & j \neq 0 \quad (6.16a) \\ \Pr[(0,j,1)|(i,0,k)] = \frac{1}{W_0}(1-p_c)(1-p_{\pi_k}), & i < M, k < K \, (6.16b) \\ \Pr[(i+1,j,k+1)|(i,0,k)] = \frac{1}{W_{i+1}}(1-p_c)p_{\pi_k}, & i < M, k < K (6.16c) \\ \Pr[(i+1,j,k)|(i,0,k)] = \frac{1}{W_{i+1}}p_c, & i < M \quad (6.16d) \\ \Pr[(0,j,1)|(i,0,K)] = \frac{1}{W_0}(1-p_c), & i < M \quad (6.16e) \\ \Pr[(0,j,1)|(M,0,k)] = \frac{1}{W_0}. & \quad (6.16f) \end{cases}$$

The differences between (6.16) and (6.4) are (6.16c) and (6.16f). This is due to the different designs taken in this approach when a message is collision-free but unverified due to the lack of the matched key at the receiver.

Lemma 2: The steady-state probability of state triplet (i,j,k) of Design 2 is given by $(i \geq k-1)$

$$\Pr(i,j,k) = \frac{W_i - j}{W_i}\binom{i}{k-1}\frac{(1-p_c)^{k-1}}{p_c^{k-1-i}}\prod_{x=1}^{k-1} p_{\pi_x} \Pr(0,0,1), \tag{6.17}$$

where both p_c and τ are dependent on rekeying, and can be evaluated by solving

$$\begin{cases} p_c = 1-(1-\tau)^N; \\ \tau = \dfrac{\sum_{k=1}^{K}\sum_{i=k-1}^{M}\binom{i}{k-1}\frac{(1-p_c)^{k-1}}{p_c^{k-1-i}}\prod_{x=1}^{k-1}p_{\pi_x}}{\sum_{k=1}^{K}\sum_{i=k-1}^{M}\frac{W_i+1}{2}\binom{i}{k-1}\frac{(1-p_c)^{k-1}}{p_c^{k-1-i}}\prod_{x=1}^{k-1}p_{\pi_x}}. \end{cases} \tag{6.18}$$

Proof 6.2: Note that state $(i,j,1)$ satisfies (6.17), as can be derived based on (6.16) in the same way as (6.9) is based on (6.4). To prove (6.17), we first assume that state $(i,j,k-1)$ satisfies (6.17), given that state $(i,j,1)$ satisfies (6.17). The steady-state probability of state $(k-1, j, k), k > 1$ can be given by

$$\Pr(k-1,j,k) = \frac{(1-p_c)p_{\pi_{k-1}}}{W_{k-1}} \Pr(k-2,0,k-1) + \Pr(k-1,j+1,k) \tag{6.19a}$$

$$= \frac{W_{k-1}-j}{W_{k-1}}(1-p_c)p_{\pi_{k-1}} \Pr(k-2,0,k-1) \tag{6.19b}$$

$$= \frac{W_{k-1}-j}{W_{k-1}}(1-p_c)^{k-1} \prod_{x=1}^{k-1} p_{\pi_x} \Pr(0,0,1), \tag{6.19c}$$

where (6.19a) is due to the fact that state $(k-1, j, k)$ can only transit from states $(k-2, 0, k-1)$ and $(k-1, j+1, k)$, i.e., (6.16c) and (6.16a); (6.19b) is obtained by recursively incrementing j in (6.19a) and substituting that into the RHS of (6.19a); (6.19c) is based on the assumption that (6.17) holds for state $(k-2, 0, k-1)$, and substituted into (6.19b).

To further prove the steady-state probability of state $(i, j, k), i > k-1$ satisfying (6.17), we further assume that state $(i', j, k), i' < i$ satisfies (6.17), which is reasonable based on (6.19). Therefore $\Pr(i, j, k)$ for $i > k-1$ can be likewise evaluated as follows,

$$\Pr(i,j,k) = \frac{W_i - j}{W_i}\Big[p_c \Pr(i-1,0,k) + (1-p_c)p_{\pi_{k-1}} \Pr(i-1,0,k-1)\Big] \tag{6.20a}$$

$$= \frac{W_i - j}{W_i}\binom{i}{k-1}\frac{(1-p_c)^{k-1}}{p_c^{k-1-i}} \prod_{x=1}^{k-1} p_{\pi_x} \Pr(0,0,1). \tag{6.20b}$$

From (6.19) and (6.20), the validity of (6.17) is confirmed. Using (6.17), we have

$$\sum_k \sum_i \sum_j \Pr(i,j,k) = \sum_{k=1}^{K} \sum_{i=k-1}^{M} \frac{W_i+1}{2}\binom{i}{k-1}\frac{(1-p_c)^{k-1}}{p_c^{k-1-i}} \prod_{x=1}^{k-1} p_{\pi_x} \Pr(0,0,1) = 1,$$

which leads to

$$\Pr(0,0,1) = \frac{1}{\sum_{k=1}^{K} \sum_{i=k-1}^{M} \frac{W_i+1}{2}\binom{i}{k-1}\frac{(1-p_c)^{k-1}}{p_c^{k-1-i}} \prod_{x=1}^{k-1} p_{\pi_x}}. \tag{6.21}$$

The transmission probability, τ, can be given by

$$\begin{aligned}\tau =&\sum_{k=1}^{K}\sum_{i=0}^{M}\Pr(i,0,k) & (6.22a)\\ =&\sum_{k=1}^{K}\sum_{i=k-1}^{M}\binom{i}{k-1}\frac{(1-p_c)^{k-1}}{p_c^{k-1-i}}\prod_{x=1}^{k-1}p_{\pi_x}\Pr(0,0,1) & (6.22b)\\ =&\frac{\sum_{k=1}^{K}\sum_{i=k-1}^{M}\binom{i}{k-1}\frac{(1-p_c)^{k-1}}{p_c^{k-1-i}}\prod_{x=1}^{k-1}p_{\pi_x}}{\sum_{k=1}^{K}\sum_{i=k-1}^{M}\frac{W_i+1}{2}\binom{i}{k-1}\frac{(1-p_c)^{k-1}}{p_c^{k-1-i}}\prod_{x=1}^{k-1}p_{\pi_x}}, & (6.22c)\end{aligned}$$

since a (re)transmission takes place if and only if $j = 0$. (6.22b) is obtained by substituting (6.17) into the RHS of (6.22a), and (6.22c) is by substituting (6.21) into (6.22b). p_c and τ can be obtained by jointly solving (6.1) and (6.22).

Authentication Throughput

The protocol of Design 2 achieves the throughput, T_A, as given by

$$T_A = \frac{L_{\text{pkt}}}{\sigma}\sum_{k=1}^{K}\sum_{i=0}^{M}\Pr(i,0,k)(1-p_c)(1-p_{\pi_k})(k+E_{\text{cyc}}), \qquad (6.23)$$

where E_{cyc} is the average number of unexpired, collision-free but unverified messages from the preceding $(K-1)$ cycles, and can be given by

$$\begin{aligned}E_{\text{cyc}} =&\sum_{x=0}^{K-2}\sum_{y=0}^{K-2-x}\binom{x+y}{x}(\lambda_{\text{k}})^{x}(\lambda_{\text{c}})^{y}\lambda_{\text{s}}\Big(x\overline{E_k}+y\overline{E_c}\Big)\\ &+\sum_{x=0}^{K-1}\binom{K-1}{x}(\lambda_{\text{k}})^{x}(\lambda_{\text{c}})^{K-1-x}\Big(x\overline{E_k}+(K-1-x)\overline{E_c}\Big),\end{aligned}$$

where $\overline{E_c}$ and $\overline{E_k}$ are the average numbers of collision-free messages in Ω_{c} and Ω_{k}, respectively; and are given by

$$\begin{aligned}\overline{E_c} =&\sum_{k=0}^{K-1}k\frac{\Big(P_c(k)+P_c^{*}(k)\Big)}{\lambda_{\text{c}}};\\ \overline{E_k} =&\min\{K, M+1\},\end{aligned}$$

where $P_c(x) = p_c\Pr[M, x+1], 0 \le x < K$ and $P_c^{*}(x) = (1-p_c)p_{\pi_x}\Pr[M, x], 1 \le x < K$, are the probabilities of a collided cycle Ω_{c} with a total of $(x+1)$ and x planes, respectively, in both of which x messages are delivered collision-free to the receiver.

Authentication Delay and Link Duration

The average steady-state authentication delay is given by

$$\begin{aligned}\overline{D} =&\sigma_{\mathrm{s}} + \sum_{x=0}^{K-2}\sum_{y=0}^{K-2-x}\binom{x+y}{x}(\lambda_{\mathrm{k}})^{x}(\lambda_{\mathrm{c}})^{y}\lambda_{\mathrm{s}}(x\sigma_{\mathrm{k}}+y\sigma_{\mathrm{c}})\\ &+\sum_{x=0}^{K-1}\binom{K-1}{x}(\lambda_{\mathrm{k}})^{x}(\lambda_{\mathrm{c}})^{K-1-x}(x\sigma_{\mathrm{k}}+(K-1-x)\sigma_{\mathrm{c}}),\end{aligned}\tag{6.24}$$

where σ_{s}, σ_{c} and σ_{k} are the average delays of a successful and collided cycle, as well as an unsuccessful cycle with all K keys failed, respectively:

$$\begin{aligned}\sigma_{\mathrm{s}} =&\sum_{k=1}^{K}\sum_{i=0}^{M}\big(\sum_{i'=0}^{i}d_{i'}\big)\frac{(1-p_c)(1-p_{\pi_k})\Pr[i,k]}{\lambda_{\mathrm{s}}};\\ \sigma_{\mathrm{k}} =&\sum_{i=0}^{M}\big(\sum_{i'=0}^{i}d_{i'}\big)\frac{(1-p_c)p_{\pi_K}\Pr[i,K]}{\lambda_{\mathrm{k}}};\\ \sigma_{\mathrm{c}} =&\sum_{i'=0}^{M}d_{i'},\end{aligned}$$

where $(1-p_c)(1-p_{\pi_k})\Pr[i,k]/\lambda_{\mathrm{s}}$ is the possibility that a successful cycle ends at state $(i,0,k)$, and transits to state $(0,j,1)$ with possibility $(1-p_c)(1-p_{\pi_k})$; see (6.16). $(1-p_c)p_{\pi_K}\Pr[i,K]/\lambda_{\mathrm{k}}$ is the possibility that an unsuccessful cycle ends at state $(i,0,K)$, and transits to state $(0,j,1)$ with possibility $(1-p_c)p_{\pi_K}$; see (6.16). The duration of the cycle in both cases are $\sum_{i'=0}^{i}d_{i'}$. The remainder is the case that a collided cycle ends at state $(M,0,k)$.

As discussed in Section 6.4.1, we evaluate the tolerance of this protocol design to the link duration τ_{link} by analyzing the worst-case authentication delay. The worst-case authentication delay of this design can be given in the same way as (6.15), which corresponds to the scenario that each cycle ends at the maximum backoff window, W_M.

6.4.3 Design 3: Collision-aware Retransmission and Rekeying

Design 3 arises in attempt to reduce the authentication delay of Design 2, by exploiting the capability of the transmitter, enabled by NACKs, to differentiate transmission collisions and failed authentications. Specifically, the transmitter does not change its current backoff window but changes its key if a failed authentication is due to a lack of matched key (i.e., a NACK is returned), as opposed to resetting or doubling the window (as done in Designs 1 and 2, respectively). The transmitter doubles its backoff window and does not change its key, if a failed authentication is due to a transmission collision (i.e., neither ACK nor NACK is returned). The transmitter resets its backoff window to W_0 if the authentication succeeds (i.e., an ACK is returned).

The transition probabilities of triplet (i, j, k) of Design 3 are given by

$$\begin{cases} \Pr[(i, j-1, k)|(i, j, k)] = 1, & j \neq 0 \quad (6.25a) \\ \Pr[(0, j, 1)|(i, 0, k)] = \dfrac{1}{W_0}(1-p_c)(1-p_{\pi_k}), & i < M, k < K (6.25b) \\ \Pr[(i, j, k+1)|(i, 0, k)] = \dfrac{1}{W_i}(1-p_c)p_{\pi_k}, & k < K \quad (6.25c) \\ \Pr[(i+1, j, k)|(i, 0, k)] = \dfrac{1}{W_{i+1}}p_c, & i < M \quad (6.25d) \\ \Pr[(0, j, 1)|(M, 0, k)] = \dfrac{1}{W_0}[1-(1-p_c)p_{\pi_k}], & k < K \quad (6.25e) \\ \Pr[(0, j, 1)|(i, 0, K)] = \dfrac{1}{W_0}(1-p_c), & i < M \quad (6.25f) \\ \Pr[(0, j, 1)|(M, 0, K)] = \dfrac{1}{W_0}. & \quad (6.25g) \end{cases}$$

The main difference of (6.25) to (6.16) and (6.4) lies in (6.25c), where the key is unmatched for a collision-free transmission at state $(i, 0, k), k < K$ and transits to state $(i, 0, k+1)$ with a new key selected under the same backoff window.

Lemma 3: The steady-state possibility of state triplet (i, j, k) in this protocol is given by

$$\Pr(i, j, k) = \frac{W_i - j}{W_i}\binom{i+k-1}{i}(1-p_c)^{k-1}p_c^i \prod_{x=1}^{k-1} p_{\pi_x} \Pr(0, 0, 1), \qquad (6.26)$$

where both p_c and τ are dependent on rekeying, and can be evaluated by solving

$$\begin{cases} p_c = 1-(1-\tau)^N, \\ \tau = \dfrac{\sum_{k=1}^{K}\sum_{i=0}^{M}\binom{i+k-1}{i}(1-p_c)^{k-1}p_c^i\prod_{x=1}^{k-1}p_{\pi_x}}{\sum_k\sum_i\sum_j\frac{W_i-j}{W_i}\binom{i+k-1}{i}(1-p_c)^{k-1}p_c^i\prod_{x=1}^{k-1}p_{\pi_x}}. \end{cases} \tag{6.27}$$

Proof 6.3: Note that state $(i,j,1)$ satisfies (6.26), as can be derived based on (6.25) in the same way as (6.9) is based on (6.4). To prove (6.26), we first assume that state $(i,j,k-1)$ satisfies (6.26), given that state $(i,j,1),\forall i,j$ satisfies (6.26). Then, the steady-state probability of state $(0,j,k), k>1$ can be given by

$$\Pr(0,j,k) = \Pr(0,j+1,k) + \frac{1}{W_0}(1-p_c)p_{\pi_{k-1}}\Pr(0,0,k-1) \tag{6.28a}$$

$$= \frac{W_0-j}{W_0}(1-p_c)p_{\pi_{k-1}}(1-p_c)^{k-2}\prod_{x=1}^{k-2}p_{\pi_x}\Pr(0,0,1) \tag{6.28b}$$

$$= \frac{W_0-j}{W_0}(1-p_c)^{k-1}\prod_{x=1}^{k-1}p_{\pi_x}\Pr(0,0,1), \tag{6.28c}$$

where (6.28a) is due to the fact that state $(0,j,k)$ can only transit from state $(0,j+1,k)$; see (6.25a), and from state $(0,0,k-1)$ provided key π_{k-1} is unavailable at the receiver; see (6.25c). (6.28b) is obtained by substituting $\Pr(0,0,k-1)$ with (6.26) based on the assumption that state $(i,j,k-1)$ satisfies (6.26). Obviously, (6.28c) confirms that state $(0,j,k)$ satisfies (6.26).

The steady-state possibility of state $(i,j,k), i>0$ is provided in (6.29), since state $(i,j,k'), k'<k, \forall i,j$ and $(i-1,j,k)$ all satisfies (6.26), based on (6.28), respectively.

$$\begin{aligned}
\Pr(i,j,k) =&\Pr(i,j+1,k) + \frac{1}{W_i}\Big[p_c \Pr(i-1,0,k) + \\
&(1-p_c)p_{\pi_{k-1}} \Pr(i,0,k-1)\Big] && (6.29a)\\
=&\frac{W_i-j}{W_i}\Big[p_c \Pr(i-1,0,k) + (1-p_c)p_{\pi_{k-1}} \Pr(i,0,k-1)\Big] && (6.29b)\\
=&\frac{W_i-j}{W_i}\Pr(0,0,1)\Big[p_c\binom{i+k-2}{i-1}(1-p_c)^{k-1}p_c^{i-1}\prod_{x=1}^{k-1}p_{\pi_x}\\
&+(1-p_c)p_{\pi_{k-1}}\binom{i+k-2}{i}(1-p_c)^{k-2}p_c^i\prod_{x=1}^{k-2}p_{\pi_x}\Big] && (6.29c)\\
=&\frac{W_i-j}{W_i}\Big[\binom{i+k-1}{i}(1-p_c)^{k-1}p_c^i\prod_{x=1}^{k-1}p_{\pi_x}\Big]\Pr(0,0,1) && (6.29d)
\end{aligned}$$

where (6.29a) is because state (i,j,k) transits from states $(i,j+1,k)$, $(i-1,0,k)$ and $(i,0,k-1)$ with possibilities given in (6.25a), (6.25d) and (6.25c), respectively. (6.29b) is obtained by recursively substituting $\Pr(i,j+1,k)$ with (6.29a). (6.29c) is given by substituting $\Pr(i-1,0,k)$ and $\Pr(i,0,k-1)$ with (6.26). (6.29d) proves that $\Pr(i,j,k)$ satisfies (6.26).

Note that $\sum_{k=1}^{K}\sum_{i=0}^{M}\sum_{j=0}^{W_i-1}\Pr(i,j,k)=1$. Therefore,

$$\sum_{k=1}^{K}\sum_{i=0}^{M}\sum_{j=0}^{W_i-1}\frac{W_i-j}{W_i}\binom{i+k-1}{i}(1-p_c)^{k-1}p_c^i\prod_{x=1}^{k-1}p_{\pi_x}\Pr(0,0,1)=1.$$

As a result,

$$\Pr(0,0,1)=\frac{1}{\sum_k\sum_i\sum_j\frac{W_i-j}{W_i}\binom{i+k-1}{i}(1-p_c)^{k-1}p_c^i\prod_{x=1}^{k-1}p_{\pi_x}}. \qquad (6.30)$$

The transmission probability τ is given by

$$\begin{aligned}
\tau =& \sum_{k=1}^{K}\sum_{i=0}^{M}\Pr(i,0,k) && (6.31a)\\
=& \frac{\sum_{k=1}^{K}\sum_{i=0}^{M}\binom{i+k-1}{i}(1-p_c)^{k-1}p_c^i\prod_{x=1}^{k-1}p_{\pi_x}}{\sum_k\sum_i\sum_j\frac{W_i-j}{W_i}\binom{i+k-1}{i}(1-p_c)^{k-1}p_c^i\prod_{x=1}^{k-1}p_{\pi_x}}, && (6.31b)
\end{aligned}$$

since a (re)transmission takes place if and only if $j=0$. (6.31b) is achieved by substituting (6.26) and (6.30) into (6.31a). p_c and τ can be obtained by solving (6.1) and (6.31).

Authentication Throughput

The protocol of Design 3 achieves the throughput, T_A, as given by (6.23), with $\overline{E_c}$ and $\overline{E_k}$ updated by $\overline{E_c} = \sum_{k=0}^{K-1} kP_c(k)/\lambda_\text{c}$ and $\overline{E_k} = K$, and p_c and τ also updated accordingly based on Lemma 3. This is the case where the last successful cycle backtracks $(K-1)$ cycles like Design 2. $\overline{E_c}$ and $\overline{E_k}$ are updated because an unsuccessful cycle ends with K messages in the cycle, while the collided cycle ends at $(M, 0, k+1)$ with possibility $P_c(k)/\lambda_\text{c}$ and k collision-free yet unverified messages.

Authentication Delay and Link Duration

The average steady-state authentication delay can be also given by (6.24) with σ_s, σ_k and σ_c updated by

$$\sigma_\text{s} = \sum_{k=1}^{K}\sum_{i=0}^{M}\Big(\sum_{i'=0}^{i} d_{i'} + \Delta[i,k]\Big)\Big)\frac{(1-p_c)(1-p_{\pi_k})\Pr[i,k]}{\lambda_\text{s}};$$
$$\sigma_\text{k} = \sum_{i=0}^{M}\Big(\sum_{i'=0}^{i} d_{i'} + \Delta[i,K]\Big)\Big)\frac{(1-p_c)p_{\pi_K}\Pr[i,K]}{\lambda_\text{k}};$$
$$\sigma_\text{c} = \sum_{k=1}^{K}\Big(\sum_{i'=0}^{M} d_{i'} + \Delta[M,k]\Big)\Big)\frac{p_c\Pr[M,k]}{\lambda_\text{c}},$$

which can be referred to (6.25). $\Delta[i,k]$ denotes the extra delay of a cycle, compared with IEEE 802.11p, ending with state $(i, 0, k)$. This is because, at each time of rekeying, the transmission under the new key uses the same backoff window but on the next plane:

$$\begin{aligned}\Delta[i,1] &= 0;\\ \Delta[i,k] &= \frac{1}{\binom{i+k-1}{i}}\sum_{0\le i_2\le\cdots\le i_k\le i}\big(d_{i_2}+\cdots+d_{i_k}\big)\\ &= \frac{1}{\binom{i+k-1}{i}}\frac{(k-1)\binom{i+k-1}{i}}{i+1}\sum_{i'=0}^{i} d_{i'} = \frac{k-1}{i+1}\sum_{i'=0}^{i} d_{i'};\end{aligned} \tag{6.32}$$

where $\binom{i+k-1}{i}$ is the number of possible combinations that the extra transmissions take place in a cycle ending with state $(i, 0, k)$. $\big(d_{i_2}+\cdots+d_{i_k}\big)$ is the total extra delay in a cycle, provided the extra transmissions on the k'-th plane take place within the backoff window $W_{i_{k'}}, 0 \leq i_2 \leq \cdots \leq i_k \leq i$.

As done in Sections 6.4.1 and 6.4.2, the tolerance of this design to τ_{link} is evaluated by analyzing the worst-case authentication delay and identifying K. The worst-case authentication delay of the design is given by

$$D^{\text{wst}} = K\Big(\sum_{i=0}^{M} W_i\sigma + \frac{K-1}{M+1}\sum_{i=0}^{M} W_i\sigma\Big),$$

where $\frac{K-1}{M+1}\sum_{i=0}^{M} W_i\sigma$ captures the extra delay of a cycle ending with state $(M, 0, K)$, from repeated (re)transmissions without increasing the backoff windows; see (6.32).

6.5 Numerical Result

In this section, Monte-Carlo simulations are carried out in the C++ environment to validate the proposed Markov model. Our analytical results are produced by using MATLAB. We set the initial backoff window size W_0 to 16, and the maximum backoff window W_M to $2^6 W_0$. The minimum slot length is 16μs. The durations of the PLCP preamble, PLCP header, DIFS, SIFS and ACK are 20μs, 4μs, 32μs, 16μs and 40μs, respectively. The size of a signature is 56 bytes. These parameters are based on the IEEE 802.11p standard [3]. For comparison purposes, the EG scheme [38] and the piggyback scheme [100] are also simulated. For fair comparison, two neighbours shake hands to decide on common keys before message transmissions in the EG scheme, while the piggyback scheme transmits the public key and certificate, once the topology changes. Only signatures are transmitted with messages in both the EG and the piggyback schemes during a link duration.

Fig. 6.3 plots the transmission collision probabilities, p_c, of the three representative protocol designs of on-the-fly authentication, in comparison with the EG and piggyback schemes. It is seen that the simulation results coincide with the analysis, validating the proposed three Lemmas and the accuracy of the proposed model. We

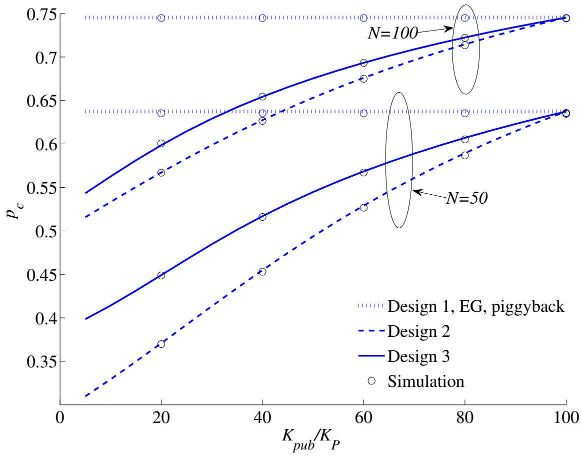
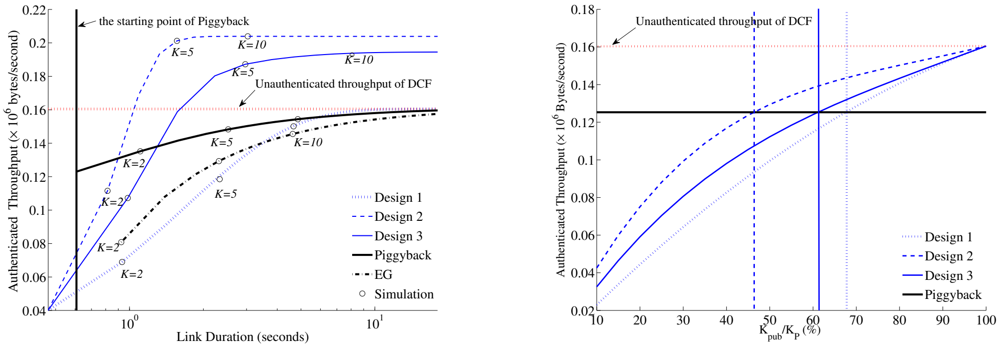


Figure 6.3 : Transmission collision probabilities.

also see that the network density, N, has strong impact on p_c, since transmission collisions between the nodes become increasingly intensive given limited wireless resource. Another finding is that p_c remains the same across Design 1, and the EG and piggyback schemes. This is due to the fact that the retransmissions and rekeying are independently carried out in these schemes, as done in IEEE 802.11p. Furthermore, we see that Design 2 incurs less severe collisions than Design 3, since Design 2 doubles the backoff windows in both cases of collisions and key failures, and alleviates collisions. In contrast, Design 3 doubles the backoff window only in response to collisions.

Fig. 6.4(a) evaluates the impact of link duration on the network throughput, where, for every link duration τ_{link}, we determine K for different protocols by setting $D^{\text{wst}} \leq \tau_{\text{link}}$, and analysis and simulations are conducted, given K. The authenticated throughput of the opportunistic authentication protocols, i.e., Designs 2 and 3, can be significantly higher than the unauthenticated throughput of DCF. Even the authenticated throughput of Design 1 can approach the unauthenticated throughput of DCF. In this sense, the cost of authentication resulting from failed opportunistic authentication attempts can asymptotically diminish, as the link du-



(a) Authenticated throughput versus link duration, where $K_{\text{pub}} = 25\% K_{\text{P}}$.

(b) Authenticated throughput versus the number of public keys per node, where $\tau_{\text{link}} = 0.7$s

Figure 6.4 : Authenticated throughput in an area, where $N = 50$, $L_{\text{pkt}} = 100$ bytes, and $K_{\text{pri}} = 50$.

rations grow. This is due to the fact that, with the growth of link duration, the number of keys which can be tested for a packet increases, and so does the likelihood of successful authentication attempts.

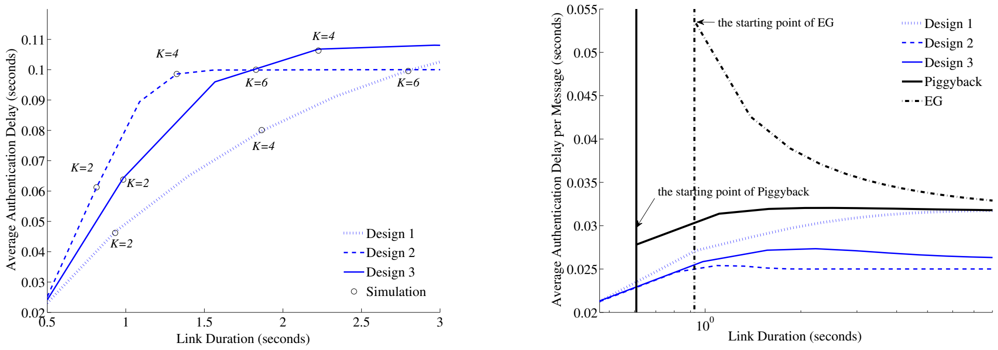
Designs 2 and 3 are also superior in terms of throughput to the piggyback scheme over a wide range of link duration, e.g., $\tau_{\text{link}} \geq 0.8$s (see Design 2) and 0.4s$\leq \tau_{\text{link}} \leq$ 0.6s, as a result of reduced authentication overhead and relieved collisions. The gains of Designs 2 and 3 over the piggyback scheme are around 30% and 20% at $\tau_{\text{link}} = 10$s, respectively. In the case of very short link durations, i.e., $\tau_{\text{link}} \leq 0.6$s, the overhead and the consequently prolonged transmission durations of messages in the piggyback scheme can lead to incomplete, interrupted transmissions, resulting in the loss of throughput. In contrast, the proposed opportunistic authentication protocols neither incur this extra overhead, nor require handshaking (as required in the EG scheme). Therefore, they can operate under significantly short link durations, i.e., $\tau_{\text{link}} \geq 0.4$s; see Fig. 6.4(a).

When τ_{link} is small (around 0.6s to 0.8s), the piggyback scheme can outperform the rest, as shown in Fig. 6.4(a). Such link durations are long enough for prolonged transmissions of the piggyback scheme to get throughput, but insufficient for the opportunistic authentication protocols or the EG scheme. Because the EG scheme incurs delays to decide on common keys, while the opportunistic protocols undergo a delay to match the keys at the receiver. However, the opportunistic protocols can outperform the piggyback scheme with the growing number of preloaded public keys per node, even under a short link duration of $\tau_{\text{link}} = 0.7$s, as shown in Fig. 6.4(b). This is due to the increasing probability that the receivers are preloaded the public keys matching the private keys the transmitter uses.

It is interesting to see that Design 2 can substantially outperform Design 3 in terms of throughput in Fig. 6.4, although these two designs bear a strong resemblance and the latter can also exploit the assistance of NACK to a greater extent. However, Design 2 doubles the backoff window in response to both key failures and collisions, while Design 3 only does so in response to collisions. The conclusion drawn is that on-the-fly authentication protocols are sensitive to collisions. Differentiating key failures from collisions, with the assistance of NACK, may not help in terms of throughput.

As expected in Section 6.4.3, Fig. 6.5(a) indicates that the assistance of NACK does help reduce the authentication delay of Design 3, compared to Design 2, in the case of short link durations. The capability of identifying failed authentications caused by collisions from those caused by key failures can help Design 3 avoid unnecessary expansion of the backoff window, and so cut the delay short.

Fig. 6.5(b) compares the average authentication delay of a message between the different authentication protocols. We see that the proposed opportunistic authentication protocols are superior in terms of the average authentication delay, across the entire spectrum of link duration. In particular, Design 2 of opportunistic au-



(a) The average delay versus the link duration (b) The average delay per message versus the link duration

Figure 6.5 : Average delay versus the link duration, where K varies from 1 to 20, $N = 50$ and $K_{\text{pub}} = 25\% K_{\text{P}}$.

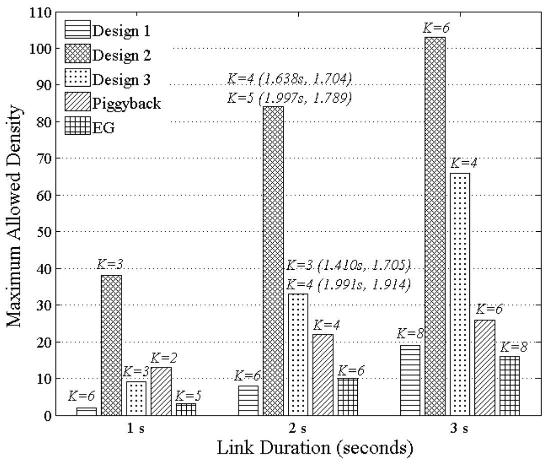


Figure 6.6 : Maximum number of neighbours where T_A is no less than 1.7×10^5 bytes/second and $K_{\text{pub}} = 25\% K_{\text{P}}$.

thentication protocol, which in general provides the highest throughput, is shown to achieve the shortest average authentication delay, even under link durations between $\tau_{\text{link}} = 0.4$s and 0.8s. This is due to the fact that, without extra overhead, Design 2 incurs short delays for successful authentications, compared to the piggyback and EG schemes. Design 2 takes a longer time to identify the right key than Design 3,

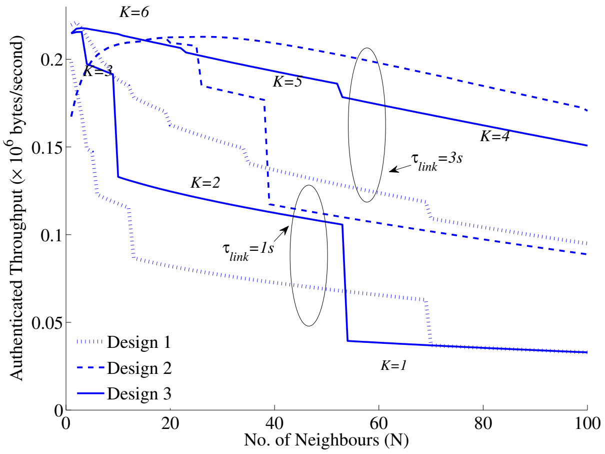


Figure 6.7 : Authenticated throughput, where N varies from 1 to 100, $K_{\text{pub}} = 25\% K_{\text{P}}$ and $K_{\text{pri}} = 50$.

but can verify disproportionally more messages after identifying the key (as a result of alleviated collisions), as respectively shown in Figs. 6.5(a) and 6.5(b).

Fig. 6.6 demonstrates the scalability of the on-the-fly authentication protocols in dense and large distributed mobile IoT network environments with changing topologies, where τ_{link} = 1s, 2s and 3s, and the throughput is set to be no less than 1.7×10^5 bytes/s. Specifically, given τ_{link}, K and N are jointly evaluated for each of the protocols, and the maximum N achieving more than 1.7×10^5 bytes/s are recorded. We can see that Design 2 can support the densest networks, followed by Design 3, due to the relieved collisions, as shown in Fig. 6.3. Design 1 can have limited scalability, close to the EG and piggyback schemes, as shown earlier in Fig. 6.3. In this sense, direct use of CSMA/CA without cross-layer consideration of rekeying would not provide much gain to on-the-fly authentication.

Fig. 6.7 plots the throughput under different link durations, and different network densities, ranging from 1 to 100. As done in Fig. 6.4, K is calibrated to meet the link duration, i.e., $D^{\text{wst}} \leq \tau_{\text{link}}$, given N and τ_{link}. In the case of low network densities, Design 3 typically outperforms other on-the-fly protocols when the network density is low, e.g., less than 5 and 15 at $\tau_{\text{link}} = 1\text{s}$ and 3s, respectively. This is due to the fact that Design 3 can alleviate collisions, as compared to Design 1, while reducing authentication delays (i.e., adversely contributing to the throughput), as compared to Design 2. In the case that the network is dense, i.e., $N \geq 20$, Design 2 has the highest throughput, followed by Designs 3 and 1. This is because Design 2 reduces the probability of packet collisions p_c. As a matter of fact, Design 2 has the highest probability of collision-free transmissions, i.e., $\tau(1-p_c) = [1-(1-p_c)^{1/N}](1-p_c)$ among the three designs. This can be evaluated by substituting the different values of p_c in (6.7), (6.18) and (6.27) for Designs 1, 2 and 3, respectively. As a result of the highest probability of collision-free transmissions, Design 2 has the largest number of keys tested within a link duration, leading to the highest authentication success rates.

It is noteworthy that multiple values of K need to be taken to plot every curve, and the curves become inconsistent, as the network density grows and in turn K decreases. However, in the case of $\tau_{\text{link}} = 3\text{s}$, Design 2 displays a consistent change of throughput with growing density, even though different values of K also need to be taken. This is because K typically takes a large value of greater than 6. As shown in Fig. 6.4(a), the throughput converges under a large K value.

6.6 Summary

This chapter analyses the impact of short link duration on on-the-fly authentication protocols which can instantly verify and forward messages. A general 4D Markov model is proposed to capture unexpired messages between cycles of the protocols in the fourth dimension, in addition to the first three dimensions to model a cycle of the protocols. Validated by simulations, the proposed model is general and is able to characterize a range of protocols. The analysis reveals that the on-the-fly authentication is sensitive to transmission collisions. The proposed model is able to facilitate holistic cross-layer designs over retransmission and rekeying, allowing the protocols to significantly outperform the state of the art.

Chapter 7

Conclusion

The thesis studied vulnerabilities and security challenges to wireless data transmission in decentralized IoT networks. Security solutions in terms of location privacy, data confidentiality and data integrity were discussed, proposed and analysed in the last four chapters (Chapter 3 to Chapter 6). This chapter concludes this thesis by recalling the contributions of each chapter, with the future work presented based on our current research.

7.1 Contribution

IoT network prevails with its ability to interconnect numerous devices possessing various sensing and computing abilities with little human interventions. With the rise in the number of connected IoT devices, the potential vulnerabilities in IoT increase as well. Lack of IoT data security will affect the large-scale deployment of IoT technology. However, simply extending computationally demanding and costly Internet security solutions to IoT is neither scalable nor practical, due to the specific characteristics of IoT networks, i.e., an enormous number of nodes and data, decentralization, heterogeneity, and unpredictable connections.

In this thesis, we studied the location privacy protection and secure data transmission issues to guarantee the data confidentiality, integrity, non-repudiation and availability in IoT networks. The attack model considered in this thesis comprises of passive, active, internal and external adversaries, which collaboratively launch eavesdropping, traffic analysis, pollution attack, and data injection in IoT networks.

We proposed an authenticated source-location privacy solution, which integrates a homomorphic signature algorithm with key predistribution schemes, to provide data integrity, as well as source location privacy, in a wireless decentralized IoT network. Furthermore, the complex impacts of uncoordinated transmissions and key distributions on the secure data transmission performance were taken into consideration. We proposed a series of opportunistic encrypted and authenticated data transmission protocols, where a new acknowledgment was designed to distinguish the cause of a failed (re)transmission between a packet collision and a mismatched key. To tackle the dynamic topology, these proposed protocols complete secure data transmission in an on-the-fly manner. Moreover, a set of Markov models were also designed to analyze the network performance of secure data transmission mechanisms. Validated by simulation results, these analysis models are accurate in capturing the impact of key selections, channel collisions and link durations on secure data transmission mechanisms. Insight guidance for selecting appropriate secure transmission protocols in various network environments was also provided based on the analysis results. The contributions of individual chapters are summarized as follows.

Chapter 1 covered the introduction to the Internet of Things and an overview of the thesis. The development, architecture and characteristics of IoT networks were presented, followed by two typical types of IoT networks, i.e., wireless sensor networks and vehicular ad hoc networks, analysed from the aspects of network components, features and vulnerabilities. With the research motivations stated, Chapter 1 further provided contributions and the organization of this thesis.

Chapter 2 provided background studies of IoT security. A brief analysis on the IoT network vulnerability was given from the bottom layer to the top layer. Then this chapter presented the attack model with the attack behaviour and ability clearly clarified. To further protect IoT data security, we pointed out the fundamental requirements for IoT security, i.e., confidentiality, integrity, non-repudiation, and availability. As traditional cryptographic mechanisms are widely used to meet security requirements of Internet, this chapter briefly analysed the limitations of traditional cryptographic mechanisms in IoT. Related works on securing IoT data transmissions were also provided in Chapter 2 from aspects of source location privacy, secure data transmission protocols, and the analysis model. By analyzing existing security solutions, unsolved issues were pointed out to define the need for studying security protocols and analysis models in IoT.

Chapter 3 proposed an Anti-Pollution Source-Location Privacy scheme (AP-SLP) to tackle the conflict between privacy and authentication in IoT. It integrates a homomorphic authentication algorithm with network coding based dummy traffic. Homomorphic signatures filter out dummy and polluted traffic during transmission, while the network coding based dummy traffic conceals the traffic pattern from eavesdroppers. The keys probabilistically distributed for the homomorphic authentication algorithm further prevent internal attackers from distinguishing the real traffic from dummy. In this way, privacy and integrity are guaranteed in the proposed scheme. The enhanced data integrity not only relieves the traffic burden but also saves energy for transmissions and computations. The simulation results demonstrated that the proposed AP-SLP improves the message delivery rate and saves transmission energy by around 22% and 40% respectively, compared with previous network coding based solution.

Chapter 4 provided an opportunistic encrypted data transmission protocol to ensure data confidentiality in dynamic IoT networks, where the transmitter switches between backing off transmissions and changing keys, adapting to the different causes of a failed (re)transmission attempt. To avoid the extra communication overhead for finding common keys between neighbours, the proposed algorithm predistributes keys among nodes and tries keys with messages until a matched one is found at the receiver. Chapter 4 also proposed a new 3-dimensional (3D) Markov chain model, the accuracy of which is confirmed by simulation results, to characterize the proposed protocol and analyse its secure transmission ability. According to the analysis and simulation results, the proposed protocol can save the overhead for handshaking and achieve the highest transmission success rate among the previous key predistribution schemes. The key to releasing its potential is to increase the key matched possibility, which can be achieved by embracing the opportunistic routing protocol.

Chapter 5 proposed an opportunistic authentication protocol for protecting data integrity in decentralized IoT networks. The proposed protocol embraces opportunistic routing to tackle the fast-changing topology. In order to reduce the communication overhead for authentication, a node increasingly combines collision-free yet unauthenticated messages and a new message for digital signature or message authentication code (MAC) generation, while trying different keys on-the-fly. A three-dimensional (3D) Markov chain was proposed in Chapter 5 to capture interactions among collisions, key selections, and the lifetime of unauthenticated messages. Closed-form expressions for authentication rate, delay and throughput were derived based on the 3D Markov model. The analytical results, based on the 3D Markov chain, confirmed the tolerance of the proposed protocol against changing topologies compared to the prior art, as the proposed protocol doubles the authenticated throughput in the case of a short link duration. The performance comparisons between symmetric and asymmetric keys were also provided in terms of authentication performance and the resistance against collusion attacks.

Chapter 6 developed a unified analytic framework, a new four-dimensional (4D) model, to analyse a variety of on-the-fly authentication protocol designs. The dynamic topology was quantified as the link duration in this chapter. The general 4D Markov model captures unexpired messages between cycles of the protocols in the fourth dimension, in addition to the first three dimensions to capture a cycle of channel access and opportunistic authentication until either is the leading message of the cycle authenticated or exhausts the largest backoff window or the maximum number of keys allowed. Validated by simulations, the proposed model is general and is able to capture the impact of the link duration on a range of on-the-fly authentication protocols. Three on-the-fly authentication protocols, which couple opportunistic authentication and channel access to different extents, were proposed and compared in Chapter 6. The analysis revealed that on-the-fly authentication protocols incurs less severe collisions and their gains on authenticated throughput over the prior art can achieve 30%. Chapter 6 also pointed out that the cross-layer consideration to jointly design retransmissions and rekeying is key to achieving the significant gain of on-the-fly authentication in decentralized IoT networks.

7.2 Future Work

Our previous research mainly focused on providing security in the process of data transmission. The secured data is then stored in IoT infrastructures for further processing. However, the data stored in distributed infrastructures may be tampered, fabricated and deleted, leading to inconsistent and unavailable data service. Blockchain attracts worldwide attentions for its anti-tampering property in decentralized networks [141]. It is able to keep stored data permanently in a verifiable way, as the signatures of senders in the Blockchain transactions can guarantee the integrity and non-repudiation of the transactions while the hash chain structure of Blockchain ensures that any recorded data cannot be updated, even partly. Consensus protocol, the core component of Blockchain, enables Blockchain to maintain a distributed and consistent ledger without centralized coordination.

Recently, we have completed a comprehensive survey on existing Blockchain technologies with an emphasis on the IoT applications, where the benefits and the key challenges of Blockchain in IoT applications are investigated. As analysed in the survey, Blockchain has the potential to secure the integrity of storage and prevent data tampering in IoT applications [33, 60]. The peer-to-peer network setting of Blockchains is inherently suited for IoT networks which are typically distributed. Moreover, Blockchains can use changeable public keys as users' identities to preserve anonymity and privacy [58]. This is attractive to many IoT applications and services, especially those which need to keep confidential identities and privacy. While providing data security for IoT, Blockchain also encounters a number of critical challenges inherent to IoT, such as a huge number of IoT devices, non-homogeneous network structure, limited communication capacity, and dynamic topologies [26, 87]. Particularly, physical characteristics of IoT devices and networks, such as limited bandwidth and connectivity, non-trivial network topology, and unpredictable link delays, can cause discrepancy or inconsistency between the records maintained in a distributed fashion at different locations [122]. Therefore, existing Blockchain technologies can be inefficient for IoT applications.

Our work in the near future includes the specifically designed consensus protocol to benefit data-centric IoT applications and editable Blockchain to save storage for IoT applications.

The consensus protocol can be designed to reach data consensus by validating transaction data instead of the syntax of the transactions only. Sensor observations are highly correlated in the space domain, due to high density in the network topology. Moreover, the nature of the physical phenomenon constitutes the temporal correlation between consecutive observations of a sensor node [117, 105]. Therefore, spatial and temporal correlations, along with the collaborative nature of IoT, raise potentials to develop content-oriented consensus protocol. The correctness of sensory data can be cross-validated with sensory data from its neighbours and historical record.

The storage of IoT devices can be limited for the explosively growing size of a Blockchain ledger, as a huge number of IoT devices keep recording a large number of events in the long term. However, the data of some IoT applications will be meaningless after a constant duration. For example, the logistics record of food is meaningless after the food has been consumed. Hence, such data can be deleted from the Blockchain. Also, fraud actions and records on IoT-Blockchains raise demand for editable Blockchain technology without breaking the trust of stored data. As the “editability” is somewhat contrary to the inherent “immutability” of Blockchain, the editable Blockchain is required to guarantee secure conditions and records for any edit actions. Editable Blockchain is practical as some cryptographic algorithms, such as variations of the chameleon hash function, have been proposed to edit data with its hash value unchanged [13]. Specific designs and key management schemes of editable Blockchain in IoT applications will be our future work.

Bibliography

[1] "Internet of Things (IoT) Connected Devices Installed base Worldwide from 2015 to 2025 (in billions)," https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[2] "ITU Internet Report 2005: The Internet of Things," 2005.

[3] "IEEE Standard for Information Technology – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std 802.11p-2010*, pp. 1–51, July 2010.

[4] "IEEE Standard for Wireless Access in Vehicular Environments Security Services for Applications and Management Messages," *IEEE Std 1609.2-2013 (Revision of IEEE Std 1609.2-2006)*, pp. 1–289, April 2013.

[5] M. Abolhasan, J. Lipman, W. Ni, and B. Hagelstein, "Software-Defined Wireless Networking: Centralized, Distributed, or Hybrid?" *IEEE Netw.*, vol. 29, no. 4, pp. 32–38, July 2015.

[6] M. Ahlmeyer and A. M. Chircu, "Securing the Internet of Things: A Review," *Issues in information Systems*, vol. 17, no. 4, 2016.

[7] S. R. Ahmed, J. Hassan, B. J. D'Auriol, L. Heejo, L. Sungyoung, and Y. J. Song, "Achieving Network Level Privacy in Wireless Sensor Networks," *Sensors*, vol. 10, no. 3, pp. 1447–1472, 2010.

[8] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys Tut.*, vol. 17, no. 4, pp. 2347–2376, 2015.

[9] J. Almeida, S. Shintre, M. Boban, and J. Barros, "Probabilistic Key Distribution in Vehicular Networks with Infrastructure Support," in *Proceedings of IEEE Global Communications Conference (GLOBECOM'2012)*, 2012, pp. 973–978.

[10] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, "Toward a Statistical Framework for Source Anonymity in Sensor Networks," *IEEE Trans Mobile Computing*, vol. 12, no. 2, pp. 248–260, Feb 2013.

[11] E. Alsaadi and A. Tubaishat, "Internet of Things: Features, Challenges, and Vulnerabilities," *Int. J. Advanced Comput. Sci. Inf. Technol.*, vol. 4, no. 1, pp. 1–13, 2015.

[12] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet," in *26th USENIX Security Symp. (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[13] G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable Blockchain - or-Rewriting History in Bitcoin and Friends," in *2017 IEEE Eur. Symp. Secur. Privacy (EuroSP' 17)*, Apr. 2017, pp. 111–126.

[14] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A Survey,” *Comput. Netw.*, vol. 54, no. 15, pp. 2787 – 2805, 2010.

[15] J. Bernsen and D. Manivannan, “Unicast Routing Protocols for Vehicular Ad Hoc Networks: A Critical Comparison and Classification,” *Pervasive and Mobile Computing*, vol. 5, no. 1, pp. 1–18, 2009.

[16] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[17] C. Biao, L. Zhishu, and L. Zhen, “Threshold Secret Sharing Based Trust Security in Structured P2P Network,” in *2010 Third International Symposium on Intelligent Information Technology and Security Informatics.* IEEE, 2010, pp. 320–323.

[18] D. Boneh, D. Freeman, J. Katz, and B. Waters, “Signing a Linear Subspace: Signature Schemes for Network Coding,” *Public Key Cryptography*, pp. 68 – 87, 2009.

[19] T. Borgohain, U. Kumar, and S. Sanyal, “Survey of Security and Privacy Issues of Internet of Things,” *arXiv preprint arXiv:1501.02211*, 2015.

[20] J. E. Boritz, “Is Practitioners’ Views on Core Concepts of Information Integrity,” *Int. J. Accounting Inf. Syst.*, vol. 6, no. 4, pp. 260 – 279, 2005.

[21] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, “On the Performance of Secure Vehicular Communication Systems,” *IEEE Trans. Dependable and Secure Comput.*, vol. 8, no. 6, pp. 898–912, 2011.

[22] N. Chakchouk, “A Survey on Opportunistic Routing in Wireless Communication Networks,” *Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2214–2241, Fourthquarter 2015.

[23] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," in *IEEE Symposium on Security and Privacy (S&P'2003)*, 2003, pp. 197–213.

[24] C.-Y. Chen and H.-C. Chao, "A Survey of Key Distribution in Wireless Sensor Networks," *Security and Communication Networks*, vol. 7, no. 12, pp. 2495–2508, 2014.

[25] J. Chen, S. H. G. Chan, and S.-C. Liew, "Mixed-mode WLAN: the Integration of Ad Hoc Mode with Wireless LAN Infrastructure," in *Proceedings of IEEE Global Communications Conference (GLOBECOM'2003)*, Dec 2003, pp. 231–235.

[26] M. Chen, S. Mao, and Y. Liu, "Big Data: A Survey," *Mobile Netw. and Appl.*, vol. 19, no. 2, pp. 171–209, 2014.

[27] W. P. Chen and L. Sha, "An Energy-Aware Data-Centric Generic Utility based Approach in Wireless Sensor Networks," in *International Symposium on Information Processing in Sensor Networks*, 2004, pp. 215–224.

[28] X. Chen, K. Makki, Y. Kang, and N. Pissinou, "Sensor Network Security: A Survey," *IEEE Commun. Surveys Tut.*, vol. 11, no. 2, pp. 52–73, 2009.

[29] C. Cheng, J. Lee, T. Jiang, and T. Takagi, "Security Analysis and Improvements on Two Homomorphic Authentication Schemes for Network Coding," *IEEE Trans. Inf Forensics Security*, vol. 11, no. 5, pp. 993–1002, May 2016.

[30] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication (DSRC) Frequency

Band," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1501–1516, Oct 2007.

[31] W. Cho and H. S. Oh, "A Multi-Hop Communication Scheme for IEEE 802.11 p based V2V Communication Systems," *Communication and Networking*, pp. 26–33, 2011.

[32] M. Conti, J. Willemsen, and B. Crispo, "Providing Source Location Privacy in Wireless Sensor Networks: A Survey," *IEEE Commun. Surveys Tut.*, vol. 15, no. 3, pp. 1238–1280, Third 2013.

[33] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in Internet of Things: Challenges and Solutions," 2016. [Online]. Available: http://arxiv.org/abs/1608.05187

[34] W. Du, R. Wang, and P. Ning, "An Efficient Scheme for Authenticating Public Keys in Sensor Networks," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005, pp. 58–67.

[35] M. H. Eiza and Q. Ni, "An Evolving Graph-based Reliable Routing Scheme for VANETs," *IEEE Trans. Vehicul. Technol.*, vol. 62, no. 4, pp. 1493–1504, 2013.

[36] K. El Defrawy and G. Tsudik, "ALARM: Anonymous Location-Aided Routing in Suspicious MANETs," *IEEE Trans Mobile Computing*, vol. 10, no. 9, pp. 1345–1358, Sep. 2011.

[37] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, "VANET Security Surveys," *Computer Communications*, vol. 44, no. C, pp. 1–13, May 2014.

[38] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," in *Proc. ACM Conf. Computer and Commun. Security*, 2002, pp. 41–47.

[39] A. Esfahani, D. Yang, G. Mantas, A. Nascimento, and J. Rodriguez, "An Improved Homomorphic Message Authentication Code Scheme for RLNC-enabled Wireless Networks," in *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Dec 2014, pp. 80–84.

[40] D. Evans, "The Internet of Things: How the Next Evolution of the Internet is Changing Everything. 2011," http://www. cisco. com/web/about/ac79/docs/innov/IoT IBSG 0411FINAL. pdf, 2015.

[41] B. Fabian and O. Günther, "Security Challenges of the EPCglobal Network," *Commun. of the ACM*, vol. 52, no. 7, pp. 121–125, 2009.

[42] Y. Fan, J. Chen, X. Lin, and X. Shen, "Preventing Traffic Explosion and Achieving Source Unobservability in Multi-Hop Wireless Networks Using Network Coding," in *Proceedings of IEEE Global Communications Conference (GLOBECOM'2010)*, Dec 2010, pp. 1–5.

[43] Y. Fan, Y. Jiang, H. Zhu, J. Chen, and X. Shen, "Network Coding Based Privacy Preservation against Traffic Analysis in Multi-Hop Wireless Networks," *IEEE Trans Wireless Communications*, vol. 10, no. 3, pp. 834–843, March 2011.

[44] M. U. Farooq, M. Waseem, A. Khairi, and S. Mazhar, "A Critical Analysis on the Security Concerns of Internet of Things (IoT)," *International Journal of Computer Applications*, vol. 111, no. 7, 2015.

[45] G. Gan, Z. Lu, and J. Jiang, "Internet of Things Security Analysis," in *2011 Int. Conf. Internet Technol. and Appl.*, Aug 2011, pp. 1–4.

[46] C. Gentry, "Certificate-based Encryption and the Certificate Revocation Problem," in *International Conference on Theory and Applications of*

Cryptographic Techniques, 2003, pp. 272–293.

[47] P. Golle, D. Greene, and J. Staddon, “Detecting and Correcting Malicious Data in VANETs,” in *Proc. VANET*, 2004, pp. 29–37.

[48] K. Grover and A. Lim, “A Survey of Broadcast Authentication Schemes for Wireless Networks ,” *Ad Hoc Networks*, vol. 24, Part A, pp. 288 – 316, 2015.

[49] W. Gu, S. Chellappan, X. Bai, and H. Wang, “Scaling Laws of Key Predistribution Protocols in Wireless Sensor Networks,” *IEEE Trans. Inf Forensics Security*, vol. 6, no. 4, pp. 1370–1381, Dec 2011.

[50] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, “Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs,” *Cryptographic Hardware and Embedded Systems*, vol. 3156, pp. 119–132, 2004.

[51] B. Hagelstein, M. Abolhasan, D. Franklin, F. Safaei, and W. Ni, “Analytic performance model for state-based MAC layer cooperative retransmission protocols,” *IEEE Trans. on Mobile Comput.*, vol. 15, no. 1, pp. 32–44, Jan 2016.

[52] B. Hagelstein, M. Abolhasan, D. Franklin, F. Safaei, and W. Ni, “Analytic Performance Model for State-Based MAC Layer Cooperative Retransmission Protocols,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 1, pp. 32–44, Jan 2016.

[53] Y. Hao, Y. Cheng, C. Zhou, and W. Song, “A Distributed Key Management Framework with Cooperative Message Authentication in VANETs,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 3, pp. 616–629, Mar. 2011.

[54] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A Random Linear Network Coding Approach to Multicast,” *IEEE Trans Information Theory*, vol. 52, no. 10, pp. 4413–4430, Oct 2006.

[55] P. Kamat, Y. Zhang, W. Trappe, and C. Ozturk, “Enhancing Source-Location Privacy in Sensor Network Routing,” in *IEEE International Conference on Distributed Computing Systems (ICDCS’2005)*, 2005, pp. 599–608.

[56] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour, “A Survey of Routing Attacks in Mobile Ad Hoc Networks,” *Wireless Commun.*, vol. 14, no. 5, pp. 85–91, Oct 2007.

[57] M. Katagi and S. Moriai, “Lightweight Cryptography for the Internet of Things,” *Sony Corporation*, pp. 7–10, 2008.

[58] M. C. K. Khalilov and A. Levi, “A Survey on Anonymity and Privacy in Bitcoin-like Digital Cash Systems,” *IEEE Commun. Surveys Tut.*, pp. 1–1, 2018.

[59] M. Khatua and S. Misra, “D2D: Delay-Aware Distributed Dynamic Adaptation of Contention Window in Wireless Networks,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 2, pp. 322–335, Feb 2016.

[60] N. Kshetri, “Can Blockchain Strengthen the Internet of Things?” *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[61] M. T. Lazarescu, “Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications,” *IEEE Journal on Emerging and Selected Topics in Circuits and System*, vol. 3, no. 1, pp. 45–54, 2013.

[62] A. Le and A. Markopoulou, “Cooperative Defense Against Pollution Attacks in Network Coding Using SpaceMac,” *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 442–449, Feb 2012.

[63] J. S. Lee, Y. W. Su, and C. C. Shen, “A Comparative Study of Wireless

Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Proc. 33rd Annu. Conf. the IEEE Ind. Elect. Soc. (IECON'07)*, Nov 2007, pp. 46–51.

[64] A. K. Lenstra and E. R. Verheul, "Selecting Cryptographic Key Sizes," *Journal of Cryptology*, vol. 14, no. 4, pp. 255–293, 2001.

[65] K. Li, W. Ni, X. Wang, and R. P. Liu, "EPLA: Energy-Balancing Packets Scheduling for Airborne Relaying Networks," in *Proceedings of IEEE International Conference on Communications (ICC'2015)*, London, UK, June 2015, pp. 6246–6251.

[66] K. Li, W. Ni, X. Wang, R. P. Liu, S. S. Kanhere, and S. Jha, "Energy-Efficient Cooperative Relaying for Unmanned Aerial Vehicles," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1377–1386, June 2016.

[67] N. Li, N. Zhang, S. K. Das, and B. Thuraisingham, "Privacy Preservation in Wireless Sensor Networks: A State-of-the-art Survey," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1501–1514, 2009.

[68] P. Li, Q. Liu, C. Huang, J. Wang, and X. Jia, "Delay-Bounded Minimal Cost Placement Of Roadside Units In Vehicular Ad Hoc Networks," in *Proceedings of IEEE International Conference on Communications (ICC'2015)*, June 2015, pp. 6589–6594.

[69] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct 2017.

[70] X. Lin, X. Sun, X. Wang, C. Zhang, P. H. Ho, and X. Shen, "TSVC: Timed Efficient and Secure Vehicular Communications with Privacy Preserving," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4987–4998, Dec 2008.

[71] K. Liu and V. C. S. Lee, "RSU-based Real-Time Data Access in Dynamic Vehicular Networks," in *13th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2010, pp. 1051–1056.

[72] R. P. Liu, G. J. Sutton, and I. B. Collings, "A New Queueing Model for QoS Analysis of IEEE 802.11 DCF with Finite Buffer and Load," *IEEE Trans. Wireless Commun.*, vol. 9, no. 8, pp. 2664–2675, Aug. 2010.

[73] X. Liu, M. Zhao, S. Li, F. Zhang, and W. Trappe, "A Security Framework for the Internet of Things in the Future Internet Architecture," *Future Internet*, vol. 9, no. 3, 2017.

[74] M. M. E. A. Mahmoud and X. Shen, "A Cloud-Based Scheme for Protecting Source-Location Privacy against Hotspot-Locating Attack in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1805–1818, 2012.

[75] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of Threats to The Internet of Things," *IEEE Commun. Surveys Tut.*, pp. 1–1, 2018.

[76] S. S. Manvi and S. Tangade, "A Survey on Authentication Schemes in VANETs for Secured Communication," *Vehicular Communications*, vol. 9, pp. 19–30, 2017.

[77] J. Manyika, *The Internet of Things: Mapping the Value beyond the Hype.* McKinsey Global Institute, 2015.

[78] V. Matyas and J. Kur, "Conflicts between Intrusion Detection and Privacy Mechanisms for Wireless Sensor Networks," *IEEE Security Privacy*, vol. 11, no. 5, pp. 73–76, Sept 2013.

[79] K. Mehta, D. Liu, and M. Wright, “Protecting Location Privacy in Sensor Networks against a Global Eavesdropper,” *IEEE Trans. Mobile Comput.*, vol. 11, no. 2, pp. 320–336, Feb 2012.

[80] K. Mehta, D. Liu, and M. Wright, “Location Privacy in Sensor Networks Against a Global Eavesdropper,” in *IEEE International Conference on Network Protocols (ICNP’2007).*, Oct 2007, pp. 314–323.

[81] J. Molina-Gil, P. Caballero-Gil, and C. Caballero-Gil, “Aggregation and Probabilistic Verification for Data Authentication in VANETs,” *Information Sciences*, vol. 262, no. C, pp. 172–189, Mar. 2014.

[82] A. Mosenia and N. K. Jha, “A Comprehensive Study of Security of Internet-of-Things,” *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 4, pp. 586–602, Oct 2017.

[83] N. Namvar, W. Saad, N. Bahadori, and B. Kelley, “Jamming in the Internet of Things: A Game-Theoretic Perspective,” in *Proceedings of IEEE Global Communications Conference (GLOBECOM’2016)*, 2016, pp. 1–6.

[84] W. Ni, M. Abolhasan, B. Hagelstein, R. P. Liu, and X. Wang, “A New Trellis Model for MAC Layer Cooperative Retransmission Protocols,” *IEEE Trans. Vehicul. Technol.*, vol. 66, no. 4, pp. 3448–3461, April 2017.

[85] W. Ni, I. B. Collings, and R. P. Liu, “Relay Handover and Link Adaptation Design for Fixed Relays in IMT-Advanced Using a New Markov Chain Model,” *IEEE Trans. Vehicul. Technol.*, vol. 61, no. 4, pp. 1839–1853, May 2012.

[86] W. Ni, I. B. Collings, and R. P. Liu, “Decentralized User-Centric Scheduling with Low Rate Feedback for Mobile Small Cells,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6106–6120, December 2013.

[87] C. O'Connor, "What Blockchain Means for You, and the Internet of Things," Feb. 2017. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/watson-iot-blockchain/

[88] C. Ozturk, Y. Zhang, and W. Trappe, "Source-Location Privacy in Energy-Constrained Sensor Network Routing," in *ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2004, pp. 88–93.

[89] J. Pacheco and S. Hariri, "IoT Security Framework for Smart Cyber Infrastructures," in *IEEE International Workshops on Foundations and Applications of Self Systems*, 2016, pp. 242–247.

[90] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for the Internet of Things: A Survey," *IEEE Commun. Surveys Tut.*, vol. 16, no. 1, pp. 414–454, First 2014.

[91] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," in *IEEE Symposium Security and Privacy, (S&P'2000*, 2000, pp. 56–73.

[92] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: Security Protocols for Sensor Networks," *Wireless Netw.*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[93] J. S. Perry, "Anatomy of an IoT Malware Attack," https://www.ibm.com/developerworks/library/iot-anatomy-iot-malware-attack/.

[94] J. Petit, F. Schaub, M. Feiri, and F. Kargl, "Pseudonym Schemes in Vehicular Networks: A Survey," *Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 228–255, 2015.

[95] K. Prasad and B. Rajan, “Single-generation Network Coding for Networks with Delay,” in *IEEE International Conference on Communications*, 2009, pp. 1 – 6.

[96] M. Pticek, V. Podobnik, and G. Jezic, “Beyond the Internet of Things: The Social Networking of Machines,” *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 6, p. 8178417, 2016.

[97] F. Qu, Z. Wu, F. Wang, and W. Cho, “A Security and Privacy Review of VANETs,” *IEEE Trans. Intell. Transport Syst.*, vol. 16, no. 6, pp. 2985–2996, Dec 2015.

[98] M. Ramkumar, “The Subset Keys and Identity Tickets (SKIT) Key Distribution Scheme,” *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 39–51, Mar. 2010.

[99] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, “NB-IoT System for M2M Communication,” in *2016 Proc. IEEE Wireless Commun. and Netw. Conf. Workshops (WCNCW' 16)*, April, pp. 1–5.

[100] M. Raya and J.-P. Hubaux, “Securing Vehicular Ad Hoc Networks,” *Journal of Computer Security*, vol. 15, pp. 39–68, Dec. 2007.

[101] C. Ren, X. Lyu, W. Ni, H. Tian, and R. P. Liu, “Distributed Online Learning of Fog Computing under Non-uniform Device Cardinality,” *IEEE Internet Things J.*, pp. 1–1, 2018.

[102] D. Ren, S. Du, and H. Zhu, “A Novel Attack Tree Based Risk Assessment Approach for Location Privacy Preservation in the VANETs,” in *Proceedings of IEEE International Conference on Communications (ICC'2011)*, 2011, pp. 1–5.

[103] J. H. Ren, Y. K. Hsiao, and Y. F. Liu, “Secure Communication Scheme of VANET with Privacy Preserving,” in *IEEE International Conference on Parallel and Distributed Systems*, 2011, pp. 654–659.

[104] R. Roman, P. Najera, and J. Lopez, “Securing the Internet of Things,” *Comput.*, vol. 44, no. 9, pp. 51–58, Sep. 2011.

[105] D. Romero, V. N. Ioannidis, and G. B. Giannakis, “Kernel-Based Reconstruction of Space-Time Functions on Dynamic Graphs,” *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 856–869, Sep. 2017.

[106] K. Rose, S. Eldridge, and L. Chapin, “The Internet of Things: An Overview,” *The Internet Society*, pp. 1–50, 2015.

[107] M. Rudack, M. Meincke, and M. Lott, “On the Dynamics of Ad Hoc Networks for Inter Vehicle Communications (IVC),” in *Proc. ICWN*, 2002.

[108] M. Shao, Y. Yang, S. Zhu, and G. Cao, “Towards Statistically Strong Source Anonymity for Sensor Networks,” in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM’08)*, April 2008.

[109] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, “Advanced Lightweight Encryption Algorithms for IoT Devices: Survey, Challenges and Solutions,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2017.

[110] X. Sun and X.-M. Li, “Study of the Feasibility of VANET and its Routing Protocols,” in *International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM’08)*. IEEE, 2008, pp. 1–4.

[111] X. Sun, X. Lin, and P.-H. Ho, “Secure Vehicular Communications Based on Group Signature and ID-Based Signature Scheme,” in *Proceedings of IEEE International Conference on Communications (ICC’2007)*, 2007, pp. 1539–1545.

[112] Z. Tao and S. Panwar, "Throughput and Delay Analysis for the IEEE 802.11e Enhanced Distributed Channel Access," *IEEE Trans. Commun.*, vol. 54, no. 4, pp. 596–603, April 2006.

[113] I. Tinnirello and G. Bianchi, "Rethinking the IEEE 802.11e EDCA Performance Modeling Methodology," *IEEE/ACM Trans. Networking*, vol. 18, no. 2, pp. 540–553, April 2010.

[114] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, L. T. Yang *et al.*, "Data Mining for Internet of Things: A Survey," *IEEE Commun. Surveys Tut.*, vol. 16, no. 1, pp. 77–97, 2014.

[115] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-cluster-based IEEE 802.11 p and LTE Hybrid Architecture for VANET Safety Message Dissemination," *IEEE Trans. Vehicul. Technol.*, vol. 65, no. 4, pp. 2621–2636, 2016.

[116] O. Vermesan, P. Friess, P. Guillemin *et al.*, "Internet of Things Strategic Research Roadmap," *Internet of Things-Global Technological and Societal Trends*, vol. 1, no. 2011, pp. 9–52, 2011.

[117] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-Temporal Correlation: Theory and Applications for Wireless Sensor Networks," *Comput. Netw.*, vol. 45, no. 3, pp. 245 – 259, 2004, in Memory of Olga Casals.

[118] C. Wang, G. Wang, W. Zhang, and T. Feng, "Reconciling Privacy Preservation and Intrusion Detection in Sensory Data Aggregation," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM'11)*, April 2011, pp. 336–340.

[119] E. K. Wang, Y. Ye, and X. Xu, "Location-Based Distributed Group Key Agreement Scheme for Vehicular Ad Hoc Network," *International Journal of Distributed Sensor Networks*, vol. 2014, no. 4, pp. 1–8, 2014.

[120] H. Wang, R. P. Liu, W. Ni, W. Chen, and I. B. Collings, “A New Analytical Model for Highway Inter-Vehicle Communication Systems,” in *Proceedings of IEEE International Conference on Communications (ICC’2014)*, Sydney, Australia, June 2014, pp. 2581–2586.

[121] H. Wang, R. P. Liu, W. Ni, W. Chen, and I. B. Collings, “VANET Modeling and Clustering Design Under Practical Traffic, Channel and Mobility Conditions,” *IEEE Trans. Commun.*, vol. 63, no. 3, pp. 870–881, 2015.

[122] K. H. Wang and B. Li, “Group Mobility and Partition Prediction in Wireless Ad-Hoc Networks,” in *Proceedings of IEEE International Conference on Communications (ICC’2002)*, vol. 2, pp. 1017–1021 vol.2.

[123] X. Wang, W. Ni, K. Zheng, R. P. Liu, and X. Niu, “Virus Propagation Modeling and Convergence Analysis in Large-Scale Networks,” *IEEE Trans. Inf Forensics Security*, vol. 11, no. 10, pp. 2241–2254, Oct 2016.

[124] X. Wang, K. Zheng, X. Niu, B. Wu, and C. Wu, “Detection of Command and Control in Advanced Persistent Threat based on Independent Access,” in *Proceedings of IEEE International Conference on Communications (ICC’2016)*, May 2016, pp. 1–6.

[125] A. Wasef and X. Shen, “EMAP: Expedite Message Authentication Protocol for Vehicular Ad Hoc Networks,” *IEEE Trans. on Mobile Comput.*, vol. 12, no. 1, pp. 78–89, 2013.

[126] R. H. Weber, “Internet of Things – New Security and Privacy Challenges,” *Comput. Law and Secur. Review*, vol. 26, no. 1, pp. 23 – 30, 2010.

[127] Y. Xi, K. Sha, W. Shi, L. Schwiebert, and T. Zhang, “Enforcing Privacy Using Symmetric Random Key-Set in Vehicular Networks,” in *Eighth*

International Symposium on Autonomous Decentralized Systems (ISADS'07), March 2007, pp. 344–351.

[128] Y. Xiao, "Performance Analysis of Priority Schemes for IEEE 802.11 and IEEE 802.11e Wireless LANs," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1506–1515, July 2005.

[129] L. D. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[130] R. Xu, R. Wang, Z. Guan, L. Wu, J. Wu, and X. Du, "Achieving Efficient Detection Against False Data Injection Attacks in Smart Grid," *IEEE Access*, vol. 5, pp. 13 787–13 798, 2017.

[131] O. Yagan, "Performance of the Eschenauer–Gligor Key Distribution Scheme under an ON/OFF Channel," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3821–3835, 2012.

[132] G. Yan and S. Olariu, "A Probabilistic Analysis of Link Duration in Vehicular Ad Hoc Networks," *IEEE Trans. Intell. Transport Syst.*, vol. 12, no. 4, pp. 1227–1236, Dec 2011.

[133] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, and G. Cao, "Towards Event Source Unobservability with Minimum Network Traffic in Sensor Networks," in *Proceedings of the first ACM conference on Wireless network security*. ACM, 2008, pp. 77–88.

[134] L. Yao, L. Kang, P. Shang, and G. Wu, "Protecting the Sink Location Privacy in Wireless Sensor Networks," *Personal and Ubiquitous Computing*, vol. 17, no. 5, pp. 883–893, 2013.

[135] 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.

[136] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular Ad Hoc Networks (VANETS): Status, Results, and Challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[137] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil Attacks and their Defenses in the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 372–383, 2014.

[138] M. Zhang, F. Sun, and X. Cheng, "Architecture of Internet of Things and its Key Technology Integration based-on RFID," in *2012 Fifth International Symposium on Computational Intelligence and Design*, vol. 1, no. 4, 2012, pp. 294–297.

[139] J. Zhao, "On Resilience and Connectivity of Secure Wireless Sensor Networks Under Node Capture Attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 557–571, March 2017.

[140] J. Zhao, O. Yagan, and V. Gligor, " k -Connectivity in Random Key Graphs with Unreliable Links," *IEEE Trans. Inf. Theory*, vol. 61, no. 7, pp. 3810–3836, July 2015.

[141] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, "Blockchain Challenges and Opportunities: A Survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352 – 375, 2018.

[142] D. Zhu, X. Yang, and W. Yu, "Towards Effective Defense against Pollution Attacks on Network Coding," in *Proceedings of IEEE International Conference on Communications (ICC'2012)*, June 2012, pp. 830–834.

[143] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "IoT Gateway: Bridging Wireless Sensor Networks into Internet of Things," in *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2011, pp. 347–352.

[144] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From Today's INTRAnet of Things to a Future INTERnet of Things: a Wireless- and Mobility-related View," *IEEE Wireless Commun.*, vol. 17, no. 6, pp. 44–51, December 2010.