Efficient Authentication, Node Clone Detection, and Secure Data Aggregation for Sensor Networks

by

Zhijun Li

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Sensor networks are innovative wireless networks consisting of a large number of low-cost, resource-constrained sensor nodes that collect, process, and transmit data in a distributed and collaborative way. There are numerous applications for wireless sensor networks, and security is vital for many of them. However, sensor nodes suffer from many constraints, including low computation capability, small memory, limited energy resources, susceptibility to physical capture, and the lack of infrastructure, all of which impose formidable security challenges and call for innovative approaches. In this thesis, we present our research results on three important aspects of securing sensor networks: lightweight entity authentication, distributed node clone detection, and secure data aggregation.

As the technical core of our lightweight authentication proposals, a special type of circulant matrix named circulant-P2 matrix is introduced. We prove the linear independence of matrix vectors, present efficient algorithms on matrix operations, and explore other important properties. By combining circulant-P2 matrix with the learning parity with noise problem, we develop two one-way authentication protocols: the innovative LCMQ protocol, which is provably secure against all probabilistic polynomial-time attacks and provides remarkable performance on almost all metrics except one mild requirement for the verifier's computational capacity, and the $\mathrm{HB}^{\mathcal{C}}$ protocol, which utilizes the conventional HB-like authentication structure to preserve the bit-operation only computation requirement for both participants and consumes less key storage than previous HB-like protocols without sacrificing other performance. Moreover, two enhancement mechanisms are provided to protect the HB-like protocols from known attacks and to improve performance. For both protocols, practical parameters for different security levels are recommended. In addition, we build a framework to extend enhanced HB-like protocols to mutual authentication in a communication-efficient fashion.

Node clone attack, that is, the attempt by adversaries to add one or more nodes to the network by cloning captured nodes, imposes a severe threat to wireless sensor networks. To cope with it, we propose two distributed detection protocols with difference tradeoffs on network conditions and performance. The first one is based on distributed hash table, by which a fully decentralized, key-based caching and checking system is constructed to deterministically catch cloned nodes in general sensor networks. The protocol performance of efficient storage consumption and high security level is theoretically deducted through a probability model, and the resulting equations, with necessary adjustments for real application, are supported by the simulations. The other is the randomly directed exploration protocol, which presents notable communication performance and minimal storage consumption by an elegant probabilistic directed forwarding technique along with random initial direction and border determination. The extensive experimental results uphold the protocol design and show its efficiency on communication overhead and satisfactory detection probability.

Data aggregation is an inherent requirement for many sensor network applications, but designing secure mechanisms for data aggregation is very challenging because the aggregation nature that requires intermediate nodes to process and change messages, and the security objective to prevent malicious manipulation, conflict with each other to a great extent. To fulfill different challenges of secure data aggregation, we present two types of approaches. The first is to provide cryptographic integrity mechanisms for general data aggregation. Based on recent developments of homomorphic primitives, we propose three integrity schemes: a concrete homomorphic MAC construction, homomorphic hash plus aggregate MAC, and homomorphic hash with identity-based aggregate signature, which provide different tradeoffs on security assumption, communication payload, and computation cost. The other is a substantial data aggregation scheme that is suitable for a specific and popular class of aggregation applications, embedded with built-in security techniques that effectively defeat outside and inside attacks. Its foundation is a new data structure—secure Bloom filter, which combines HMAC with Bloom filter. The secure Bloom filter is naturally compatible with aggregation and has reliable security properties. We systematically analyze the scheme's performance and run extensive simulations on different network scenarios for evaluation. The simulation results demonstrate that the scheme presents good performance on security, communication cost, and balance.

Acknowledgements

I would like to express tremendous gratitude to my supervisor, Prof. Guang Gong, for her guidance and supports in the past four years. I especially appreciate the cares, patience, and encouragements that she gives all of us. I would like to thank Prof. Radha Poovendran at the University of Washington for serving as my external examiner and insightful comments. I am also grateful to my committee members: Prof. Anwar Hasan, Prof. Douglas Stinson, and Prof. Paul Ward at the University of Waterloo for all of their quality time and valuable helps. It is a privilege to have such a great committee.

I want to thank Prof. Zhiguan Qin at the University of Electronic Science and Technology of China, for leading me into the fantastic network security research field and motivating me to pursue the study in Canada. His charming personality presents a wonderful role model for me. I am indebted to Prof. Kefei Cheng at the Shanghai Jiaotong University for his supports on my academic career. I also want to thank Prof. Alfred Menezes, Prof. David Jao, Prof. Ian Goldberg, and Prof. Pin-Han Ho at the University of Waterloo for their inspiring discussions.

There are many other people who helped me in one way or another. I am thankful to my lab colleagues in the Communication Security Laboratory: Dr. Xinxin Fan, Dr. Honggang Hu, Anuchart Tassanaviboon, Qi Chai, Wen Hong, Kalikinkar Mandal, Yiyuan Luo, Zilong Wang, Fei Huo, and Bo Zhu. It is really a pleasure to work with them. Moreover, I am extremely grateful to my friends who make my Canada life filled with joyce, with special thanks to Zhuo Zhang, Yang Liu, Jun Chen, Ying Liu, Enchen Dong, Xiaofen Wu, Qin Chen, Emma Shen, and Yongqin Luo. Without them, those days would have been much less enjoyable.

Last but not least, I would like to thank my parents for their unconditional love and my parents-in-laws for helping taking care of my son. Most of all, I am thankful to my wife for all the sacrifice.

To my son

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Advances in electronics and wireless communication technologies have enabled the development of large-scale wireless sensor networks (WSNs) that consist of distributed, autonomous, low-power, low-cost, small-size sensor nodes to collect information and cooperatively transmit data through infrastructureless wireless networks. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance, and then the progresses on miniaturization, low-cost circuit design, simple low-power wireless communication system, and improved small-scale energy supplies have enabled extensive application areas of WSNs, including environment and habitat monitoring, health-care applications, home automation, traffic control, etc..

Security plays a fundamental role in many wireless sensor network applications. Due to WSNs' unique characteristics, security techniques used in conventional networks cannot be directly applied to sensor networks. First of all, sensor nodes are very sensitive of manufacturing cost. Consequently, most sensor nodes are resource constrained in terms of energy, memory, computation, and communication capabilities. Normally, sensor nodes are powered by batteries, and recharging batteries is infeasible in many circumstances. Then energy consumption becomes a primary consideration for most sensor network protocols. Second, sensor nodes may be deployed in public hostile locations without attendance, which makes sensor nodes vulnerable to a variety of physical attacks by adversaries. Generally, adversaries are assumed to be able to undetectably take control of certain portion of sensor nodes and extract all secret data in the nodes. Subsequently, previously legitimate nodes may turn into being malicious. Furthermore, the scale of sensor networks is considerably large, and the network topology is dynamically adjusted, because some nodes may die from running out of energy or failure, and new nodes need to join the network to maintain desirable functionality. At last, sensor networks use wireless transmission channels without infrastructure supports and most of communications are delivered in an ad-hoc, multi-hop manner. All of them impose formidable challenges such that existing security mechanisms

are inadequate and new approaches are demanded.

In this thesis, we present our results on three important aspects of securing sensor networks: lightweight entity authentication algorithms, distributed node clone detection protocols, and secure data aggregation schemes. This chapter starts with a general discussion on sensor networks architectures in Section 1.1. Then we briefly state sensor networks applications and introduce two major standards related to WSNs in Section 1.2. After that, we summarize influential network modeling methods of supporting sensor networks research in Section 1.3, followed by an instructive illumination of simulations tools for wireless sensor networks security protocols in Section 1.4. Section 1.5 provides the research motivations and related work. Finally, we outline the organization of the thesis and list our contributions in Section 1.6.

1.1 Sensor Networks Architectures

Typically, a wireless sensor network is composed of a base station and hundreds and thousands of sensor nodes, as depicted in Figure 1.1. Sensor nodes both collect and forward information, and they are equipped with batteries, sensors, data processing units of restricted computation capability, limited memory space, and short-range radio communication. On the other hand, the base station releases task commands, collects network reports, and serves as the gateways to other networks or systems, with abundant data processing/storage centers, or access points for human interface. In general, the base station has many orders of magnitude more powerful than ordinary sensor nodes. Even though individual nodes only possess basic processing capacities, through the collaboration of a large scale of networked nodes by careful protocol design, sensor networks can perform advanced and sophisticated functionalities. As a common assumption in security protocols, the base station is believed to be trustworthy and to be tamper resistant, whereas low-cost sensor nodes are subject to a variety of attacks. WSNs are dynamic in the sense that radio range and network connectivity change over time; some sensor nodes die due to failure or power exhaustion, and new sensor nodes may be added to the networks.

There are different settings about wireless sensor networks architectures, which substantially shape system design and affect protocol performance.

- **Hierarchical Structure vs. Distributed Structure**

  In many scenarios, sensor nodes are organized as a hierarchical structure. They are grouped into a number of clusters controlled by some high-ranked nodes which play a particular role denoted as cluster heads. Member nodes for local sensing and intra-cluster forwarding are associated with a cluster via a one-hop or multi-hop link to connect to cluster heads. After gathering or aggregating localized sensing

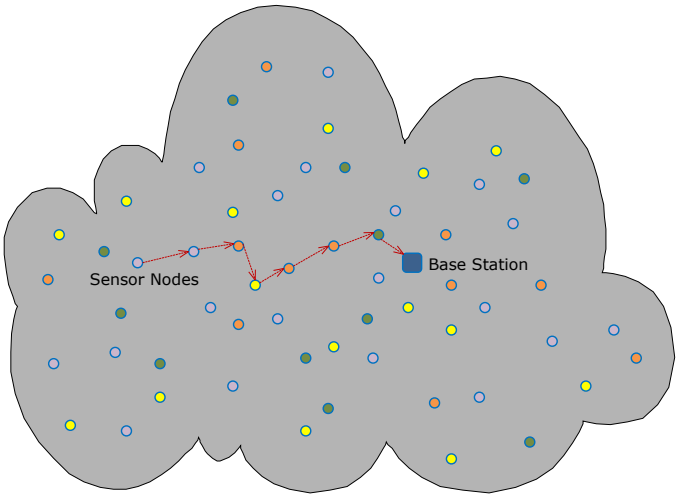


Figure 1.1: A wireless sensor network

information from their cluster members, the cluster heads transmit results to the base station. This is the case of two-level hierarchy, and there may exist several layers of clustering. On the other hand, we may think a centralized sensor network as a one-level hierarchy, in which a base station is necessary and vital for network management and protocol operations.

In contrast, there is no concept of cluster or rank in a distributed structure, similar to but not completely identical to P2P (peer-to-peer) networks. Once sensor nodes are deployed, they scan their radio coverage area to figure out neighbors and manage to form a fully distributed network. The base station does not participate in ordinary network management and may only play a supportive role for protocol executions, mainly for command issuing and result retrievals. Instead, sensor nodes sustain the network and carry out protocol procedures in an autonomous manner. From time to time, nodes may perform different tasks, but essentially they are equal.

- **Homogeneous Nodes vs. Heterogeneous Nodes**

  In a homogeneous system, all nodes possess the same level capacities of communication, computation, and storage. In this kind of sensor network, if there are varied roles for senor nodes, the overall balance on protocol requirements and energy consumption is an important metric of evaluating protocol applicability. By contrast, heterogeneous wireless nodes are armed with various transport mediums providing different ranges of coverage and distinct specifications including CPU, memory, and power supply to meet specific needs. This extra facility may provide a more flexible system design background for hierarchical sensor networks.

- **Multi-Hop or One-Hop Connectivity**

  For general sensor networks, the communication among nodes and base station are through multi-hop: nodes need to organized an ad-hoc wireless network to deliver messages. In some particular cases, it is assumed that the message transmission can be achieved in one-hop. In other words, for centralized sensor networks, every nodes has a direct link with the base station; for distributed networks, all nodes form a complete graph. This assumption is hardly satisfied in reality, but it might be useful in a hierarchical network for local intra-cluster nodes communication.

- **End-to-End vs. Hop-by-Hop**

  Those two principles may apply to message transport or security services layer, with subtly different implications. Any communications ultimately are for end-to-end data transmission, and they are implemented by hop-by-hop buffering-and-forwarding in a multi-hop wireless sensor networks. From the perspective of sensor network message transport layer, it is very fragile and probably infeasible to maintain an end-to-end connection between two communication participants. Instead, connections are established only on demand. Therefore, it is generally preferable to mainly consider hop-by-hop transport while end-to-end connectivity is only available intermittently. As for security protocols, those principles reflect different achievable service layers. Generally speaking, analogue to other networks, end-to-end security is more desired, and hop-by-hop security mechanisms may play an auxiliary role for overall system security. However, in some applications, it might be extremely challenging to design acceptable end-to-end security protocols due to reality constraints; henceforth, hop-by-hop security mechanisms may partially fulfill specific security objects and also can serve as an enlightening starting-point for further complete approach in future.

- **One-to-One, Many-to-One, and One-to-Many Communications**

  According to the sizes of intended senders and receivers from the high level of abstraction and modeling, communication in wireless sensor networks can be classified as one-to-one, one-to-many, and many-to-one. One-to-one communication is the basic case for the base station interacting with an individual node and a node reporting data to the base station, but frequent communication of many pairs of distant nodes rarely happens in reality and usually is conceived as an inappropriate requirement for protocol implementation.

  Unlike other networks, many-to-one is a very important communication scenario for sensor networks. One of main advantages of employing sensor networks is to combine a great number of nodes readings into generalized, more accurate reports, and this aggregating process probably or even preferably takes place during message transmission. In fact, effective many-to-one message transmission and aggregation

mechanisms have been and remain an active research topic for sensor networks. One-to-many is also a common scenario for regular network communication, such as the base station commanding the whole network and a node sending some information to a set of nodes to fulfill specific protocol functionalities.

- **Unicast, Broadcast, Multicast, and Anycast**

  Those characterizations are from the the aspect of message routing and delivery mechanisms on wireless sensor networks communication. Intuitively, one-to-one multi-hop communication can be achieved by unicast routing protocols. A lot of one-to-many transmissions in WSNs are conducted by broadcasting, especially for the base station as the sender to communicate with all the network. In addition, routing establishment often starts with initial flooding to pinpoint the recipient. However, this transmission mechanism should only be casually used by nodes, since it is very energy-consuming. Instead, multicast might be more suited for sensor nodes one-to-many communication in general cases. Alternatively for some one-to-many communication scenarios where it is only required to ensure at least one of targeted receivers to retrieve data, anycast mechanism can accomplish this intension.

- **Stationary Network vs. Mobile Network**

  This division is based on sensor nodes' mobility. Generally speaking, WSNs are data-centric networks with emphasis on message in-network processing by a large number of low-cost sensor nodes, and then nodes mobile capacity is much less significant in wireless sensor networks than mobile ad-hoc networks (MANETs). In addition, frequently changing network topology due to nodes movements may consume previous energy of nodes greatly. Therefore, it is reasonable to assume that sensor nodes remain relatively static during a period of protocol procedure, while protocols that adapt to high mobility environments are only discussed in special cases.

1.2 Sensor Networks Applications and Standards

1.2.1 Applications

There are diverse applications of wireless sensor networks [13, 85, 44], such as Great Duck (bird observation on Great Duck island), Cattle Herding, Bathymetry, ZebraNet, Glacier Monitoring, Ocean Water Monitoring, Cold Chain Management, Grape Monitoring, Rescue of Avalanche Victims, Vital Sign Monitoring, Power Monitoring, Self-healing Mine Field and Sniper Localization, Parts Assembly, and Tracking Military Vehicles. According to the deployment areas, the WSN applications can be categorized in the following fields:

military, environmental, industrial, agricultural, location oriented, public safety oriented, airport oriented, automotive, emergency handling, medical and oceanic.

Among them, military and medical solutions might be the two of the most security-oriented application fields of wireless sensor networks. Military sensing networks are designed to detect and gain as much information as possible about enemy movements, explosions, and other phenomena. Typically, wireless sensor nodes are integrated with military command, control, communications, computing, intelligence, surveillance, reconnaissance and targeting systems. Examples of military wireless sensor network applications are battlefield surveillance, guidance systems for intelligent missiles, detection of attacks by weapons of mass destruction such as nuclear, biological, or chemical, and other monitoring applications. Due to the nature of the military, it is apparent that those applications could not be mounted without appropriate security assurance.

Many medical systems are equipped with a large number of tiny, non-invasive sensors, located on or close to the patient's body, for health monitoring purposes. Such systems have been designed to measure diverse physiological values, including blood pressure, blood oxygen level, heart activities, activity recognition, etc., and are available in many different forms, including wrist wearable, ambulatory devices and as part of biomedical smart clothes. The term of body sensor network (BSN) [154] is coined to represent this kind of application. A number of intelligent physiological sensors are integrated into a wearable wireless body sensor network, which can be used for computer-assisted rehabilitation and even early detection of medical conditions. Those applications imply that outpatients can be monitored from their homes, freeing space in hospital beds. As the physiological patient data is legally required to be kept private, the implemented networks must invoke strong, long living security protocols in the sense that the methods and mechanism for medical data should guarantee the security being maintained over the life time of the individual members.

1.2.2 Standards

A number of standards have been ratified or are under development for wireless sensor networks communication. Among them, predominate are IEEE 802.15.4 standard [6] and ZigBee specification [7].

IEEE 802.15.4 Standard

IEEE 802.15.4 is a standard established by IEEE 802.15 working group to define the physical and medium access control (MAC) layers for low-rate wireless personal area networks (LR-WPAN). It is aimed to offer the fundamental lower network layers for a type of wireless personal area network which focuses on low-cost, low-speed ubiquitous communication

between devices. The emphasis is on very low communication cost and low power consumption of nearby devices with little or no underlying infrastructure, which makes it particularly suited for wireless sensor networks.

IEEE 802.15.4 standard includes a link layer security protocol to address four basic security services—access control, message integrity, message confidentiality, and replay protection. The Advanced Encryption Standard (AES) [52] with 128-bit key is employed in this standard, where the CTR (Counter) mode [114] is used for encryption only, the CBC-MAC (Cipher Block Chaining Message Authentication code) mode [114] for authentication only, and the CCM (Counter with CBC-MAC) mode [114] for both encryption and authentication. However, other necessary security mechanisms, such as how to manage keys and what kind of authentications policies to apply, are undefined in this standard and have to be provided by upper layer protocols, one of which is ZigBee.

ZigBee Specification

Built upon IEEE 802.15.4, ZigBee specifies a suite of high layer communication protocols with the same application intention of using small, low-power digital radios for low-rate wireless personal area networks. The specification is maintained by the ZigBee Alliance, which is founded by a group of companies. Its main purposes are constructing a network topology, providing application services, and facilitating communication features such as encryption and authentication. The network layer (NWK) is in charge of organizing and providing routing over an IEEE 802.15.4 multi-hop wireless network, while the application layer (APL) aims at providing a framework for distributed application development and communication. In accordance with the OSI model, IEEE 802.15.4 and Zigbee are layered in Figure. 1.2.

ZigBee makes use of all the basic security components in IEEE 802.15.4 standard. In addition, the ZigBee security specification contains mechanisms for key establishment, key transport, frame protection, and trust management. There are three types of keys employed: the master key, the link key and the network key. Moreover, the ZigBee specification defines the role of a trust center as a device that would be trusted by all other devices on the network and would distribute keys for the purpose of network and end-to-end application configuration management.

Security plays a fundamental role in many wireless sensor network applications. Due to WSNs' unique characteristics, security techniques used in conventional networks cannot be directly applied to WSNs. First, sensor nodes are very sensitive of manufacturing cost since sensor networks consist of a large number of sensor nodes. Consequently, most sensor nodes are resource restrained in terms of energy, memory, computation, and communication capabilities. Normally, sensor nodes are powered by batteries, and recharging batteries

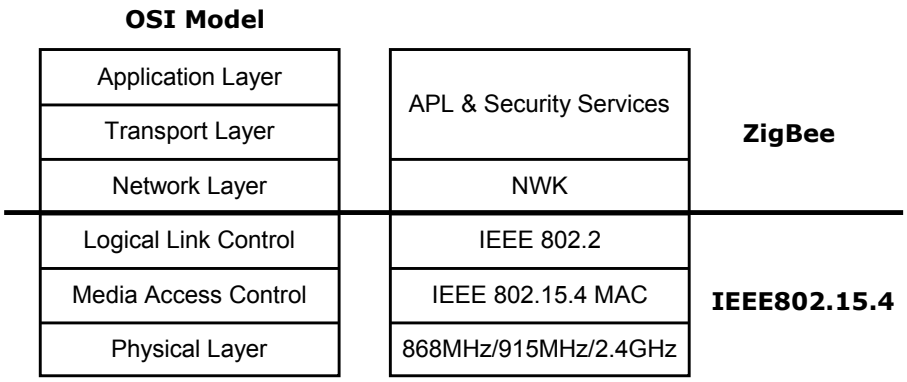


Figure 1.2: IEEE 802.15.4 and ZigBee protocol layers

are infeasible in many circumstances. Then energy consumption becomes a primary consideration for most sensor network protocols. Second, Sensor nodes may be deployed in public hostile locations, which makes sensor nodes vulnerable to physical attacks by adversaries. Generally, adversaries are assumed to be able to undetectably take control of certain portion of sensor nodes and extract all secret data in the nodes. Furthermore, the scale of sensor networks is considerably large, and the network topology is dynamically adjusted, because some nodes may die from running out of energy or failure, and new nodes may join the network to maintain desirable functionality. At last, sensor networks use insecure wireless communication channel and lack infrastructure. As a result, existing security mechanisms are inadequate, and new approaches are demanded.

1.3 Modeling Sensor Networks

How to model WSNs properly is fundamental to sensor network protocol design and performance analysis. Intuitively, we may think of a wireless sensor network as a graph, nodes as vertices, and links as edges. Generally, sensor node are randomly deployed in a targeted area, and the number of nodes in a sensor network is massive. In the literature, the following models have been used for characterizing sensor networks.

1.3.1 Erdös-Rényi Random Graph Model

A random graph is a graph that is generated by starting with a set of n vertices and then randomly adding edges between them. In the groundbreaking Erdös-Rényi model [57], a random graph is denoted by $G(n, p_b)$, in which every possible edge independently occurs with probability p_b.

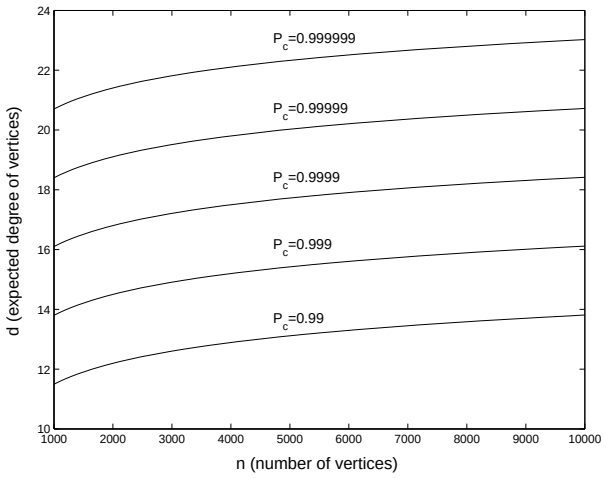


Figure 1.3: Expected degree of vertices vs. number of vertices, where P_c is desired connectivity of random graph

To achieve a desired probability P_c of graph connectivity, the threshold value of expected vertex degree d in a random graph $G(n, p_b)$ is determined by

$$d = \frac{n-1}{n}(\ln(n) - \ln(-\ln(P_c))) \ , \tag{1.1}$$

because $d = p_b(n-1)$ and

$$P_c = \lim_{n\to\infty} \mathsf{Pr}\left(G\left(n, p_b = \frac{\ln(n)}{n} + \frac{c}{n}\right) \text{ is connected}\right) = e^{-e^{-c}} \ ,$$

where c is a real constant.

Figure 1.3 depicts the plot of the expected vertex degree d as a function of the graph size n for various values of desired graph connectivity P_c. This figure shows that, to increase the probability that a random graph is connected by one order, the expected degree of vertices increases only by 2. Moreover, the curves in the figure are almost flat when n is large, indicating that the size of the graph has insignificant impact on the expected degree of vertices in an almost connected random graph.

After Eschenauer and Gligor introduced the Erdös-Rényi random graph theory into wireless sensor networks in their seminal paper [59], it is quite popular in sensor networks and serves as a principal modeling tool for a variety of WSNs security protocols [152]. In the basic mode of the Eschenauer-Gligor random probabilistic key predistribution scheme [59], an offline trusted key distribution server generates a key pool and each node is preloaded with a fix number of random keys out of the key pool before deployment such that the

probability of every pair of nodes sharing at least one key is not less than p_b and then the connectivity of the whole network is not below P_c, based on the Erdös-Rényi random graph theory. Chan, Perrig, and Song [41] proposed a q-composite mode of the Eschenauer-Gligor scheme which requires that two nodes have at least q common keys to set up a link and use all common keys instead of the first one to establish the pairwise key. Liu, Ning, and Li [111] introduced a key predistribution scheme which combines the Eschenauer-Gligor scheme with polynomial-based key predistribution protocol in [29]. Moreover, Du *et al.* [54] independently presented a technique which is equivalent to Liu-Ning-Li's scheme. In addition, Traynor *et al.* [142] proposed a random key distribution scheme based on the heterogeneous sensor network model. Even though Pietro *et al.* [125] questioned the realistic assumption of random graph model in WSNs and proposed another geometric random model for WSNs, Wu and Stinson [151] further investigated these models and validated the use of the random graph model in computing the connectivity of WSNs.

1.3.2 Unit-Disk Graph

A unit-disk graph [47] is a geometric graph constructed by a collection of vertices in the Euclidean plane and edges between a pair of vertices if and only if their distance is less than a unique threshold. Essentially, a vertex is represented by a disk of unit radius in the plane and is connected with all vertices within its corresponding disk. A unit-disk graph example with 100 vertices randomly deployed in a square is given in Figure 1.4.

Since the work of Huson and Sen [86], unit-disk graph has been used to model the topology of ad-hoc wireless networks with homogeneous nodes that have equal transmission ranges, and many wireless sensor networks clearly comply with the classification. Even though Breu and Kirkpatrick [33] showed that given a graph, determining whether it is a unit-disk graph is NP-hard, many important graph optimization problems can be effectively approximated [18]. Moreover, in many protocol proposals [128, 121], sensor nodes are assumed to randomly deployed in a geographic area, thus a random unit-disk graph in which nodes are uniformly deployed in a square and follow the standard unit-disk bidirectional communication model naturally become a useful network scenario for protocol simulations. This can be generalized as a random geometric graph, and its properties are investigated by Avin [15].

1.3.3 Percolation Theory

Percolation theory, introduced by Broadbent and Hammersley [36], describes the properties of connected clusters in a geographic graph. One of its applications in wireless sensor networks is to analyze the impact of deployment errors on sensor worm propagation [156].

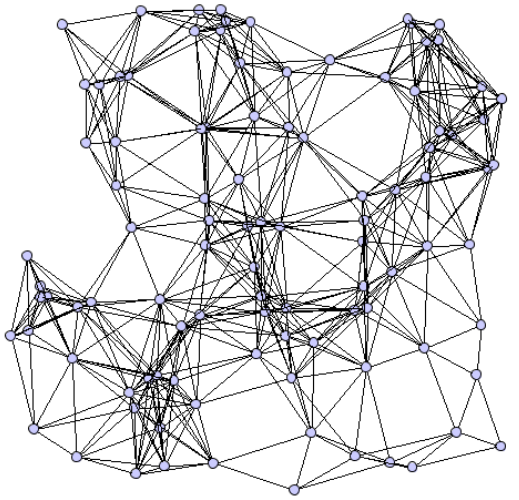


Figure 1.4: A unit-disk graph

As a representative question of this theory, for a three-dimensional network of $m \times m \times m$ vertices which every pair of neighbor vertices are connected independently with probability p_g, given p_g, what is the probability that an open path exists from the top to the bottom? The main concept of percolation theory is the existence of a percolation threshold p_t below which the probability is almost 0 and above which the probability is nearly 1.

In some cases p_t may be calculated explicitly. For example, for the square lattice in two dimensions, $p_t = 0.5$ [95]. This is the case of bond percolation. If we change to observe open/close vertices rather that edges, it then is called site percolation, and the site percolation threshold of the previous square lattice becomes $p_t = 0.593$. In addition, a limit case for lattices in many dimensions is given by the Bethe lattice, whose bond threshold is $p_t = \frac{1}{z-1}$ where z is the coordination number [31].

A more complex continuum percolation problem, originally introduced by Gilbert [73], can be used to find a critical density of a Poisson point process at which an unbounded connected component almost surely appears so that the network can provide long-distance multi-hop communication for wireless sensor networks [11].

1.3.4 Deployment Models

Generally speaking, in large-size, dense sensor networks, nodes are randomly deployed in targeted areas. However, in many scenarios, some deployment patters can be exploited to facilitate system design and improve protocol performance. For example, if sensor nodes are scattered by an airplane, these nodes might be grouped or placed in a particular

order before deployment; by careful research on those patterns, a deployment distribution may be approximately created such that a key-predistribution protocol can estimate nodes geographic or relative locations in advance and avoids unnecessary key assignments.

In [110], a sensor node is expected to be deployed, instead of at a specific location, within an area with certain probability, which is modeled by a probability density function. Du *et al.* [55] proposed a group-based deployment model, in which sensor nodes are divided into equal-size groups, each group is intended to be deployed at a different point in a grid network, and the derivation of a node actual position from its group target follows a non-uniform probability distribution such as Gaussian distribution.

1.4 Simulation Tools

Similar to other large-scale networks, simulation is an extremely useful and vital method in wireless sensor networks to inspire creative proposals, provide necessary abstraction to swiftly deploy prototyping system, comprehensively evaluate protocol performance, substantiate arguments, and verify theoretical analysis.

1.4.1 Categories

There are various means to carry out simulations for sensor networks. According to the implementation foundation, they can be categorized in three types.

- Experiments are directly realized by general-purpose programming languages: Python, Matlab, C++, Java, etc.. Those languages are broadly grasped by researchers, and they are quite handy and helpful to execute trials for abstract models.
- Simulations are built upon discrete event network simulators, mainly NS-2 [3, 87] and OMNeT++ [2, 146]. NS-2 supports an assortment of network protocols and provides interfaces through a script language to mange simulation. In contrast, OMNeT++ is an extensible, modular, component-based simulation framework, with emphasis on an efficient event dispatcher, ample data collection gadgets, and minimalist core simulation library, while specific system architectures and network protocols are offered by extended packages.
- Simulators use platforms that are specially designed for wireless sensor networks, such as TOSSIM [5], which is a built-in simulator in the popular sensor network operation system TinyOS [4]. In addition, Castalia [1], as a simulator for wireless sensor networks, is constructed based on the OMNeT++ framework to test distributed protocols in a realistic wireless channel and radio model. Those platforms

supply thorough sensor network protocol stacks and main-stream wireless medias, and proposals performance can be measured in realistic metrics.

1.4.2 Selections for WSNs Security Protocol Simulations

Although the sensor network specialized simulators appear attractive for research on securing sensor networks at first glance, they are rarely used in practice, because those lower-layer details, like wireless media and MAC specifications, are in principle irrelative to most security proposals for network and application layers. As far as those protocols designers are concerned, abstract metrics are generally sufficient or even preferable to physical readings for performance evaluation and fair comparisons in a large scale network application scenario. For instance, to measure a family of security protocols' energy consumption on communication, instead of gauging consumed power of node transmitters for a deliberately configured sensor networks, average number of messages sent per node in an abstract network scenario is actually adequate and appropriate. As a matter of fact, many security proposals present their simulation results by software programs without any simulator's support.

However, for our proposed security protocols of wireless sensor networks in this thesis, the simulations are primarily conducted on the OMNeT++ framework, except for a few Python-driven experiments for probability distribution models, due to the following considerations. First, a simulation library certainly facilitates efficient protocol implementation. Second, by means of its comprehensive data collection tools and inherent statistical supports [2], we can quickly absorb experimental results. In addition, this kind of open source simulator will decrease the chance of implementation biases and enlarge result credibility. For example, all our simulations results can be reproduced and verified because of the use of its deterministic pseudorandom number generators. Finally and most importantly, the OMNeT++ framework grants us the ability to implement general network scenarios including the models in last section, which can serve for a broad range of security protocols simulations. We delve into this below.

We extend the OMNeT++ framework to support main modeled scenarios of sensor networks: random graph, unit-disk graph with configurable network shape, two-dimensional square lattice, tree topology, cluster-based structure, with more scenarios under development such as different types of random trees. By standard software engineering design, those extensions are provided as a software package, and can be reused for general security protocols when the corresponding network scenarios are demanded.

1.5 Motivations and Related Work

1.5.1 Lightweight Entity Authentication

In the past few years, designing lightweight, unconventional, secure entity authentication schemes [91, 34, 76, 138] for low-cost pervasive devices, such as sensor nodes and radio frequency identification (RFID) tags, has been a hot topic in the cryptography and security communities due to the imperative practical demand and the formidable theoretical challenge.

Entity Authentication in Sensor Networks

The basic function of wireless sensor networks is to collect information for authorized users. Typically, base stations or users issue various commands of tasks to nodes; then nodes start to work accordingly, gathering data and forwarding to base stations or users. To operate properly, base stations and users should be authenticated to be the acclaimed entities by nodes. This is because, without entity authentication, adversaries can easily abuse the sensor networks to collect information maliciously or launch energy-exhaustion denial-of-service attacks by frequently ordering nodes to perform nonsense tasks. On the other hand, entities of nodes should also be authenticated by other nodes, base stations, and users. Otherwise, adversaries can insert invalid nodes into sensor networks to corrupt the result of information collection. Moreover, any further advanced access control mechanisms require entity authentication. In a word, mutual entity authentication plays a significant role in security of wireless sensor networks.

A number of entity authentication schemes in wireless sensor networks have been proposed. Benenson, Gedicke, and Raivio [23] introduced an entity authentication scheme of WSNs, based on elliptic curve cryptography. Jiang and Xu [89] presented a distributed entity authentication scheme in wireless sensor networks. It is built upon the self-certified keys cryptosystem, which is modified to use elliptic curve cryptography to establish pairwise keys for use in the entity authentication scheme. Wong *et al.* [149] proposed a dynamic strong-password-based entity authentication scheme for WSNs; then Tseng, Jan, and Wang [144] enhanced Wong *et al.*'s scheme to thwart potential replay and forgery attacks. Tripathy and Nandi [143] utilized cellular automata based components to provide entity authentication.

All of those approaches are based on conventional cryptographic mechanisms, symmetric or public-key. Since sensor networks consist of a large number of sensor nodes, the cost of a single node is critical to justify the overall cost of the network. In many applications of sensor networks, the production cost of nodes would dominate the success of systems. Akyildiz *et al.* [10] argued that the cost of a sensor node should be much less than one

dollar in order for sensor networks to be feasible. Under this constraint, sensor nodes may not be equipped with necessary hardware to perform costly standard cryptographic operations, even symmetric primitives. Therefore, innovative lightweight authentication schemes are strongly demanded for many sensor networks applications. Interestingly, another kind of pervasive devices: RFID tags suffer from similar or even more rigorous resource constraints, and a study on the identification protocols for RFID tags is surely conductive to fulfilling the authentication challenges in wireless sensor networks.

RFID Identification

Typically, RFID systems consist of simple, low-cost tags that are attached to physical objects and powerful readers that queue data from tags. As an revolutionary, efficient technique for automated identification of physical entities using radio frequency transmissions, RFID systems are employed in a wide variety of applications, such as supply chain management, payment, inventory monitoring, electronic password; and new applications are emerging every year. It is widely expected that RFID tags will inevitably replace barcodes correctly affixed to most of our daily consumer products and RFID systems will prevail in the physical identification mechanism market.

The low production expenditure of RFID tags is critical and essential to the appealing of RFID systems [91]. Roughly speaking, RFID tag's price must be below ten cents to be considered affordable for most RFID applications [123]. On the other hand, there are a number of security and privacy challenges which have to be addressed before the prevalence of RFID systems. Secure and efficient entity authentication is a crucial one, because it is a natural approach to prevent counterfeiting—the most severe attack to the identification devices.

The HB-like authentication protocols [91, 93, 76] have gained much attention in this field. The lightweight computation requirement of imposing only bit operations on authentication participants, the solid security foundation on a well-studied learning parity with noise (LPN) hard problem, and their elegant security reductionist proofs make them very attractive for entity authentication in the resource-constrained devices. Unfortunately, Ouafi, Overbeck, and Vaudenay [119] discovered an advanced man-in-the-middle attack, which is beyond the scope of the security modes used in [91, 93, 76], efficiently breaks down all HB-like protocols, and renders this kind of lightweight approaches like a dead end. The detailed evolution of HB-like protocols are their attacked will be given in next chapter.

Aside from the LPN-based approaches, SQUASH proposed by Shamir [138] might be tempting for RFID tags authentication, because of its simpleness and provable security equivalence to Rabin's public key encryption scheme. However, its security equivalence argument has been challenged by Ouafi and Vaudenay [120]. They successfully mounted an attack against a previous version of SQUASH: SQUASH-0, which uses a linear mixing

function while SQUASH employs non-linear mapping. Even thought it is not clear how or whether this attack can be adopted to SQUASH, they demonstrated that the security equivalence claim between SQUASH and Rabin cryptosystem is invalid. The security of SQUASH remains an open problem.

1.5.2 Countermeasures against Node Clone

Wireless sensor networks are subject to many physical attacks, and node clone is a severe one. Generally speaking, nodes are randomly deployed in surveillance areas, and working without attendance. Some nodes perish over time, due to failure or running out of power. To maintain or enhance the network functionalities, new nodes may be deployed into the current network. Because of production expense limitations, sensor nodes are usually short of tamper-resistance hardware components; thus an adversary can capture a few nodes, extract code and all secret credentials, and use those materials to clone many nodes out of off-the-shelf sensor hardware. Those cloned nodes that seem legitimate can freely join the sensor network and would significantly enlarger the adversary capacities to maliciously manipulate the network. For example, those vicious nodes occupy strategic positions and cooperatively corrupt the collected information. With a large number of cloned nodes in command, the adversary may even gain control of the whole network. And certainly, the node clone would exacerbate most of inside attacks against sensor networks, such as wormhole [84], spam [49], and denial of service [150].

In general, previous approaches against node clone can be categorized into three categories: prevention schemes that inherently forbid cloned nodes to join network, centralized detection in which there exists a central and powerful party (the base station at most of time) that is responsible for receiving reports and making judgements of node clone, and distributed detection where all nodes cooperatively process information and detect node clone in a distributed manner.

Prevention

Zhang *et al.* [157] proposed the use of location-based keys to thwart and defend against several attacks, one of which is node clone attack. The identity-based cryptography is used in their protocol such that nodes private keys are bounded by both their identities and locations. Once nodes are deployed, some trusted mobile agents travel around the sensor network, and issue the location-based keys to sensor nodes. Since those location-based keys cannot be used in nodes at other locations, node clone attack is inherently frustrated.

By similar arguments, we review key distribution protocols for sensor networks and it can be claimed that some of them prevent node clone as well. For example, in schemes

[159, 12] based on initial trust which assume that it takes adversaries a certain amount of time to compromise nodes after their deployment, valid keys only can be established during that safety period, and henceforth controlling nodes would not grant adversaries extra advantages, including the ability to clone nodes. Those prevention schemes might be useful on particular applications, but their assumptions as trusted mobile agents and initial trust are too strong to be applicable in general cases.

Centralized Detection

In a simplest centralized detection approach, each node sends a list of its neighbor nodes and their locations to a base station. If the base station finds that there are two far distant locations for one node ID, then the node clone must have occurred. SET, proposed by Choi, Zhu, and Porta [46], manages to reduce the communication cost of the approach above by computing set operations of exclusive subsets in the network. First, an exclusive subset maximal independent set (ESMIS) algorithm is performed by nodes to collaboratively form exclusive unit subsets among one-hop neighbors. As a result, each node is grouped into one and only one disjointed subset which is controlled by a randomly selected leader. Then those subsets, in the basic scheme, are transmitted by leaders to the base station such that it can construct all nodes locations and detect clones. Since the subset division procedure eliminates redundancy in the node location reports, SET lowers the communication cost. However, in order to prevent malicious nodes in the ESMIS algorithm, an authenticated subset covering protocol has to be performed, which considerably increases the communication burden and complicates the detection procedure.

Brooks *et al.* [37] proposed a clone detection protocol in the context of random key predistribution [59]. Its assumptions and application scenarios are quite different from other approaches. In fact, it is detecting compromised keys rather than cloned nodes. The basic idea is that the keys employed in random key predistribution scheme should follow a certain pattern, and those keys whose usage exceeds a threshold can be thought of as suspicious. In the protocol, every node reports its keys to a base station and then the base station performs an abnormality-based intrusion-detection-like statistical analysis to catch cloned keys. A common concern for this kind of approaches is their high false negative and positive rates. Furthermore, the authors do not address how to assure malicious nodes to honestly report their keys, which is critical to the protocol effectiveness. In addition, the use of bloom filter to transmit keys in the scheme is inappropriate and its corresponding analysis is flawed.

Ho, Wright, and, Das [81] discussed the clone detection in mobile sensor networks. All nodes still report their neighbors information to a base station. By the assumption of a maximum speed limit on sensor nodes, if some nodes exceed the upper limit, then they would be considered cloned.

Table 1.1: Comparison of previous distributed detection protocols, where n is network size, d is node degree

Protocols	Requirements for Nodes	Comm. Cost	Memory Cost
Node-To-Network Broadcasting [121]	Neighbors information	$O(n)$	$O(d)$
Randomized Multicast [121]	Awareness of all nodes	$O(n)$	$O(d\sqrt{n})$
Line-Selected Multicast [121]	Awareness of all nodes	$O(\sqrt{n})$	$O(d\sqrt{n})$
Randomized, Efficient, and Distributed [48]	Knowledge of whole network geography	$O(\sqrt{n})$	$O(d\sqrt{n})$
Single Deterministic Cell [158]	Knowledge of whole network geography	$O(\sqrt{n})$	$O(\sqrt{n})$
Parallel Multiple Probabilistic Cells [158]	Knowledge of whole network geography	$O(\sqrt{n})$	$O(\sqrt{n})$

As pointed out in [121], centralized approaches are prone to single-point of failure, and the nodes surrounding the base station suffer an undue communication burden that may shorten the network's life expectancy. In general, a distributed and balanced detection scheme is more desirable.

Distributed Detection

The straightforward node-to-network broadcasting [121] is a quite practical way to distributively detect the node clone, in which every node collects all of its neighbors identities along with their locations, and broadcasts to the network. When a node receives a broadcasted message from others, it compares those nodes listed in the message with its own neighbors and revokes neighbor nodes that have collided locations. The main problem in this approach is its extremely high communication overhead.

Parno, Perrig, and Gligor [121] provided two probabilistic detection protocols in a completely distributed, balanced way. Randomized multicast scheme distributes node location information to randomly-selected nodes as inspectors, exploiting the birthday paradox to detect cloned nodes, while line-selected multicast scheme uses the topology of the network to improve detection, that is, in addition to inspector nodes, the nodes within the multicast path check the node clone too. Unfortunately, to obtain acceptable detection probability, nodes have to buffer a great many of messages. Moreover, the communication cost in the randomized multicast is similar to that in the node-to-node broadcasting. For the procedure of choosing random inspectors, those schemes both imply that every node is aware of all other nodes' existence, which is a very strong assumption for large-scale sensor networks

and thus limits their applicability.

A Geographic Hash Table (GHT) [132] maps a key into a geographical coordination. Based on GHT, Zhu *et al.* [158] proposed a localized multicast to detect the node clone. In the system, the inspector nodes for an examined node are chosen from nodes that are located within a geographical limited region (named cell) which is determined by a GHT hash result of the node identification. They presented two variants of localized multicast: single deterministic cell, in which only one unique cell is determined for one node, and parallel multiple probabilistic cell, in which the location claim is mapped and forwarded to multiple deterministic cells with various probabilities. Conti *et al.* [48] proposed another GHT-based clone detection approach. Those approaches rely on the nodes knowledge of the general deployed geography of sensor networks. This prerequisite may hold in some circumstances, but cannot be guaranteed generally.

Table 1.1 compares those distributed detection protocols in terms of assumption, communication cost, and memory consumption. In summary, those distributed detection protocols either rely on an implicit assumption that every node is aware of all other nodes existence, or require that nodes realize the general network deployment graph. Those assumptions hardly hold for ordinary large scale, randomly deployed sensor networks. In addition, their performance is not quite satisfactory on certain metrics.

1.5.3 Secure Data Aggregation

In a typical wireless sensor network, hundreds and thousands of low-cost sensor nodes scatter within a surveillance area, receive commands from a base station, perform designated detection tasks accordingly, and collaboratively transmit results back to the base station. In many cases, instead of forwarding every individual message to the base station, sensor network protocols support data aggregation—the operation by intermediate nodes that combines many messages and sends out aggregated results. As a matter of fact, from the very beginning of wireless sensor networks development [60, 10, 130, 62, 19], it has already been widely accepted that data aggregation plays a critical role in the practicability and appealing of WSNs. Due to infeasibility of recharging nodes batteries in most circumstances, energy becomes the most valuable resource for sensor nodes. Among all nodes operations, data transmission consumes energy the most [10, 19]. Intuitively, data aggregation during message transmission is an effective method to preserve sensor nodes precious energy. Moreover, in the absence of data aggregation, sensor nodes near the base station will suffer from heavy message transmission overhead, and then die of power exhaustion much sooner than other nodes, breaking down the whole network's functionality. As a result, data aggregation attracts a great deal of attention and many a data aggregation scheme has been proposed in recent years. Systematic surveys on this topic can be retrieved in [130, 62].

When sensor nodes are deployed in a hostile environment, security measurements should be taken into consideration for network protocols. Attacks to wireless sensor networks not only come from outsider adversaries, but also can be conducted by compromised, previously legitimate nodes. Thus applicable secure protocols should prevent malicious inside nodes from damaging the whole network's functionality, or at least constrain their impacts to a reasonable level. Unfortunately, data aggregation, which requires intermediate nodes to process and change messages, and security objectives, one of which is preventing malicious manipulation, conflict with each other in this regard. As a result, designing secure and practical data aggregation schemes, which are critical to many sensor network applications, imposes an interesting and formidable challenge.

Previous Approaches

Homomorphic primitives, besides standard cryptographic functionalities, allow users without secret key to legitimately perform acceptable algebraic operations on protected data block. Since aggregation is essentially some operation, it is intuitive to use homomorphic primitives for securing data aggregation, and a number of approaches [148, 39, 139] use homomorphic encryption for this application. Generally, it is very difficult to design secure symmetric homomorphic block encryption, whereas stream ciphers naturally support homomorphic exclusive-OR operations, which is exactly utilized in [39]. In contrast, public-key homomorphic encryption is an interesting topic, and there exist several relatively practical public-key homomorphic cryptosystems, such as Unpadded RSA, ElGamal, Goldwasser-Micali, Benaloh and Paillier [63], though all of them only support limited operations on ciphertexts. In 2009, Gentry [71], for the first time, presented a fully homomorphic encryption scheme, which outstandingly allows arbitrary operations on ciphertext. Even though the only two fully homomorphic encryption schemes [71, 145] by now cannot provide competitive performance for most applications, practical fully homomorphic encryption is expected to appear eventually. Unfortunately, sole public-key homomorphic encryption does not suffice for secure data aggregation in sensor networks, because then anyone can maliciously insert or manipulate results.

Hu and Evans [83] described a secure hop-by-hop data aggregation scheme, in which every node shares with the base station a different key, from which temporary session MAC keys will be derived, and by adopting hash-chain-based delayed message authentication, such as μTESLA [124], intermediate modes, after the base station reveals session MAC keys, will be able to verify the integrity of messages that they buffered. This scheme suffers from communication penalties, as the introduction of μTESLA for distributing session MAC keys incurs considerable communication cost. More disturbingly, in order to detect one inside malicious node that manipulates other nodes input, intermediate nodes have to obtain and buffer all their grandchildren's messages and corresponding MACs, that

is, two-hop messages buffer only being able to detect *one* misbehavior node. Although Jadia and Muthuria [88] extended the Hu-Evans scheme by all two nodes in the two-hop communication range sharing pairwise keys and then the scheme eliminates the usage of μTESLA, the fact that both schemes are only capable of preventing a single inside malicious node at an appreciable communication cost makes them impractical.

Yang *et al.* [155] presented a secure hop-by-hop data aggregation protocol for sensor networks named SDAP, using the principles of divide-and-conquer and commit-and-attest, which is a typical example of retroactive detection approach. In SDAP, a probabilistic grouping technique is utilized to dynamically partition the nodes in a tree topology into subtrees. A commitment-based hop-by-hop aggregation is conducted in each subtree to generate a group aggregate, and accordingly the base station identifies the suspicious subtrees based on the set of group aggregates. Finally, each subtree under suspect participates in an attestation procedure to prove the correctness of its group aggregate. Those complicated algorithms cause significant transmission overhead, and may cancel off all communication benefits from data aggregation.

Przydatek, Song, and Perrig [129] proposed secure information aggregation (SIA) to identify forged aggregation values from malicious nodes. In the SIA scheme, a special node named aggregator computes an aggregation result over raw data together with a commitment to the data based on a Merkle-hash tree and sends them back to a remote user, which later challenges the aggregator to verify the aggregation. Later Chan, Perrig, and Song [42] built a hierarchical data aggregation on the aggregate-commit-prove framework in [129], but extended their single aggregator model to a fully distributed setting. Frikken and Dougherty [66] further improved the Chan-Perrig-Song scheme. Moreover, Chan and Perrig [40] derived several security primitives from this kind of algorithms.

In summary, when purely cryptographic mechanisms are used for securing data aggregation, homomorphic primitives might be the only suitable candidates, but more research is needed in this field. For other kinds of secure data aggregation approaches, sophisticated protocols are designed to detect malicious behaviors/nodes. Some of those mechanisms rely on unrealistic assumptions, while other involve heavy communication overhead, which conflicts with the very intention of data aggregation and makes it hard to evaluate their applicability. There are some well-designed secure data aggregation schemes, which understandably have different tradeoffs and are suitable for particular aggregation operations.

1.6 Overview of the Thesis

In the rest of the thesis, Chapter 2 and Chapter 3 deal with lightweight entity authentication. Then Chapter 4 presents two distributed node clone detection protocols. Afterwards,

Chapter 5 and Chapter 6 demonstrate our work on secure data aggregation. Finally Chapter 7 summarizes the conclusions of our work and suggests possible directions for future research. The chapter outlines and our contributions are described as follows:

- **Chapter 2** presents an innovative, lightweight, efficient one-way authentication protocol named LCMQ and prove it secure in a general man-in-the-middle model. The technical core in our proposal is a special type of circulant matrix name circulant-P2 matrix, for which we prove the linear independence of matrix vectors, present efficient algorithms on matrix operations, and describe a secure encryption against ciphertext-only attack. By combining all of those with learning parity with noise and multivariate quadratic problems, the LCMQ protocol not only is provably secure against all probabilistic polynomial-time adversaries, but also outperforms all HB-like protocols, in terms of tag's computation overhead, storage expense, and communication cost.

- **Chapter 3** addresses the mutual authentication challenge for extremely computation-constrained sensor nodes that only can perform bit operations. We first provide a one-way authentication protocol that is still based on the LPN problem and circulant-P2 matrix, but uses the HB-like structure to maintain the bit-operation-only requirement for both parties. This $\mathrm{HB}^{\mathcal{C}}$ protocol's key storage is approximately half of that in the pervious best HB-like protocol. To address inefficiencies in the HB-like protocols, we propose two enhancements that prevent the OOV attack and improve protocol performance. Lastly, we extend the enhanced Hb-like protocols into mutual authentication.

- **Chapter 4** proposes two innovative node clone detection protocols with difference tradeoffs on network conditions and performance. The first one, based on distributed hash table (DHT), presents good performance on efficient storage consumption and high security level. The performance evaluation is theoretically deducted through a probability model, and the resulting equations, with necessary adjustments for real application, are supported by the simulations. The other distributed detection protocol, named randomly directed exploration, incurs splendid communication performance for dense sensor networks, by an elegant probabilistic directed forwarding technique along with random initial direction and border determination. The experimental results uphold the protocol design and show its efficiency on communication overhead and satisfactory detection probability.

- **Chapter 5** shows three secure aggregation schemes that provide provably secure message integrity with different tradeoffs between computation cost, communication payload, and security assumptions. The first one is a homomorphic MAC, which

is a purely symmetric approach, and is the most computation- and communication-efficient, but requires all data-collecting nodes to share one global key with the base station. The other two make use of (public-key based) homomorphic hashing, combined with aggregate MAC and identity-based aggregate signature (IBAS) respectively. The scheme with aggregate MAC allows the base station to share a distinct key with every node, while the scheme with a paring-based IBAS enables all intermediate nodes beside the base station to verify the authenticity of aggregated messages.

- **Chapter 6** exhibits a succinct and practical secure aggregation protocol by combining HMAC (associated with a cryptographic hash function) with Bloom filter, which then is defined as secure Bloom filter. Unlike most previous approaches, which are aimed to provide security mechanisms for ordinary aggregation operations, our proposal firstly is an effective aggregation protocol, suitable for a specific but popular class of aggregation in wireless sensor networks. Benefiting from secure Bloom filter, the protocol, without any unrealistic assumptions, fulfills the fundamental security objective of preventing outside adversaries and compromised inside nodes from harming the overall network result. We systematically analyze the protocol performance and run extensive simulations on different network scenarios for evaluation. The simulation results demonstrate that the proposed protocol presents remarkable performance on security, communication cost, and degree of node energy consumption balance.

Chapter 2

Secure and Efficient LCMQ Entity Authentication Protocol

In this chapter, we present an innovative, efficient one-way entity authentication protocol named LCMQ (standing for the combination of learning parity with noise, circulant matrix, and multivariate quadratic), which is especially suitable for RFID systems and can be used in wireless sensor networks. To compare with similar protocols fairly, we use RFID systems as the protocol demonstration scenario. By a general man-in-the-middle model, we prove that the protocol is secure against all probabilistic polynomial-time adversaries. The protocol's security is still based on the hardness of the LPN problem, but the architecture cannot be categorized in the HB-like schemes. Instead, the protocol greatly benefits from the gentle properties and efficient algorithms of a special type of circulant matrix, to which the whole Section 2.2 is devoted. Furthermore, surprisingly, the protocol performance, in terms of computation, storage, and communication costs, outweighs all HB-like protocols, from the viewpoint of RFID tags, while it merely requires readers to additionally perform one extended Euclidean algorithm per authentication, which is trivial for those supposedly powerful devices.

The chapter is structured as follows. We begin with the definition of the learning with parity problem and the overview of an instructive journey of HB-like protocols in Section 2.1. Then Section 2.2 is focused on the technical core of the proposal: a special type of circulant matrix, for which we prove the linear independence of matrix vectors, present efficient algorithms on matrix operations, and describe a secure encryption. After that, the LCMQ protocol is specified and we prove it secure in a general man-in-the-middle model in Section 2.3. We discuss the protocol's performance and recommend practical parameters in Section 2.4. Finally, Section 2.5 concludes the work. The research results in this chapter have been presented in [107].

All vectors and matrices discussed in this chapter are binary. Subsequently, the oper-

ations on the vectors and matrices are over the finite field $GF(2)$. The following symbols will be used throughout this and next chapters:

$\boldsymbol{a} \oplus \boldsymbol{k}$	Bitwise exclusive-or (XOR) operation on two vectors (or matrices) $\boldsymbol{a}$, $\boldsymbol{k}$
$\boldsymbol{a} \cdot \boldsymbol{k}$	Inner-product of two vectors $\boldsymbol{a}$ and $\boldsymbol{k}$
$\mathbf{A} \circ \mathbf{K}$	Multiplication of two matrices $\mathbf{A}$ and $\mathbf{K}$
$\boldsymbol{a} \mid\mid \boldsymbol{b}$	Concatenation of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$
$\mathsf{Hwt}(\boldsymbol{k})$	Hamming weight of vector $\boldsymbol{k}$, that is, the number of ones in the bit vector
$\boldsymbol{\theta} \ggg i$	Right cyclic shift operation on vector $\boldsymbol{\theta}$ by i position
$\mathbf{0}_m$	m-bit vector in which all bits are zeros
$\mathbf{1}_m$	m-bit vector in which all bits are ones
$\boldsymbol{e}_i$	m-bit vector in which only bit at position i is one
$\bar{\boldsymbol{\theta}}$	Compliment of vector $\boldsymbol{\theta}$, i.e., $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta} \oplus \mathbf{1}_m$
$\mathbb{S}_m$	Set of all m-bit vectors except $\mathbf{0}_m$ and $\mathbf{1}_m$
$\mathbb{S}_m^{\mathrm{e}}$	Set of all vectors in $\mathbb{S}_m$ whose Hamming weights are even
$\mathbb{S}_m^{\mathrm{o}}$	Set of all vectors in $\mathbb{S}_m$ whose Hamming weights are odd

2.1 LPN Problem and HB-Family Protocols

2.1.1 Learning Parity with Noise Problem

Suppose the tag pre-shares a secret κ-bit vector $\boldsymbol{k}$ with the reader for subsequent authentications. First the reader randomly generates a sequence of binary vectors $\boldsymbol{a}_0, \boldsymbol{a}_1, \cdots, \boldsymbol{a}_{q-1}$ and transmits those challenges to the tag, which responds with $y_i = \boldsymbol{a}_i \cdot \boldsymbol{k}$, for $i \in [0, q-1]$ accordingly. The reader accepts the tag's authentication if and only if $\boldsymbol{a}_i \cdot \boldsymbol{k} = y_i$. Unfortunately, after observing κ linearly-independent challenge-response pairs of $\langle \boldsymbol{a}_i, y_i \rangle$, an adversary can readily recover the authentication key $\boldsymbol{k}$ by the Gaussian elimination.

In the presence of noise, however, where each response bit y_i is independently flipped by a noise bit one with probability $\eta \in (0, \frac{1}{2})$, determining $\boldsymbol{k}$ becomes much more difficult. This problem is known as learning parity with noise, or the *LPN Problem* for short. Formally, it is defined as follows.

Definition 2.1 (LPN Problem)**.** *View κ as a security parameter. Let $\boldsymbol{k}$ be a κ-bit secret vector, $\eta \in (0, \frac{1}{2})$ be a noise level. For $i \in [0, q-1]$ (q is a polynomial in κ), let $\boldsymbol{a}_i$ be a κ-bit random vector, and v_i be a noise bit that follows the Bernoulli probability distribution of parameter η. Given η and q pairs $\langle \boldsymbol{a}_i, y_i = (\boldsymbol{a}_i \cdot \boldsymbol{k}) \oplus v_i \rangle$, recover $\boldsymbol{k}$.*

The LPN problem has long been studied as the following equivalent problems: syndrome decoding problem [113, 25] and minimal disagreement parity problem [51]. It has been proven that the LPN problem is NP-hard [25]. Moreover, finding a vector satisfying

more than half of the challenge-response pairs, even though it looks like an easier problem, remains NP-hard [79]. Furthermore, Regev [134] introduced a natural extension of the LPN problem, referring to as the learning with error (LWE) problem, by generalizing binary field $GF(2)$ in the LPN problem into prime field $GF(p)$, where p is a prime number. Impressively, Regev [134] proved the reduction from worse-case lattice problems, such as shortest-vector problem (SVP), to the LWE problem. However, the reduction proof employs a quantum algorithm, which is, generally speaking, weaker than a classical reduction mechanism, as there is still no practical quantum computer available by now.

In reality, the security of LPN-based authentication protocols, similar to other NP-hard problems for application in the cryptography, still depends on the hardness of the average case of the LPN problem, while the NP-hard allegation only guarantees the intractability in the worse case. Intuitively, the combination of the key length κ and the noise level η determines the security level of LPN instances. Blum, Kalai, and Wasserman [28] provided the first sub-exponential algorithm (BKW algorithm) for the LPN problem, which requires $2^{O(\kappa/\log\kappa)}$ equations/operations. Fossorier *et al.* [64] improved the BKW algorithm. At present, the fastest algorithm is the LF algorithm, another enhancement of BKM algorithm, presented by Levieil and Fouque [100]. According to the LF algorithm, a common parameter set for 80-bit security level is $(\eta = 0.25, \kappa = 512)$. Should LPN-based protocols be widely employed, it is highly likely that algorithms of the LPN problem can be improved notably, then bigger key lengths are demanded, as we have witnessed the significant increase of RSA (and discrete logarithm) public key length in the three decades. Since typical LPN-based protocols involve $(\kappa \times \kappa \times O(\kappa))$ matrix multiplication, such big values of κ would incur considerable computation and implementation costs so as to push protocols away from *lightweight*. Fortunately, in our proposed protocol, LPN instances are encrypted by a succinct secure scheme so that the protocol does not suffer from the restriction and a practical value of key length can be as low as 163.

2.1.2 The Journey of HB-Family Authentication Protocols

HB and HB$^+$ Protocols

In the HB protocol [82] , the tag and the reader have a secret vector $\boldsymbol{k}$ in common. They interact n rounds of two passes for authentication. In each round, the reader generates and sends a random binary vector $\boldsymbol{a}$ as challenge; and then the tag responds with the inner-product of the challenge vector and the secret $\boldsymbol{k}$, but with noise of probability η on purpose. After n rounds, the tag is authenticated provided the number of rejected challenge-response pairs is not greater than $n\eta$.

Assuming the intractability of LPN problem, the HB protocol is provably secure against passive eavesdroppers [91, 93]. However, an active adversary can easily overcome the noise

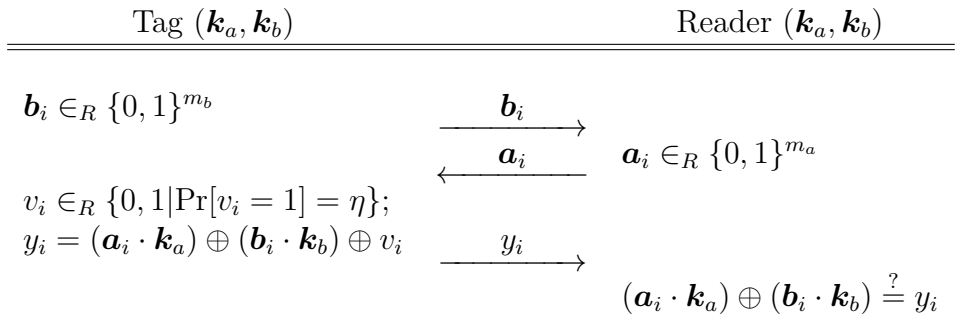


Figure 2.1: The ith round of the HB$^+$ authentication protocol, where m_a-bit $\boldsymbol{k}_a$ and m_b-bit $\boldsymbol{k}_b$ are two vectors as authentication key, $\eta \in (0, \frac{1}{2})$, $\boldsymbol{b}_i$ is a *blinding vector*, $\boldsymbol{a}_i$ is a *challenge vector*

and then recover the secret $\boldsymbol{k}$: If the same challenge $\boldsymbol{a}$ is repeated many times by the adversary, he can learn the error-free value of $\boldsymbol{a} \cdot \boldsymbol{k}$ with overwhelming probability (*JW attack*) [91]. To defend against the JW active attack, Juels and Weis [91] proposed the HB$^+$ three-pass authentication protocol. HB$^+$ still involves n rounds. One single round of HB$^+$ is outlined in Figure 2.1. Similar to the HB protocol, after n rounds, the authentication succeeds if no more than $n\eta$ responses do not match challenges.

Juels and Weis [91] presented an elegant reduction security proof of the HB$^+$ protocol in a limited active model: detection-based-model, which is primarily addressing active attacks similar to the JW attack. Originally, the security proof of HB$^+$ in [91] demands the sequential execution of n rounds three-pass interactions. To overcome this limitation, Katz and Shin [93] brought an security proof of the HB$^+$ protocol in the case of parallel and concurrent executions. Moreover, Katz and Smith [94] extended the reduction results to a larger range of noise levels $\frac{1}{4} \leq \eta < \frac{1}{2}$ whereas the Katz-Shin proof [93] holds only on the condition of $\eta < \frac{1}{4}$. In spite of those gentle security proofs, Gilbert, Robshaw, and Sibert [74] discovered a simple, effective man-in-the-middle (MIM) attack (referred to as the *GRS attack*), which is outside of the detection-based-model, and fully compromises the HB$^+$ protocol.

GRS Attack

In the second pass of every round of one HB$^+$ authentication procedure, an MIM adversary intercepts challenge $\boldsymbol{a}_i$ from the reader, and transmits to the tag a modified challenge $\boldsymbol{a}_i \oplus \boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is a constant vector for one authentication procedure. By observing this manipulated authentication procedure outcome—acceptance or rejection, the adversary learns the result of $\boldsymbol{\alpha} \cdot \boldsymbol{k}_a$, that is, one bit information of $\boldsymbol{k}_a$. The adversary simply repeats m times of manipulating authentication procedures with linearly independent $\boldsymbol{\alpha}$'s, and

completely recovers $\boldsymbol{k}_a$. Now the adversary is able to impersonate a valid tag by choosing $\mathbf{0}_m$ as the blinding vector; or the adversary can further determine $\boldsymbol{k}_b$ by acting as a tag to interact with a genuine reader, using a constant blinding vector $\boldsymbol{b}'$ in one authentication procedure, responding challenge $\boldsymbol{a}_i$ with $\boldsymbol{a}_i \oplus \boldsymbol{k}_a$, and learning the result of $\boldsymbol{b}' \oplus \boldsymbol{k}_b$ according to acceptance or rejection. In addition, although the original GRS attack is restricted to the interference of challenges from the reader to the tag, the same GRS manipulation strategy can be applied to blinding vectors to recover $\boldsymbol{k}_b$; after that, the adversary can launch the original JW attack to retrieve $\boldsymbol{k}_a$, totally breaking the protocol.

Even after a series of HB$^+$ enhancement protocols, such as HB^{++} [35], HB* [56], HB-MP [117], modification of HB^{++} [126] and HB-MP$^+$ [99] had been proposed, Gilbert, Robshaw, and Seurin [75] demonstrated that those variants still could be attacked in the linear time while increasing the computational complexity and/or reducing the practicality. The PUF-HB protocol [78] and the Trusted-HB protocol [34] make use of a physically unclonable circuit and a lightweight hash function family respectively, intending to thwart the GRS attack. However, the introduction of such ingredients into HB$^+$ might not fully meet the motivation of designing lightweight simple-bit-operation-based authentication protocols. Moreover, Frumkin and Shamir [67] have broken the security of Trusted-HB in realistic scenarios.

Random-HB$^\#$ and HB$^\#$ Protocols

Gilbert, Robshaw, and Seurin [76] presented these two protocols, which are resistant to the GRS attack. In contrast to secret vectors in HB$^+$, Random-HB$^\#$ employs two secret matrices $\mathbf{K}_a$ and $\mathbf{K}_b$. One Random-HB$^\#$ authentication consists of a blinding vector $\boldsymbol{b}$ from the tag, a challenge vector $\boldsymbol{a}$ from the reader, and then the tag's response vector $\boldsymbol{y} = (\boldsymbol{a} \circ \mathbf{K}_a) \oplus (\boldsymbol{b} \circ \mathbf{K}_b) \oplus \boldsymbol{v}$, where $\boldsymbol{v}$ is an n-bit noise vector each bit of which independently follows the Bernoulli distribution of parameter η. Similarly, the reader validates the tag's authentication iff $\mathsf{Hwt}((\boldsymbol{a} \circ \mathbf{K}_a) \oplus (\boldsymbol{b} \circ \mathbf{K}_b) \oplus \boldsymbol{y})$ does not exceed threshold τ. The binding/challenge vectors rather than matrices in HB$^+$ exceedingly reduces the communication cost, but the secret matrices in Random-HB$^\#$ imposes too high storage burden to be practical in realistic systems. In order to overcome the drawback, they proposed to replace random matrices with Toeplitz matrices, which becomes the HB$^\#$ protocol.

Gilbert, Robshaw, and Seurin [76] defined a GRS-MIM-model, in which the MIM adversary is only allowed to manipulate the challenges from the reader to the tag, to prove that Random-HB$^\#$ and HB$^\#$ [76] are resistant to the GRS attack. In addition, Random-HB$^\#$ is provably secure in the detection-based-model, while HB$^\#$ is conjectured to be secure [76]. The security reductionist proofs in [76] are rather impressive. However, as the GRS-MIM-model does not simulate a full man-in-the-middle adversary, a general MIM

attack was discovered soon, breaking down both Random-HB$^{\#}$ and HB$^{\#}$, and making the perspectives of secure LPN-based authentication protocols gloomy.

OOV Attack

At AsiaCrypt 2008, Ouafi, Overbeck, and Vaudenay [119] presented a general man-in-the-middle attack (referred to as *OOV attack*) against all current HB-like protocols. The basic OOV attack against Random-HB$^{\#}$/HB$^{\#}$ is conducted as follows. The attacker first eavesdrops on one successful execution of the protocol, obtaining a triplet $(\widehat{\boldsymbol{b}}, \widehat{\boldsymbol{a}}, \widehat{\boldsymbol{y}})$ satisfying $\widehat{\boldsymbol{y}} = (\widehat{\boldsymbol{a}} \circ \mathbf{K}_a) \oplus (\widehat{\boldsymbol{b}} \circ \mathbf{K}_b) \oplus \widehat{\boldsymbol{v}}$ and $\mathsf{Hwt}(\widehat{\boldsymbol{v}}) \leq \tau$. Then the MIM adversary manipulates many executions of the protocol by XORing interactions $(\boldsymbol{b}_i, \boldsymbol{a}_i, \boldsymbol{y}_i)$ with $(\widehat{\boldsymbol{b}}, \widehat{\boldsymbol{a}}, \widehat{\boldsymbol{y}})$; thus each authentication result is actually decided by whether $\mathsf{Hwt}(\boldsymbol{v}_i \oplus \widehat{\boldsymbol{v}}) \leq \tau$. Based on the overall success probability, the attacker can calculate the value of $\mathsf{Hwt}(\widehat{\boldsymbol{v}})$ with a high probability. After that, the adversary changes $\widehat{\boldsymbol{y}}$ by one bit to $\widehat{\boldsymbol{y}}'$, uses $(\widehat{\boldsymbol{b}}, \widehat{\boldsymbol{a}}, \widehat{\boldsymbol{y}}')$ to interfere with many executions of the protocol, and get the result of $\mathsf{Hwt}(\widehat{\boldsymbol{v}}')$, where $\widehat{\boldsymbol{v}}' = (\widehat{\boldsymbol{a}} \circ \mathbf{K}_a) \oplus (\widehat{\boldsymbol{b}} \circ \mathbf{K}_b) \oplus \widehat{\boldsymbol{y}}'$. By comparing the values of $\mathsf{Hwt}(\widehat{\boldsymbol{v}})$ and $\mathsf{Hwt}(\widehat{\boldsymbol{v}}')$, one bit in noise vector $\widehat{\boldsymbol{v}}$ is determined. Repeating this process, the adversary eventually obtains the noise-free result of $(\widehat{\boldsymbol{a}} \circ \mathbf{K}_a) \oplus (\widehat{\boldsymbol{b}} \circ \mathbf{K}_b)$. The adversary collects enough equations that he can completely recover $\mathbf{K}_a$ and $\mathbf{K}_b$, breaking the protocol.

Ouafi, Overbeck, and Vaudenay [119] also examined the lower bounds on the parameter sets for which the OOV attack is not effective. As concluded in [119], such parameters are unpractical to use in the low-cost devices. One may argue that since the OOV attack would cause many rejections, it can be relieved by setting up an upper bound of rejection number such that an authentication key shall be revoked once the number of failed authentication using the key exceeds the bound. In fact, this cumbersome approach counts on the outside mechanism, and is not satisfactory.

Noise Modes and Error Rates

For the HB-like protocols with the Bernoulli noise mode, there exist two types of authentication errors. A *false negative*, that is, the authentication of a legitimate tag being rejected, takes place when the number of incorrect responses exceeds the pass-threshold τ. By contrast, a *false positive* is defined that the number of unmatched responses out of random bits is less than the pass-threshold τ. In other words, we assume that an illegitimate tag only responses with random bits. The false negative rate P_{FN} and the false positive rate P_{FP} are determined [100, 76] by

$$P_{\mathrm{FN}} = \sum_{i=\tau+1}^{n} \binom{n}{i} \eta^i (1-\eta)^{n-i} \text{ and } P_{\mathrm{FP}} = \sum_{i=0}^{\tau} \binom{n}{i} 2^{-n} \ . \tag{2.1}$$

Since the Bernoulli noise mode would cause a certain false negative rate in HB-like protocols, a natural method to overcome that drawback is to demand the tag to generate a noise vector $\boldsymbol{v}_i$ of bounded Hamming weight, that is $\mathsf{Hwt}(\boldsymbol{v}_i) \leq \tau$, as discussed in [93, 76]. We refer to it as the upper-bounded Binomial noise mode. Ouafi, Overbeck, and Vaudenay [119] demonstrated another simple man-in-the-middle attack (referred to as *OOV2 attack*) against HB-like protocols with this noise mode. For one iteration $(\boldsymbol{b}_i, \boldsymbol{a}_i, \boldsymbol{y}_i)$ of Random-HB$^{\#}$ with this noise mode, an OOV2 attacker manipulates the response $\boldsymbol{y}_i$ such that the reader receives $\boldsymbol{y}_i \oplus \boldsymbol{r}_i$ rather than $\boldsymbol{y}_i$, where $\boldsymbol{r}_i$ is a random vector of Hamming weight 2. Let $w_i = \mathsf{Hwt}((\boldsymbol{a}_i \circ \mathbf{K}_a) \oplus (\boldsymbol{b}_i \circ \mathbf{K}_b) \oplus \boldsymbol{y}_i)$ be the Hamming weight of the noise added by the tag. If and only if $w = \tau - 1$ or τ and the attacker flipped two non-erroneous bits, which come from the only two non-zero elements in $\boldsymbol{r}_i$, in the response, the reader rejects the authentication. In other words, from one occurrence of rejection, the attackers learn two bits of $\mathbf{K}_a/\mathbf{K}_b$. Subsequently, all bits of secret matrices can be retrieved by conducting the process many times.

2.2 Linear Independence, Efficient Computation, and Encryption Scheme on a Special Type of Circulant Matrix

In this section, we state the main technique underlining the proposed protocol: a special type of circulant matrix named *circulant-P2 matrix*. First, let us review the circulant matrix.

Traditionally, a circulant matrix is a square matrix in which each row vector is rotated one element to the right relative to the preceding row vector. That is, an $(m \times m)$ square circulant matrix with first row vector $\boldsymbol{\theta} = (\theta_0, \theta_1, \cdots, \theta_{m\text{-}1})$ is

$$\begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_{m-1} \\ \theta_{m-1} & \theta_0 & \cdots & \theta_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_1 & \theta_2 & \cdots & \theta_0 \end{bmatrix}.$$

Let n be an integer in $[1, m-1]$, we extend a circulant matrix into non-square cases: defining a landscape circulant matrix as an $(n \times m)$ matrix in which each row vector is a right cyclic shift by one of the row vector above, and a portrait circulant matrix as an $(m \times n)$ matrix in which each column vector is a right cyclic shift by one of the column vector before it, while referring to the original one as a square circulant matrix.

Table 2.1: List of P2 numbers below 2048

3, 5, 11, 13, 19, 29, 37, 53, 59, 61, 67, 83, 101, 107, 131, 139, 149, 163, 173, 179, 181, 197, 211, 227, 269, 293, 317, 347, 349, 373, 379, 389, 419, 421, 443, 461, 467, 491, 509, 523, 541, 547, 557, 563, 587, 613, 619, 653, 659, 661, 677, 701, 709, 757, 773, 787, 797, 821, 827, 829, 853, 859, 877, 883, 907, 941, 947, 1019, 1061, 1091, 1109, 1117, 1123, 1171, 1187, 1213, 1229, 1237, 1259, 1277, 1283, 1291, 1301, 1307, 1373, 1381, 1427, 1451, 1453, 1483, 1493, 1499, 1523, 1531, 1549, 1571, 1619, 1621, 1637, 1667, 1669, 1693, 1733, 1741, 1747, 1787, 1861, 1867, 1877, 1901, 1907, 1931, 1949, 1973, 1979, 1987, 1997, 2027, 2029

2.2.1 Definition of Circulant-P2 Matrix

Definition 2.2 (Circulant-P2 matrix and P2 number)**.** *A circulant-P2 matrix is an $(m \times m)$ square circulant matrix, or an $(n \times m)$ landscape circulant matrix, or an $(m \times n)$ portrait circulant matrix, satisfying the following conditions.*

1. *It is a binary matrix.*
2. *m is a prime number satisfying that 2 is a primitive element of finite field $GF(m)$. In this case, m is defined as a P2 number.*
3. *Neither $\mathbf{0}_m$ nor $\mathbf{1}_m$ is a row vector (or a column vector) of a circulant-P2 matrix.*

Note that the definition above implies $n < m$. The second condition is central for circulant-P2 matrices. If and only if $2^i \mod m \neq 1, \forall 1 \leq i \leq m-2$, then 2 is a primitive element of finite field $GF(m)$. We list all P2 numbers less than 2048 in Table 2.1.

Definition of Characteristic Vector

A *characteristic vector* of a square circulant-P2 matrix is defined as its first row vector. As for a landscape or portrait circulant-P2 matrix, since it is actually a truncated portion of a square circulant-P2 matrix, its characteristic vector is defined as the corresponding square circulant-P2 matrix's characteristic vector. For a circulant-P2 matrix with m-bit characteristic vector $\boldsymbol{\theta}$, we denote square, landscape, and portrait cases by $\mathbf{C}_{\boldsymbol{\theta}}$, $\mathbf{C}_{\boldsymbol{\theta}}^{[n\times m]}$, and $\mathbf{C}_{\boldsymbol{\theta}}^{[m\times n]}$ respectively.

2.2.2 Linear Independence

Linear independence for vectors in a circulant-P2 matrix is the most important property for our proposed protocol and we discuss it as follows.

To facilitate discussions, we define an auxiliary term of a circulant-P2 matrix.

Definition 2.3 (Equivalence Class)**.** *For two vectors in $\mathbb{S}_m$ (recall that $\mathbb{S}_m$ is the set of all m-bit vectors except $\mathbf{0}_m$ and $\mathbf{1}_m$), say $\boldsymbol{a}$ and $\boldsymbol{b}$, if $\exists i \in \{0, \cdots, m-1\}$ such that $\boldsymbol{b} = \boldsymbol{a} \ggg i$, then we define that $\boldsymbol{a}$ and $\boldsymbol{b}$ are cyclically shift equivalent and they are in an equivalence class.*

An equivalence class can be represented by any one of its members.

Lemma 2.1. *If m is a prime number, then there are $\frac{2^m-2}{m}$ disjoint equivalence classes in $\mathbb{S}_m$. Each equivalence class contains m elements.*

Proof. An equivalence class in $\mathbb{S}_m$ has at most m elements; and any two different equivalence classes are disjoint—they do not share any common elements. Since $\mathbf{0}_m$ and $\mathbf{1}_m$ are not elements in $\mathbb{S}_m$, every equivalence class contains at least two elements. Suppose there is an equivalence class $\not\leq$ that has less than m elements. It means that there exists at least one element $\boldsymbol{\theta'}$ satisfying $\boldsymbol{\theta'} \ggg i = \boldsymbol{\theta'}$ where $1 < i < m$ (i cannot be 1; otherwise the equivalence class only has one element). Due to the characteristic of equivalence class, the relation $\boldsymbol{\theta} \ggg i = \boldsymbol{\theta}$ holds for every element $\boldsymbol{\theta}$ in $\not\leq$. Consequently, i should be a factor of m. However, it contradicts the fact that m is prime, since m only has two factors 1 and m while $1 < i < m$. Therefore, every equivalence class of $\mathbb{S}_m$ exactly has m elements, and there are $\frac{2^m-2}{m}$ disjoint equivalence classes in $\mathbb{S}_m$. □

A proof of Lemma 2.1 also can be found in [77]. For completeness, we present this proof.

Lemma 2.2. *If m is prime and 2 is a primitive element of finite field $GF(m)$, then the polynomial $x^{m-1} + x^{m-2} + \cdots + x + 1$ is irreducible over $GF(2)$.*

This lemma is proven in [108].

Lemma 2.3. *If m is a P2 number, then any $m-1$ elements in every equivalence class of $\mathbb{S}_m$ are linearly independent. In other words, all row vectors in a landscape circulant-P2 matrix (and all column vectors in a portrait circulant-P2 matrix) are linearly independent.*

Proof. Let $\boldsymbol{\theta} = (\theta_0, \theta_1, \cdots, \theta_{m\text{-}1}) \in \mathbb{S}_m$, we may view the square circulant matrix $\mathbf{C}_{\boldsymbol{\theta}}$ as a linear feedback shift register sequence $\tilde{\boldsymbol{\theta}} = (\theta_0, \theta_1, \cdots, \theta_{m\text{-}1}, \theta_0, \theta_1, \cdots, \theta_{m\text{-}1}, \cdots)$ of characteristic polynomial $x^m + 1$ over finite field $GF(2)$, according to [77]. Note that $x^m+1 = (x+1)(x^{m-1}+x^{m-2}+\cdots+x+1)$ over $GF(2)$. Let $g(x) = x^{m-1}+x^{m-2}+\cdots+x+1$. Since m is prime, according to Lemma 2.1, sequence $\tilde{\boldsymbol{\theta}}$ has period m. Thus, we only need to consider the following two cases.

Case 1: $\theta_0 \oplus \theta_1 \oplus \cdots \oplus \theta_{m-1} = 0$

In this case, sequence $\tilde{\boldsymbol{\theta}}$ is generated by $g(x)$. Based on Lemma 2.2, $g(x)$ is irreducible over GF(2) if 2 is a primitive element of finite field $GF(m)$. Since the degree of $g(x)$ is equal to $m-1$, then any $m-1$ vectors in $\mathbf{C}_{\boldsymbol{\theta}}$ are linearly independent.

Case 2: $\theta_0 \oplus \theta_1 \oplus \cdots \oplus \theta_{m-1} = 1$

In this case, sequence $\tilde{\boldsymbol{\theta}}$ is not generated by $g(x)$ but by polynomial $x^m + 1$. Since $x^m + 1$ has degree m, then all m vectors in $\mathbf{C}_{\boldsymbol{\theta}}$ are linearly independent.

In summary, if m is a prime number and 2 is a primitive element of $GF(m)$, then any $m-1$ elements in every equivalence class of $\mathbb{S}_m$ are linearly independent. □

The above proof explicitly concludes the following lemma.

Lemma 2.4. *All m row vectors in a square circulant-P2 matrix $\mathbf{C}_{\boldsymbol{\theta}}$ are linearly independent if and only if the Hamming weight of $\boldsymbol{\theta}$ is odd. Consequently, $\mathbf{C}_{\boldsymbol{\theta}}$ is invertible if only if the Hamming weight of $\boldsymbol{\theta}$ is odd.*

The inverse of a square circulant matrix, if it exists, is still a square circulant matrix.

Lemma 2.5. *A landscape circulant-P2 matrix always has a right inverse. That is, for an $(n \times m)$ landscape circulant-P2 matrix $\mathbf{C}$, there exists an $(m \times n)$ matrix $\mathbf{M}$ such that $\mathbf{C} \circ \mathbf{M} = \mathbf{I}_n$, where $\mathbf{I}_n$ is the $(n \times n)$ identity matrix. Likewise, an portrait circulant-P2 matrix always has a left inverse.*

Proof. According to Lemma 2.3, any $(n \times m)$ landscape circulant-P2 matrix has full rank: its rank is equal to n. Therefore, it has a right inverse. The argument for left inverse of a portrait circulant-P2 matrix is same. □

2.2.3 Matrix Operations

A better way to analyze operations on circulant-P2 matrices is to convert them to polynomials, as used in [135]. Every vector can be represented in a polynomial form, as described in the following definition.

Definition 2.4 (Associate Polynomial and Associate Vector)**.** *For a vector $\boldsymbol{\theta} = (\theta_0, \theta_1, \cdots, \theta_{m\text{-}1})$, its associate polynomial $\theta(x)$ in $GF(2)[x]$ is defined as*

$$\theta(x) = \sum_{i=0}^{m-1} \theta_i x^i \ .$$

Correspondingly, $\boldsymbol{\theta}$ is the associate vector of polynomial $\theta(x)$.

Algorithm 2.1 Inverse of circulant-P2 matrix multiplication

Input: m-bit vector $\boldsymbol{k} \in \mathbb{S}_m$, m-bit vector $\boldsymbol{z} = \boldsymbol{y} \circ \mathbf{C}_{\boldsymbol{k}}^{[n\times m]}$
Output: n-bit vector $\boldsymbol{y}$
1: calculate $k^{-1}(x)$ by extended Euclidean algorithm
2: $t(x) \Leftarrow z(x) * k^{-1}(x) \mod f_m(x)$
3: **if** $\mathsf{Hwt}(\boldsymbol{k})$ is odd **then**
4: $\quad \boldsymbol{y} \Leftarrow$ the leftmost n-bit sub-vector of $\boldsymbol{t}$
5: **else**
6: $\quad y_0 \Leftarrow t_0$
7: $\quad i \Leftarrow 1$
8: $\quad$ **while** $i < n$ **do**
9: $\quad\quad y_i \Leftarrow y_{i-1} \oplus t_i$
10: $\quad\quad i \Leftarrow i + 1$

Henceforth, we will freely use those two forms to represent a vector. If we define a vector by one form, then we can use the other representation without explicit explanations.

Matrix Operations in Polynomial Forms

We define $f_m(x) = x^m + 1$, a polynomial in $GF(2)[x]$. Let $\boldsymbol{\phi}, \boldsymbol{k}, \boldsymbol{z} \in \mathbb{S}_m$. We now work with polynomials modulo $f_m(x)$, so that the cyclic shift can be effected by polynomial multiplication modulo $f_m(x)$. That is, the vector $(\boldsymbol{\phi} \ggg i)$, or equivalently $\boldsymbol{\phi} \circ \mathbf{C}_{\boldsymbol{e}_i}$, where $0 \le i \le m-1$, is associated with the polynomial

$$\phi(x) * x^i \mod f_m(x) \ ;$$

reducing modulo $f_m(x)$ achieves the effect of the cyclic shift. Computing $\boldsymbol{\phi} \circ \mathbf{C}_{\boldsymbol{k}}$ combines the several cyclic shifts on $\boldsymbol{\phi}$, each of which is decided by a different bit one in $\boldsymbol{k}$. Subsequently, the computation of $\boldsymbol{z} = \boldsymbol{\phi} \circ \mathbf{C}_{\boldsymbol{k}}$, or $\mathbf{C}_{\boldsymbol{z}} = \mathbf{C}_{\boldsymbol{\phi}} \circ \mathbf{C}_{\boldsymbol{k}}$, can be performed by

$$z(x) = \phi(x) * k(x) \mod f_m(x) \ .$$

It is clear from the above equation that $\boldsymbol{z} = \boldsymbol{\phi} \circ \mathbf{C}_{\boldsymbol{k}} = \boldsymbol{k} \circ \mathbf{C}_{\boldsymbol{\phi}}$.

An efficient method of calculating the right inverse for a landscape circulant-P2 matrix is described in Algorithm 2.1. Main technique in Algorithm 2.1 is adopted from [135], and we develop the solution for the case of $\mathsf{Hwt}(\boldsymbol{k})$ being even. This algorithm applies to all kinds of circulant-P2 matrices' inverses if they exist.

Correctness Proof of Algorithm 2.1. Let $\boldsymbol{\phi} = \boldsymbol{y} || \mathbf{0}_{m-n}$, thus $\boldsymbol{z} = \boldsymbol{\phi} \circ \mathbf{C}_{\boldsymbol{k}}$.

We can use the extended Euclidean algorithm on input polynomials $k(x)$ and $f_m(x)$ to find polynomials $k^{-1}(x)$—the general inverse of $k(x)$— and $w(x)$ such that

$$k^{-1}(x) * k(x) + w(x) * f_m(x) = g(x) \ ,$$

where $g(x) = \gcd(k(x), f_m(x))$.

If $\mathsf{Hwt}(\boldsymbol{k})$ is odd, then $\mathbf{C}_{\boldsymbol{k}}$ is invertible, according to Lemma 2.4. In other words, $g(x) = 1$. Therefore, $\phi(x) = t(x) = z(x) * k^{-1}(x) \mod f_m(x)$.

If $\mathsf{Hwt}(\boldsymbol{k})$ is even, $\mathbf{C}_{\boldsymbol{k}}$ is not invertible, namely $g(x) \neq 1$. According to Lemma 2.2, the polynomial $x^{m-1} + x^{m-2} + \cdots + x + 1$ is irreducible; thus the factorization of $f_m(x)$ is equal to $(x+1)(x^{m-1} + x^{m-2} + \cdots + x + 1)$. Since $\boldsymbol{k}$ is neither $\mathbf{0}_m$ nor $\mathbf{1}_m$, $k(x)$ is not equal to $x^{m-1} + x^{m-2} + \cdots + x + 1$. Therefore, $g(x) = x + 1$. Consequently, polynomial $t(x) = z(x) * s(x) \mod f_m(x)$ is associated with vector

$$\boldsymbol{t} = \boldsymbol{\phi} \circ \mathbf{C}_{\boldsymbol{g}} \ .$$

Since $\boldsymbol{\phi} = \boldsymbol{y}||\mathbf{0}_{m-n}$, then

$$y_i = \begin{cases} t_0 & \text{if } i = 0 \\ y_{i-1} \oplus t_i & \text{if } 1 \leq i \leq n-1 \end{cases} \ . \tag{2.2}$$

Let $\boldsymbol{t}'$ be the leftmost n-bit sub-vector of $\boldsymbol{t}$. For future reference, we denote by $\boldsymbol{y} = \mathsf{Tran}(\boldsymbol{t}')$ the transformation in Equation (2.2). Correspondingly, $\boldsymbol{t}' = \mathsf{Tran}^{-1}(\boldsymbol{y})$.

In either case, the algorithm correctly outputs $\boldsymbol{y}$. □

Demonstrated Examples with Small Parameters

Two examples with small parameters are provided to demonstrate using Algorithm 2.1. Let $m = 5$, $n = 4$, $\boldsymbol{y} = 1011$. Accordingly, we have $\boldsymbol{\phi} = 10110$ and $\phi(x) = 1 + x^2 + x^3$, $f_m(x) = x^5 + 1$. The following two examples are corresponding to the two cases respectively.

(i) $\boldsymbol{k} = 10011$, a case that $\mathsf{Hwt}(\boldsymbol{k})$ is odd.

Then $k(x) = 1 + x^3 + x^4$; and $z(x) = \phi(x) * k(x) \mod f_m(x) = x^4$.

Given $\phi(x)$ and $z(x)$, determine $k^{-1}(x) = x + x^3 + x^4$ by the extended Euclidean algorithm, and then

$$\phi(x) = t(x) = k^{-1}(x) * z(x) \mod f_m(x) = 1 + x^2 + x^3 \ .$$

Thus $\boldsymbol{\phi} = 10110$ and $\boldsymbol{y} = 1011$.

(ii) $\boldsymbol{k} = 10010$, a case that $\mathsf{Hwt}(\boldsymbol{k})$ is even.

Then $k(x) = 1 + x^3$, and $z(x) = \phi(x) * k(x) \mod f_m(x) = x + x^2$.

Given $\phi(x)$ and $z(x)$, determine $k^{-1}(x) = 1 + x^3$, and then

$$t(x) = k^{-1}(x) * z(x) \mod f_m(x) = 1 + x + x^2 + x^4 \ .$$

That is, $\boldsymbol{t} = 11101$. By Equation (2.2), finally recover $\boldsymbol{y} = 1011$.

Remark on Algorithm 2.1

Remark 2.1. *In Algorithm 2.1, let $\boldsymbol{t}'$ be the leftmost n-bit sub-vector of $\boldsymbol{t}$. It is easy to see that if $\boldsymbol{y}$ in Equation (2.2) is uniformly distributed over $\{0,1\}^n$, then $\boldsymbol{t}'$ is uniformly distributed over $\{0,1\}^n$; and vice versa. Moreover, for an n-bit vector $\boldsymbol{\gamma}$, if $\boldsymbol{\gamma} \oplus \boldsymbol{y}$ is uniformly distributed over $\{0,1\}^n$, then $\boldsymbol{\gamma} \oplus \boldsymbol{t}'$ is uniformly distributed over $\{0,1\}^n$; and vice versa.*

2.2.4 Other Properties

Fact 2.1. *Let m be a prime number satisfying 2 is a primitive element of $GF(m)$. For all vectors in $\mathbb{S}_m$, with respect to matrix multiplication of corresponding square circulant-P2 matrices (or equivalently, modular polynomial multiplication of their associate polynomials),*

1. *All vectors in $\mathbb{S}_m^{\mathrm{o}}$ constitute an Abelian multiplicative group of size $2^{m-1} - 1$, with identity element $\boldsymbol{e}_0$;*
2. *All vectors in $\mathbb{S}_m^{\mathrm{e}}$ constitute an Abelian multiplicative group of size $2^{m-1} - 1$, with identity element $\bar{\boldsymbol{e}}_0$;*
3. *The complement of a vector in $\mathbb{S}_m^{\mathrm{o}}$ is an element in $\mathbb{S}_m^{\mathrm{e}}$, and vice versa;*
4. *If vector $\boldsymbol{\theta} \in \mathbb{S}_m^{\mathrm{o}}$, then $\boldsymbol{\theta} \circ \mathbf{C}_{\bar{\boldsymbol{e}}_0} = \bar{\boldsymbol{\theta}}$; if $\boldsymbol{\theta} \in \mathbb{S}_m^{\mathrm{e}}$, then $\boldsymbol{\theta} \circ \mathbf{C}_{\bar{\boldsymbol{e}}_0} = \boldsymbol{\theta}$;*
5. *For two vectors $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ in $\mathbb{S}_m$, $\boldsymbol{\theta} \circ \mathbf{C}_{\boldsymbol{\phi}} \in \mathbb{S}_m^{\mathrm{o}}$ if and only if $\boldsymbol{\theta}, \boldsymbol{\phi} \in \mathbb{S}_m^{\mathrm{o}}$.*

Proof. Sub-fact 3 is obvious. From the correctness proof of Algorithm 2.1, we can easily get sub-fact 1. As for sub-fact 4, let $\boldsymbol{\theta}' = \boldsymbol{\theta} \circ \mathbf{C}_{\bar{\boldsymbol{e}}_0}$, then θ'_i is equal to $\theta_i \oplus$ parity of $\mathsf{Hwt}(\boldsymbol{\theta})$ (0 for even, 1 for odd) for $i = 0, 1, \cdots, m-1$. Therefore, if $\boldsymbol{\theta} \in \mathbb{S}_m^{\mathrm{o}}$, $\boldsymbol{\theta}' = \boldsymbol{\theta} \oplus \mathbf{1}_m = \bar{\boldsymbol{\theta}}$; if $\boldsymbol{\theta} \in \mathbb{S}_m^{\mathrm{e}}$, $\boldsymbol{\theta}' = \boldsymbol{\theta}$, completes this sub-fact. Sub-facts 2 and 5 can be directly derived from sub-facts 1, 3, and 4. □

2.2.5 A Secure Encryption Against Ciphertext-Only Attack

A symmetric-key encryption scheme based on circulant-P2 matrix can be constructed by:

$$\boldsymbol{z} = \boldsymbol{\theta} \circ \mathbf{C}_{\boldsymbol{k}}^{[(m-1)\times m]} \quad , \tag{2.3}$$

where plaintext $\boldsymbol{\theta}$ is an $(m-1)$-bit random vector and $\boldsymbol{\theta} \neq \mathbf{0}_{m-1}$, encryption key $\boldsymbol{k}$ is randomly selected from $\mathbb{S}_m^{\mathrm{e}}$, and ciphertext $\boldsymbol{z}$ is, subsequently, an element in $\mathbb{S}_m^{\mathrm{e}}$. Accordingly, the sizes of plaintext space, key space, and ciphertext space are all the same: $2^{m-1}-1$. The encryption operation can alternatively represented by $\boldsymbol{z} = \mathsf{Enc}(\boldsymbol{\theta}, \boldsymbol{k})$; and the corresponding decryption, denoted by $\boldsymbol{\theta} = \mathsf{Dec}(\boldsymbol{z}, \boldsymbol{k})$ is performed via Algorithm 2.1.

Secure Against Ciphertext-Only Attack

It is easy to see from the properties of circulant-P2 matrix that by choosing a random vector $k' \in \mathbb{S}_m^{\mathrm{e}}$, a valid $\boldsymbol{\theta}'$ satisfying $\boldsymbol{z} = \boldsymbol{\theta}' \circ \mathbf{C}_{\boldsymbol{k}'}^{[(m-1)\times m]}$ can always be retrieved via Algorithm 2.1; for any different $\boldsymbol{\theta}'$, a different $\boldsymbol{k}'$ can be found to map them to any $\boldsymbol{z}$, and vice versa. This fact guarantees the scheme's security against ciphertext-only attack. In other words, given a ciphertext, an adversary cannot learn any useful information about the encryption key and the plaintext, because each ciphertext is corresponding to $2^{m-1}-1$ distinct combinations of plaintext-key pairs. Thus every plaintext/key is equally possible for any ciphertext. Hence the encryption is semantically secure against ciphertext-only attack as long as plaintexts are random.

Discussions

Alternatively, we may use the encryption: $\boldsymbol{z} = \boldsymbol{\theta} \circ \mathbf{C}_{\boldsymbol{k}}$, where plaintext $\boldsymbol{\theta}$, key $\boldsymbol{k}$, and ciphertext $\boldsymbol{z}$ all belong to $\mathbb{S}_m^{\mathrm{o}}$. The arguments above apply to it, and the encryption scheme with random plaintexts is semantically secure ciphertext-only attack.

In practice, a stand-alone encryption scheme only secure against ciphertext-only attack is rarely useful. However, coupled with the hardness of the LPN problem, this scheme can lead to a succinct, highly efficient, and secure entity authentication scheme, which we will describe in next section.

Tag $(\boldsymbol{k}_1, \boldsymbol{k}_2)$		Reader $(\boldsymbol{k}_1, \boldsymbol{k}_2)$
	$\xleftarrow{\boldsymbol{a}}$	$\boldsymbol{a} \in_R \mathbb{S}_m^{\mathrm{e}}$
$\boldsymbol{b} \in_R \mathbb{S}_m$; $\boldsymbol{v} \in_R \{\{0,1\}^n \mid \Pr[v_j = 1] = \eta,$ where $0 \le j \le n-1\}$; $\boldsymbol{y} = (\boldsymbol{b} \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{v}$; $\boldsymbol{r} \in_R \{0,1\}^{m-n-1}$; $\boldsymbol{z} = (\boldsymbol{y} \| \boldsymbol{r}) \circ \mathbf{C}_{\boldsymbol{k}_2 \oplus \boldsymbol{a}}^{[(m-1)\times m]}$	$\xrightarrow{\boldsymbol{b}, \boldsymbol{z}}$	$\boldsymbol{y} \| \boldsymbol{r} = \mathsf{Dec}(\boldsymbol{z}, \boldsymbol{k}_2 \oplus \boldsymbol{a})$; ACCEPT iff $\mathsf{Hwt}((\boldsymbol{b} \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{y}) \le \tau$

Figure 2.2: LCMQ one-way authentication protocol, where m is a P2 number, $\boldsymbol{k}_1 \xleftarrow{\$} \mathbb{S}_m$ and the parity of $\mathsf{Hwt}(\boldsymbol{k}_1)$ is public, $\boldsymbol{k}_2 \xleftarrow{\$} \mathbb{S}_m^{\mathrm{e}}$, n is the interaction expansion and $n < m$, noise level $\eta \in (0, \frac{1}{2})$, integer pass-threshold $\tau \in (\eta n, \frac{n}{2})$

2.3 LCMQ Protocol

2.3.1 Protocol Specification

The LCMQ protocol with the Bernoulli noise mode is illuminated in Figure 2.2. In this scheme, two m-bit vectors $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ are shared by a tag and a reader as a pair of symmetric authentication keys with one condition that the parities of both keys' Hamming weights are known to the public. Because of the inherent requirement of circulant-P2 matrix that the interaction expansion n should be less than the key length m, and the impact of n on false rates according to Equation (2.1), $n = m - 1$ is recommended in most cases. Similar to conventional identification schemes, one LCMQ authentication procedure consists of two passes: a challenge $\boldsymbol{a}$ by the reader and a response pair $(\boldsymbol{b}, \boldsymbol{z})$ by the tag, rendering $3m$ bits transmission payload. The computation in the tag's side mainly involves two vector/matrix multiplications of roughly m^2 XOR operations by a naive implementation, while the reader additionally, to calculate an inverse of a circulant-P2 matrix, needs to perform the extended Euclidean algorithm, which is surely a trivial requirement to the supposedly powerful reader. As we will argue later, $m = 163$ would suffice to provide 80-bit security, and the LCMQ protocol achieves extraordinary performance in terms of all metrics: storage expense, computational payload, communication cost, and implementation expenditure. Most importantly, we will prove LCMQ secure against general man-in-the-middle attacks. All of those promising properties of LCMQ make it very suitable for the authentication of RFID systems.

General Regulations

In order to prevent malicious behaviors, tag should check if $\boldsymbol{a}$ belongs to $\mathbb{S}_m^{\mathrm{e}}$, and reader should check if $\boldsymbol{b} \in \mathbb{S}_m$ and if $\boldsymbol{z} \in \mathbb{S}_m^{\mathrm{e}}$, upon receiving them; if any of those abnormalities takes place, the participant will terminate this round of authentication. In addition, if $\boldsymbol{y}||\boldsymbol{r} = \mathbf{0}_{m-1}$, technically, tag should repeat its procedure of generating a new $\boldsymbol{y}||\boldsymbol{r}$. Since the probability of such an event is negligible, equal to $1/(2^{m-1}-1)$, tag need not bother to take this countermeasure. Note that it is impossible that $\boldsymbol{k}_2 \oplus \boldsymbol{a} = \mathbf{1}_m$ since both $\boldsymbol{k}_2$ and $\boldsymbol{a}$ are in $\mathbb{S}_m^{\mathrm{e}}$. If $\boldsymbol{k}_2 \oplus \boldsymbol{a} = \mathbf{0}_m$, the LCMQ protocol fails, but such a case only takes place with the negligible probability $1/(2^{m-1}-1)$. Therefore, we can safely presume that it would never happen and will not consider it in the rest of the chapter for simplicity.

Structure Comparisons

The proposed LCMQ protocol, thought it is still LPN-based, has a different structure from the HB-like protocols [91, 76]. By the encryption $\boldsymbol{z} = \mathsf{Enc}(\boldsymbol{y}||\boldsymbol{r}, \boldsymbol{k}_2 \oplus \boldsymbol{a})$, protocol LCMQ conceals the LPN answer $\boldsymbol{y}$ from adversaries such that it can use a smaller key length, which is a vital factor to determine protocols computation and communication performance. More importantly, the encryption/decryption operations provide an implicit integrity mechanism for $(\boldsymbol{a}, \boldsymbol{z})$. Benefited from linear independence of circulant-P2 matrix vectors, any alteration on $(\boldsymbol{a}, \boldsymbol{z})$ will render the authentication to fail with an overwhelming probability, so does the case of manipulating $\boldsymbol{b}$. In addition, there is no correlation effect of simultaneously manipulating $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{z}$ (The error bits introduced by changing one can be canceled off, to a notable extent, by the error bits from altering others.), thus the LCMQ protocol overcomes the flaws in the HB-like protocols [91, 76] that leave them subject to the OOV attack.

2.3.2 Security Models Definitions

To formally define security models, we denote an LCMQ authentication system by a pair of probabilistic functions $(\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}, \mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau})$, namely a tag function $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$ and a reader function $\mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}$.

Definition of Adversary's Advantage

The fundamental objective of an adversary in the entity authentication protocol is to impersonate a legitimate tag. By replying a random vector as an authentication response, the probability that an adversary impersonating the tag will success is the false positive rate P_{FP}. This is the best soundness error that we can achieve for an authentication

protocol. Therefore, we define the advantage of an adversary $\mathcal{A}$ against LCMQ in a model as its overall success probability over P_{FP} in impersonating the tag.

DET-model

Definition 2.5 (DET-model)**.** *In the DET-model, which is identical to the detection-based-model used in [91, 93, 94, 76], the DET attack is carried out in two phases:*

- Phase 1*: Adversary $\mathcal{A}$ interacts q times with $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$. On the ith invocation, $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$ takes a challenge vector $\boldsymbol{a}_i$ from $\mathcal{A}$ as input, selects a random vector $\boldsymbol{b}_i \in \mathbb{S}_m$, generates a noise vector $\boldsymbol{v}_i$ according to the Bernoulli noise mode, and calculates $\boldsymbol{y}_i = (\boldsymbol{b}_i \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{v}_i$. Furthermore, $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$ internally generates a random $(m\text{-}n\text{-}1)$-bit vector $\boldsymbol{r}_i$, and computes $\boldsymbol{z}_i = (\boldsymbol{y}_i || \boldsymbol{r}_i) \circ \mathbf{C}_{\boldsymbol{k}_2 \oplus \boldsymbol{a}_i}^{[(m\text{-}1)\times m]}$. Then $\boldsymbol{b}_i$ and $\boldsymbol{z}_i$ are transmitted to $\mathcal{A}$.*
- Phase 2*: Adversary $\mathcal{A}$ receives a random challenge $\widehat{\boldsymbol{a}}$ from $\mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}$, and then outputs $(\widehat{\boldsymbol{b}}, \widehat{\boldsymbol{z}})$ corresponding to $\widehat{\boldsymbol{a}}$, intended to pass the verification of $\mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}$ with advantage*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathrm{DET}}(m,\eta,n,\tau) \;\stackrel{def}{=}\; & \mathsf{Pr}[\boldsymbol{k}_1 \stackrel{\$}{\leftarrow} \mathbb{S}_m, \boldsymbol{k}_2 \stackrel{\$}{\leftarrow} \mathbb{S}_m^{\mathrm{e}}, \mathcal{A}^{\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}}(1^m) : \\ & \langle \mathcal{A}, \mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}, \mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}\rangle = \mathrm{ACCEPT}] - P_{\mathrm{FP}} \;\; .\end{aligned}$$

MIM-model

Definition 2.6 (MIM-model)**.** *In the MIM-model, the MIM attack is conducted in two phases:*

- Phase 1*: Adversary $\mathcal{A}$ manipulates any communications between $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$ and $\mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}$ for q executions. Figure 2.3 depicts the ith manipulation, which simulates a full MIM attacker. We define three interference vectors: $\boldsymbol{\alpha}_i = \boldsymbol{a}_i \oplus \boldsymbol{a}'_i, \boldsymbol{\beta}_i = \boldsymbol{b}_i \oplus \boldsymbol{b}'_i$ and $\boldsymbol{\zeta}_i = \boldsymbol{z}_i \oplus \boldsymbol{z}'_i$.*
- Phase 2*: Adversary $\mathcal{A}$ receives a random challenge $\widehat{\boldsymbol{a}}$ from $\mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}$, and then outputs $(\widehat{\boldsymbol{b}}, \widehat{\boldsymbol{z}})$ corresponding to $\widehat{\boldsymbol{a}}$, intended to pass the verification of $\mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}$ with advantage*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\mathrm{MIM}}(m,\eta,n,\tau) \;\stackrel{def}{=}\; & \mathsf{Pr}[\boldsymbol{k}_1 \stackrel{\$}{\leftarrow} \mathbb{S}_m, \boldsymbol{k}_2 \stackrel{\$}{\leftarrow} \mathbb{S}_m^{\mathrm{e}}, \mathcal{A}^{\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n},\mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}}(1^m) : \\ & \langle \mathcal{A}, \mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}, \mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}\rangle = \mathrm{ACCEPT}] - P_{\mathrm{FP}} \;\; .\end{aligned}$$

The MIM-model is a very strong security from the adversary's perspective and it is easy to see that the DET-model is a limited version of the MIM-model. An authentication

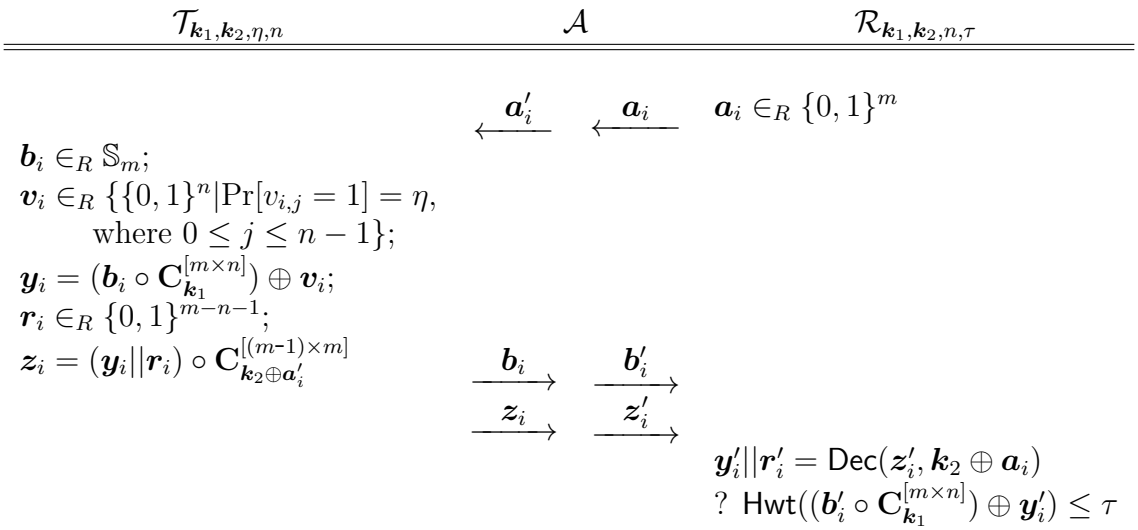


Figure 2.3: The ith manipulation to the LCMQ protocol by an adversary in the MIM-model

protocol provably secure in MIM-model will naturally resist all probabilistic polynomial-time (PPT) attacks. In the following, we first provide a concrete security reductionist proof from the DET-model to the MIM-model for the LCMQ protocol; then we prove its security in the DET-model based on the hardness of the LPN problem, with some reasonable assumptions.

2.3.3 Reduction from DET-model to MIM-model

Theorem 2.1. *If there is an adversary $\mathcal{A}$ attacking the LCMQ protocol in the MIM-model, modifying q executions of the protocol between an honest tag and an honest reader, running in time t, and achieving $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{MIM}}(m,\eta,n,\tau) \ge \delta$, then there exists an adversary $\mathcal{A}'$ attacking the LCMQ protocol in the DET-model, interacting at most q oracle queries, running in time $O(t)$, and achieving $\mathsf{Adv}_{\mathcal{A}'}^{\mathrm{DET}}(m,\eta,n,\tau) \ge \delta - q\epsilon(P_{\mathrm{FP}}+\delta)$ for some negligible function ϵ, under the assumption that P_{FP} and P_{FN} are negligible. Hence, if protocol LCMQ is secure in the DET-model, then it is provably secure in the MIM-model.*

Proof. In Phase 1, $\mathcal{A}'$ can readily simulate the honest tag for $\mathcal{A}$ since $\mathcal{A}'$ has access to $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$. Therefore, we only need to simulate $\mathcal{R}_{\boldsymbol{k}_1,\boldsymbol{k}_2,n,\tau}$ for $\mathcal{A}$. Similar to the proof method for the Random-HB$^{\#}$ protocol [76], $\mathcal{A}'$ launches Phase 1 of adversary $\mathcal{A}$, and simulates the tag and the reader q times as follows:

1. $\mathcal{A}'$ sends a random vector $\boldsymbol{a}_i$ as the challenge of the simulated reader, and let $\mathcal{A}$ modify it to $\boldsymbol{a}_i'$; then $\mathcal{A}'$ forwards $\boldsymbol{a}_i'$ to $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$.

2. $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$ faithfully responds with $(\boldsymbol{b}_i, \boldsymbol{z}_i = (((\boldsymbol{b}_i \circ \mathbf{C}_{\boldsymbol{k}_1}^{[n\times m]}) \oplus \boldsymbol{v}_i)||\boldsymbol{r}_i) \circ \mathbf{C}_{\boldsymbol{k}_2 \oplus \boldsymbol{a}_i'}^{[(m-1)\times m]})$ to $\mathcal{A}'$, which relays $(\boldsymbol{b}_i, \boldsymbol{z}_i)$ to $\mathcal{A}$. Then $\mathcal{A}$ alters them to $(\boldsymbol{b}_i', \boldsymbol{z}_i')$, and uses $(\boldsymbol{b}_i', \boldsymbol{z}_i')$ as the authentication response to $\mathcal{A}'$.

3. During the interactions, if $\boldsymbol{a}_i' \notin \mathbb{S}_m^{\mathrm{e}}$ or $\boldsymbol{b}_i' \notin \mathbb{S}_m$ or $\boldsymbol{z}_i' \notin \mathbb{S}_m^{\mathrm{e}}$, $\mathcal{A}'$ terminates the iteration and proceeds with the next, abiding by the protocol specification.

4. If $\boldsymbol{a}_i' = \boldsymbol{a}_i$ and $\boldsymbol{b}_i' = \boldsymbol{b}_i$ and $\boldsymbol{z}_i' = \boldsymbol{z}_i$, $\mathcal{A}'$ outputs "ACCEPT" to $\mathcal{A}$ as the authentication result of the simulated reader; if $\boldsymbol{a}_i' = \boldsymbol{a}_i$ and $\boldsymbol{z}_i' = \boldsymbol{z}_i$ and $\boldsymbol{\beta} = \mathbf{1}_m$ and $\mathsf{Hwt}(\boldsymbol{k}_1)$ is odd, $\mathcal{A}'$ outputs "ACCEPT" too; elsewise, it outputs "REJECT".

After Phase 1, $\mathcal{A}'$ launches Phase 2 of $\mathcal{A}$. Since Phase 2 in the DET-model is identical to that in the MIM-model, $\mathcal{A}'$ just replicates $\mathcal{A}$'s behavior with the real reader, with the same objective of passing the authentication. Therefore, if $\mathcal{A}$ achieves $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{MIM}}(m, \eta, n, \tau) \geq \delta$, then the probability of $\mathcal{A}'$ successfully impersonating a valid tag is equal to the success probability of $\mathcal{A}$, i.e., $P_{\mathrm{FP}} + \delta$, on the condition that the reader is correctly simulated by $\mathcal{A}'$ in Phase 1.

We denote by P_{err} the probability of $\mathcal{A}'$ wrongly simulating the reader for $\mathcal{A}$ in one iteration of Phase 1. Executions in Phase 1 can be divided into four different cases:

Case 1: $\boldsymbol{a}_i' = \boldsymbol{a}_i$ and $\boldsymbol{z}_i' = \boldsymbol{z}_i$ and $\boldsymbol{b}_i' = \boldsymbol{b}_i$.

In this case, $\mathcal{A}'$ outputs "ACCEPT", and fails at simulating the reader with a probability equal to the false negative rate P_{FN}.

Case 2: $\boldsymbol{a}_i' = \boldsymbol{a}_i$ and $\boldsymbol{z}_i' = \boldsymbol{z}_i$ but $\boldsymbol{b}_i \neq \boldsymbol{b}_i$.

Thus $\boldsymbol{y}_i' = \boldsymbol{y}_i$. Since

$$(\boldsymbol{b}_i' \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{y}_i = ((\boldsymbol{b}_i' \oplus \boldsymbol{b}_i) \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{v}_i = (\boldsymbol{\beta}_i \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{v}_i = (\boldsymbol{k}_1 \circ \mathbf{C}_{\boldsymbol{\beta}_i}^{[m\times n]}) \oplus \boldsymbol{v}_i \ ,$$

the authentication result in this case is equivalently decided by

$$\mathsf{Hwt}((\boldsymbol{k}_1 \circ \mathbf{C}_{\boldsymbol{\beta}_i}^{[m\times n]}) \oplus \boldsymbol{v}_i) \leq \tau \ .$$

Let

$$\boldsymbol{d}_i' = \boldsymbol{k}_1 \circ \mathbf{C}_{\boldsymbol{\beta}_i}^{[m\times n]} \tag{2.4}$$

be the error vector added by $\mathcal{A}$ by changing $\boldsymbol{b}_i$.

If $\boldsymbol{\beta}_i \in \mathbb{S}_k$, according to Lemma 2.3, all column vectors in the portrait circular-P2 matrix $\mathbf{C}_{\boldsymbol{\beta}_i}^{[m\times n]}$ are linearly independent. Following the same argument in Theorem 2 of [93], $\boldsymbol{d}_i'$ is uniformly distributed over $\{0,1\}^n$, as the column vectors of $\mathbf{C}_{\boldsymbol{\beta}_i}^{[m\times n]}$ are linearly independent. Thus the resulting error vector $\boldsymbol{d}_i' \oplus \boldsymbol{v}_i$ follows the uniform distribution over $\{0,1\}^n$ since

$\mathcal{A}$ has no extra knowledge about the random noise vector $\boldsymbol{v}_i$. As a result, the probability of $\mathcal{A}'$ wrongly outputting "REJECT" is exactly the same as the false positive rate P_{FP}.

If $\boldsymbol{\beta}_i = \mathbf{1}_m$ and $\mathsf{Hwt}(\boldsymbol{k}_1)$ is even, then $\boldsymbol{d}'_i = \mathbf{0}_n$, and the probability of $\mathcal{A}'$ wrongly outputting "ACCEPT" is exactly P_{FN}. When $\boldsymbol{\beta}_i = \mathbf{1}_m$ and $\mathsf{Hwt}(\boldsymbol{k}_1)$ is odd, then $\boldsymbol{d}'_i = \mathbf{1}_n$. Consequently, the probability of $\mathcal{A}'$ wrongly outputting "REJECT" is $\sum_{i=\tau+1}^{n} \binom{n}{i}(1-\eta)^i \eta^{n-i}$, which is always less than P_{FN} since $\eta n < \tau < n/2$.

Overall in this case, $P_{\text{err}} \leq \max(P_{\text{FP}}, P_{\text{FN}})$.

Case 3: $\boldsymbol{b}'_i = \boldsymbol{b}_i$ and at least one of $\boldsymbol{\alpha}_i$ and $\boldsymbol{\zeta}_i$ is not equal to $\mathbf{0}_m$.

The error vector introduced by the adversary through changing $\boldsymbol{z}_i$ and/or $\boldsymbol{a}_i$ is denoted by

$$\boldsymbol{d}''_i = \boldsymbol{y}_i \oplus \boldsymbol{y}'_i \ . \tag{2.5}$$

Correspondingly, the authentication result is decided by

$$\mathsf{Hwt}(\boldsymbol{d}''_i \oplus \boldsymbol{v}_i) \leq \tau \ .$$

Recall that $\boldsymbol{a}_i, \boldsymbol{a}'_i, \boldsymbol{z}_i, \boldsymbol{z}'_i \in \mathbb{S}^{\text{e}}_m$; thus $\boldsymbol{\alpha}_i \neq \mathbf{1}_m$ and $\boldsymbol{\zeta}_i \neq \mathbf{1}_m$. Let $\boldsymbol{\kappa}_i = \boldsymbol{k}_2 \oplus \boldsymbol{a}_i$, $\boldsymbol{\kappa}'_i = \boldsymbol{k}_2 \oplus \boldsymbol{a}'_i$, $s_i(x)$ and $s'_i(x)$ be the general inverses of $\kappa_i(x)$ and $\kappa'_i(x)$ respectively. In addition, let $\boldsymbol{\lambda}_i = \boldsymbol{s}_i \oplus \boldsymbol{s}'_i$. It is clear that $\boldsymbol{\lambda}_i \in \mathbb{S}^{\text{e}}_m$ if $\boldsymbol{\alpha}_i \in \mathbb{S}^{\text{e}}_m$ and $\boldsymbol{\lambda}_i = \mathbf{0}_m$ if $\boldsymbol{\alpha}_i = \mathbf{0}_m$. Moreover, $\mathcal{A}$ does not know any additional information about $s_i(x)$ and $s'_i(x)$; otherwise, $\mathcal{A}$ must have recovered some of $\boldsymbol{k}_2$.

From equations

$$\begin{aligned} s_i(x)(k_2(x) + a_i(x)) &\equiv 1 + x \mod f_m(x) \ , \\ (s_i(x) + \lambda_i(x))(k_2(x) + a_i(x) + \alpha_i(x)) &\equiv 1 + x \mod f_m(x) \ , \end{aligned}$$

we have $\lambda_i(x)(k_2(x) + a_i(x)) + s_i(x)\alpha_i(x) \equiv 0 \mod f_m(x)$, and then

$$\lambda_i(x)(1 + x) + s_i^2(x)\alpha_i(x) \equiv 0 \mod f_m(x) \ . \tag{2.6}$$

If $\mathcal{A}$ knows $\lambda_i(x)$, he can recover $s_i^2(x)$ by the equation above, and then $s_i(x)$ and $k_2(x)$ are leaked. Therefore, even though $\mathcal{A}$ can freely choose $\boldsymbol{\alpha}_i$, if $\boldsymbol{k}_2$ is unknown to him, $\mathcal{A}$ should not have useful information about $\boldsymbol{\lambda}_i$.

Let $t_i(x) = z_i(x) * s'_i(x) \mod f_m(x)$ and $t'_i(x) = z'_i(x) * s_i(x) \mod f_m(x)$; let $\boldsymbol{\gamma}_i, \boldsymbol{\gamma}'_i$ be the leftmost n-bit sub-vectors of $\boldsymbol{t}_i, \boldsymbol{t}'_i$ respectively. Then we have

$$\begin{aligned} t'_i(x) + t_i(x) &= z'_i(x) * s_i(x) + z_i(x) * (s'_i(x) - s_i(x) + s_i(x)) \mod f_m(x) \\ &= (z'_i(x) + z_i(x)) * s_i(x) + z_i(x) * (s'_i(x) + s_i(x)) \mod f_m(x) \\ &= \zeta_i(x) * s_i(x) + z_i(x) * \lambda_i(x) \mod f_m(x) \ . \end{aligned}$$

Subsequently,

$$\boldsymbol{d}_i'' = \mathsf{Tran}(\boldsymbol{\gamma}_i' \oplus \boldsymbol{\gamma}_i) = \mathsf{Tran}((\boldsymbol{s}_i \circ \mathbf{C}_{\boldsymbol{\zeta}_i}^{[m\times n]}) \oplus (\boldsymbol{z}_i \circ \mathbf{C}_{\boldsymbol{\lambda}_i}^{[m\times n]})) \ . \tag{2.7}$$

If $\boldsymbol{\zeta}_i \in \mathbb{S}_m$, then $\boldsymbol{s}_i \circ \mathbf{C}_{\boldsymbol{\zeta}_i}^{[m\times n]}$ is uniformly distributed over $\{0,1\}^n$; if $\boldsymbol{\alpha}_i \in \mathbb{S}_m$, which implies $\boldsymbol{\lambda}_i \in \mathbb{S}_m$, then $\boldsymbol{z}_i \circ \mathbf{C}_{\boldsymbol{\lambda}_i}^{[m\times n]}$ is uniformly distributed over $\{0,1\}^n$. Note that $\boldsymbol{s}_i$ and $\boldsymbol{\lambda}_i$ (if $\boldsymbol{\alpha}_i \neq \mathbf{0}_m$) are unknown to $\mathcal{A}$, and $\mathcal{A}$ receives $\boldsymbol{z}_i$ only after he has revealed his selection of $\boldsymbol{\alpha}_i$. Therefore, if only one of $\boldsymbol{\alpha}_i$ and $\boldsymbol{\zeta}_i$ is in $\mathbb{S}_m$, then $\boldsymbol{d}_i''$ is uniformly distributed oven $\{0,1\}^n$, by Remark 2.1.

If both $\boldsymbol{\alpha}_i$ and $\boldsymbol{\zeta}_i$ are not equal to $\mathbf{0}_m$, since

$$(t_i'(x) + t_i(x))(1+x) \ \equiv \ s_i(x)(\zeta_i(x)(1+x) + s_i(x)\alpha_i(x)) \mod f_m(x) \ ,$$

and $\mathcal{A}$ cannot choose a valid pair of $(\boldsymbol{\alpha}_i, \boldsymbol{\zeta}_i)$ satisfying $\zeta_i(x)(1+x) + s_i(x)\alpha_i(x) \equiv 0$ or $1 + x + x^2 + \ldots + x^{m-1} \mod f_m(x)$ without knowledge of $s_i(x)$, then $\boldsymbol{d}_i''$ is still uniformly distributed oven $\{0,1\}^n$.

As a result, in this case, $\mathcal{A}'$ erroneously outputs "REJECT" with probability P_{FP}.

Case 4: $\boldsymbol{b}_i \neq \boldsymbol{b}_i'$ and at least one of $\boldsymbol{\alpha}_i$ and $\boldsymbol{\zeta}_i$ is not equal to $\mathbf{0}_m$.

This case is the combination of Case 2 and Case 3, and the authentication result is determined by

$$\mathsf{Hwt}(\boldsymbol{d}_i' \oplus \boldsymbol{d}_i'' \oplus \boldsymbol{v}_i) \leq \tau \ ,$$

where $\boldsymbol{d}_i'$ and $\boldsymbol{d}_i''$ are defined in (2.4) and (2.5) respectively. Applying the deductions in the previous two cases, $\boldsymbol{d}_i'$ and $\boldsymbol{d}_i''$ are uniformly distributed over $\{0,1\}^n$. Because $\boldsymbol{k}_1$ used in (2.4) and $\boldsymbol{k}_2$ used in (2.5) are independent, and these is no relation between $\boldsymbol{d}_i'$ and $\boldsymbol{d}_i''$, then $\boldsymbol{d}_i' \oplus \boldsymbol{d}_i''$ is still uniformly distributed over $\{0,1\}^n$. Consequently, the probability of $\mathcal{A}'$ wrongly outputting "REJECT" is P_{FP}.

Summing all cases up, $\mathcal{A}'$ fails at simulating the reader in one execution at most with probability $\epsilon = \max(P_{\mathrm{FN}}, P_{\mathrm{FP}})$. Thus the probability of $\mathcal{A}'$ correctly simulating the reader in Phase 1 is not less than $1 - q\epsilon$, and adversary $\mathcal{A}'$ impersonates a valid tag at least with probability $(P_{\mathrm{FP}} + \delta)(1 - q\epsilon)$. Therefore, $\mathcal{A}'$ can achieve advantage

$$\mathsf{Adv}_{\mathcal{A}'}^{\mathrm{DET}}(m, \eta, n, \tau) \geq (P_{\mathrm{FP}} + \delta)(1 - q\epsilon) - P_{\mathrm{FP}} = \delta - q\epsilon(P_{\mathrm{FP}} + \delta) \ .$$

With properly chosen parameters such that P_{FN} and P_{FP} are negligible, if δ is non-negligible, then $\mathsf{Adv}_{\mathcal{A}'}^{\mathrm{DET}}(m, \eta, n, \tau)$ is non-negligible. Thus if the LCMQ protocol is secure in the DET-model, then it is secure in the MIM-model. □

2.3.4 Security in the DET-model

We first prove that the intractability of the LPN problem implies the pseudorandomness of $\boldsymbol{y}_i$ in the LCMQ protocol, and then use it to prove the LCMQ protocol's security in the DET-model.

The following probability distributions are used in the deductions.

- $\texttt{Bio}_{n,\eta}$: the distribution of n-bit vector in which each bit independently follows the Bernoulli distribution of parameter η.
- $\mathcal{D}_{\boldsymbol{k},n,\eta}$: for $\boldsymbol{k} \in \mathbb{S}_m$, the distribution of $(m+n)$-bit string:

$$\{\boldsymbol{b} \xleftarrow{\$} \mathbb{S}_m, \boldsymbol{v} \xleftarrow{\$} \texttt{Bio}_{n,\eta} : (\boldsymbol{b}, \boldsymbol{y} \leftarrow (\mathbf{C}_{\boldsymbol{b}}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v})\} \ .$$

- $\mathcal{U}_{m+n}$: the distribution of $(m+n)$-bit string:

$$\{\boldsymbol{b} \xleftarrow{\$} \mathbb{S}_m, \boldsymbol{y} \xleftarrow{\$} \{0,1\}^n : (\boldsymbol{b}, \boldsymbol{y})\} \ .$$

Lemma 2.6. *Assuming the intractability of the LPN problem, $\mathcal{D}_{\boldsymbol{k},1,\eta}$ and $\mathcal{U}_{m+1}$ are indistinguishable for all PPT algorithms.*

Proof. Lemma 1 in [93] has proven that if $\boldsymbol{b}$ is uniformly chosen from $\{0,1\}^m$ in $\mathcal{D}_{\boldsymbol{k},1,\eta}$ and $\mathcal{U}_{m+1}$, and there is no restriction on m having to be a prime number satisfying 2 is a primitive element of $GF(m)$, then $\mathcal{D}_{\boldsymbol{k},1,\eta}$ and $\mathcal{U}_{m+1}$ are indistinguishable for all PPT algorithms, assuming the intractability of the LPN problem. Apparently, the discrepancy between $\boldsymbol{b} \xleftarrow{\$} \mathbb{S}_m$ and $\boldsymbol{b} \xleftarrow{\$} \{0,1\}^m$ is ignorable. As for the requirement of m being prime, it is trivial according to the prime number theorem, which describes the asymptotic distribution of the prime numbers. Moreover, even though there is no deterministic number theory result regarding the distribution of a special class of prime number m satisfying 2 is a primitive element of $GF(m)$, according to Artin conjecture [80], the set of such primes is infinite and its density inside the set of primes is equal to Artin's constant, which can be expressed as an infinite product

$$C_{\text{Artin}} = \prod_{q \text{ prime}} \left(1 - \frac{1}{q(q-1)}\right) = 0.3739558136\ldots$$

Therefore, we conclude that $\mathcal{D}_{\boldsymbol{k},1,\eta}$ and $\mathcal{U}_{m+1}$ are indistinguishable for all probabilistic polynomial-time algorithms. □

Lemma 2.7. *If there is no PPT algorithm capable of distinguishing $\mathcal{D}_{\boldsymbol{k},1,\eta}$ from $\mathcal{U}_{m+1}$, then $\mathcal{D}_{\boldsymbol{k},n,\eta}$ and $\mathcal{U}_{m+n}$ are indistinguishable for all PPT algorithms.*

Proof. For $0 \leq i < j \leq n$, let $\mathcal{H}^{i,j}$ denote a hybrid probability distribution

$$\{b_0, b_1, \ldots, b_{m-1}, r_0, r_1, \ldots, r_{i-1}, y_i, y_{i+1}, \ldots, y_{j-1}, r_j, r_{j+1}, \ldots, r_{n-1}\},$$

where all b_*'s and y_*'s are corresponding to those bits with the same notation in $\mathcal{D}_{\boldsymbol{k},n,\eta}$, all r_*'s are independent and uniformly selected over $\{0,1\}$, and the convention holds that string $(r_j, r_{j+1}, \ldots, r_{n-1})$ is null if $j = n$. In addition, we denote by $p^{i,j}$ the maximal advantage of any PPT algorithm distinguishing $\mathcal{H}^{i,j}$ from $\mathcal{U}_{m+n}$. It is clear that $\mathcal{H}^{0,n} = \mathcal{D}_{\boldsymbol{k},n,\eta}$.

Let δ be the upper bound of any PPT algorithm's advantage distinguishing $\mathcal{D}_{\boldsymbol{k},1,\eta}$ from $\mathcal{U}_{m+1}$, we prove $p^{0,n} \leq n\delta$ by the following induction.

Basic Case: Since $\mathcal{H}^{i,i+1}$ is essentially $\mathcal{D}_{\boldsymbol{k}\ggg i,1,\eta}$ inserting $n-1$ random bits, $p^{i,i+1} \leq \delta$.

Inductive Step: Assuming $p^{i,j} = (j-i)\eta$. Because $y_i, y_{i+1}, \ldots, y_j$ are linearly independent by Lemma 2.3, we have

$$p^{i,j+1} \leq \max_{i<l<j+1}(p^{i,l} + p^{l,j+1}) = (j+1-i)\delta \ .$$

If δ is negligible, then $p^{0,n} \leq n\delta$ is also negligible. In other words, $\mathcal{D}_{\boldsymbol{k},n,\eta}$ and $\mathcal{U}_{m+n}$ are indistinguishable for all PPT algorithms if there is no PPT algorithm capable of distinguishing $\mathcal{D}_{\boldsymbol{k},1,\eta}$ from $\mathcal{U}_{m+1}$. □

Theorem 2.2. *If $\mathcal{D}_{\boldsymbol{k},n,\eta}$ and $\mathcal{U}_{m+n}$ are indistinguishable for all PPT algorithms, then all PPT adversaries are only able to attack the LCMQ protocol in the DET-model with a negligible advantage. Therefore, if the LPN problem is intractable, the LCMQ protocol is secure in the DET-model.*

Proof. For the LCMQ protocol in Phase 1 of the DET-model, let $\boldsymbol{\kappa}_i = \boldsymbol{k}_2 \oplus \boldsymbol{a}_i$, and $\boldsymbol{\theta}_i = \boldsymbol{y}_i || \boldsymbol{r}_i$. In Section 2.2.3, we have demonstrated that if $\boldsymbol{\kappa}_i$ is secret, then the encryption $\boldsymbol{z}_i = \mathsf{Enc}(\boldsymbol{\theta}_i, \boldsymbol{\kappa}_i)$ is secure against ciphertext-only attack as long as ciphertext $\boldsymbol{\theta}_i$ is random. If $\mathcal{D}_{\boldsymbol{k},n,\eta}$ and $\mathcal{U}_{m+n}$ are indistinguishable for all PPT algorithms, then $\boldsymbol{\theta}$ is random from the perspective of any PPT adversary in the DET-model. Even though an adversary $\mathcal{A}$ now can freely choose $\boldsymbol{a}_i$ in the LCMQ protocol, this encryption is still secure against $\mathcal{A}$. To see this point, we can think in this way: During ith invocation of Phase 1 in the DET-model, $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$ may respond to the adversary's challenge $\boldsymbol{a}_i$ with $(\boldsymbol{b}_i \xleftarrow{\$} \mathbb{S}_m, \boldsymbol{z}_i \xleftarrow{\$} \mathbb{S}_m^{\mathrm{e}})$; regardless of any value of $\boldsymbol{a}_i$, the decryption result $\boldsymbol{\theta}_i = \mathsf{Dec}(\boldsymbol{z}_i, \boldsymbol{k} \oplus \boldsymbol{a}_i)$ follows the uniform distribution over $\mathbb{S}_m^{\mathrm{e}}$. In this regard, it simulates the action by $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$ from the point of view of $\mathcal{A}$. In other words, we here use a random oracle: the real value of $\boldsymbol{y}_i$ does not come from $(\boldsymbol{b}_i \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{v}_i$, but is determined by the decryption result $\boldsymbol{\theta}_i = \mathsf{Dec}(\boldsymbol{z}_i, \boldsymbol{k}_2 \oplus \boldsymbol{a}_i)$. Clearly, those random responses will not leak any useful information to $\mathcal{A}$.

Now $\mathcal{A}$ proceeds with Phase 2, required to output $(\widehat{\boldsymbol{b}}, \widehat{\boldsymbol{z}})$ corresponding to a random challenge $\widehat{\boldsymbol{a}}$. The only solid chance that $\mathcal{A}$ can pass the authentication is that $\widehat{\boldsymbol{a}}$ appears as at least one of q challenges in Phase 1. This event happens with negligible probability

$$1 - \left(1 - \frac{1}{2^{m-1} - 1}\right)^q \approx 1 - e^{-\frac{q}{2^{m-1}-1}} \quad .$$

Otherwise, if $\widehat{\boldsymbol{a}}$ is different from all $\boldsymbol{a}_i$'s, as we have proven in Cases Three and Four of Theorem 2.1, $\mathcal{A}$ only can pass the authentication with probability P_{FP}.

Therefore, no PPT adversary can achieve non-negligible advantage, and the LCMQ protocol is secure in the DET-model, assuming the hardness of the LPN problem. □

Security in an Adaptive-DET-model

The LCMQ protocol can be proven secure in an enhanced DET-model—*adaptive-DET-model* in which Phase 1 is identical to that in the DET-model, but in Phase 2 the adversary $\mathcal{A}$ is equipped with more capacity: after receiving the challenge $\widehat{\boldsymbol{a}}$ and before outputting the response, $\mathcal{A}$ is permitted to query $\mathcal{T}_{\boldsymbol{k}_1,\boldsymbol{k}_2,\eta,n}$ q_2 times, only with one obvious restriction that $\mathcal{A}$ cannot use $\widehat{\boldsymbol{a}}$ as a challenge. The MIM-model can be extended to an adaptive-MIM-model in the same way. By similar security proofs , the LCMQ protocol is still secure in the both adaptive models.

The relation between adaptive-DET-model and DET-model is analogue to that between adaptive chosen-ciphertext attack (CCA2) and chosen-ciphertext attack (CCA) [26] for public-key encryption schemes. As CCA2 is generally more preferred than CCA, the security guarantee of a protocol in the adaptive-DET-model is more desired in realistic applications. As a matter of fact, those previous HB-like authentication protocols, due to the fact that all of them are vulnerable to the OOV attack, are not secure in the adaptive-DET-model, while they are (provably or presumably) secure in the DET-model.

2.4 Protocol Parameters Selections and Discussions

2.4.1 Hardness of LCMQ Instances in the DET-model

According to the LCMQ security proof in the DET-model, $m \geq d + 1$ will suffice to provide d-bit security (Adding one bit in the key length is due to the fact that the parities of $\mathsf{Hwt}(\boldsymbol{k}_1)$ and $\mathsf{Hwt}(\boldsymbol{k}_2)$ are known to the public.) and the precise value of noise level η seems insignificant as long as there are noises. Of course, this is only because we use LPN

instances as a random oracle in the proof of Theorem 2.2. Essentially, the fundamental problem that an adversary confronts in the DET-model is described below.

Definition 2.7 (LCMQ problem)**.** *Let m be a P2 number, $n < m$, $\eta \in (0, \frac{1}{2})$ be a noise level, $\boldsymbol{k}_1 \xleftarrow{\$} \mathbb{S}_m$ and the parity of $\boldsymbol{k}_1$'s Hamming weight is public, $\boldsymbol{k}_2 \xleftarrow{\$} \mathbb{S}_m^{\mathrm{e}}$. Given q pairs $\langle \boldsymbol{b}_i, \boldsymbol{z}_i = (((\boldsymbol{b}_i \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{v}_i)||\boldsymbol{r}_i) \circ \mathbf{C}_{\boldsymbol{k}_2}^{[(m\text{-}1)\times m]} \rangle$, for $i = 0, 1, \cdots, q-1$, where $\boldsymbol{b}_i \xleftarrow{\$} \mathbb{S}_m$, $\boldsymbol{v}_i \xleftarrow{\$} \mathtt{Bio}_{n,\eta}$, and $\boldsymbol{r}_i \xleftarrow{\$} \{0,1\}^{m-n-1}$, recover $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$.*

If there is no noise in the LCMQ problem and suppose $n = m - 1$, the instances degenerate to

$$\langle \boldsymbol{b}_i \xleftarrow{\$} \mathbb{S}_m, \boldsymbol{z}_i' = \boldsymbol{b}_i \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times(m\text{-}1)]} \circ \mathbf{C}_{\boldsymbol{k}_2}^{[(m\text{-}1)\times m]} \rangle \ . \tag{2.8}$$

Let $\mathbf{C}_{\boldsymbol{\kappa}} = \mathbf{C}_{\boldsymbol{k}_1} \circ \mathbf{C}_{\boldsymbol{k}_2}$, $\boldsymbol{k}_1'$ be the reverse vector of $\boldsymbol{k}_1$, $\boldsymbol{k}_2'$ be the vector that is left-rotated one element of $\boldsymbol{k}_2$. Then

$$\mathbf{C}_{\boldsymbol{k}_1}^{[m\times(m\text{-}1)]} \circ \mathbf{C}_{\boldsymbol{k}_2}^{[(m\text{-}1)\times m]} = \mathbf{C}_{\boldsymbol{\kappa}} - \boldsymbol{k}_1'^T \circ \boldsymbol{k}_2'$$

Since secret $\boldsymbol{k}_1 \xleftarrow{\$} \mathbb{S}_m$, secret $\boldsymbol{k}_2 \xleftarrow{\$} \mathbb{S}_m^{\mathrm{e}}$, and the parity of $\boldsymbol{k}_1$'s Hamming weight is public, the noise-free LCMQ instances of (2.8) effectively constitute a multivariate quadratic system in $2(m-1)$ variants. We define such an MQ system as *an $(\boldsymbol{k}_1, \boldsymbol{k}_2)$ circulant-P2-matrices generated MQ system.*

Example: For $m = 3$, an $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ circulant-P2-matrices generated MQ system is

$$\boldsymbol{b}_i \circ \begin{bmatrix} \alpha_0\beta_0 + \alpha_1\beta_2 & \alpha_0\beta_1 + \alpha_1\beta_0 & \alpha_0\beta_2 + \alpha_1\beta_1 \\ \alpha_2\beta_0 + \alpha_0\beta_2 & \alpha_2\beta_1 + \alpha_0\beta_0 & \alpha_2\beta_2 + \alpha_0\beta_1 \\ \alpha_1\beta_0 + \alpha_2\beta_2 & \alpha_1\beta_1 + \alpha_2\beta_0 & \alpha_1\beta_2 + \alpha_2\beta_1 \end{bmatrix} = \boldsymbol{z}_i' \ ,$$

where $\boldsymbol{b}_i \xleftarrow{\$} \mathbb{S}_m$.

Those noise-free LCMQ instances are related to another hard problem—the multivariate quadratic (MQ) problem [69].

Definition 2.8 (MQ Problem)**.** *Given a system of w quadratic equations in s variables over a finite field, find a valid solutions satisfying all equations.*

Generally speaking, the hardness of MQ problem depends on the relative values of w and s. When $w = 1$, it is a trivial case and a solution can be readily retrieved. If w is significantly smaller than s, as an underdefined system, finding a solution is fairly easy [50]. When w is much greater than t, as an overdefined system, the MQ Problem becomes easy too. Specifically, if there are $\frac{t(t+1)}{2} + 1$ (for $GF(2)$) or $\frac{t(t+3)}{2} + 1$ (for all other finite fields)

linearly independent equations available, the MQ problem can be solved by linearization of running time $O(t^6)$ [24]. For general values of w and s, the MQ problem is known to be NP-hard, even for quadratic equations oven $GF(2)$ [69, 65]. This problem has been used as the security foundation of cryptographic algorithms, such as the UOV [96] and Sflash [9] signature schemes, and the QUAD [24] stream cipher.

It is clear that the LCMQ problem is essentially a circulant-P2-matrices generated MQ system with noise. The name of LCMQ is exactly derived from the combination of LPN, CM, and MQ. Because of the nice linear independence property of circulant-P2 matrices, solving this noise-free MQ system generally needs $(m-1)^2$ equations. On the other hand, in Section 2.1.1, we have learned that a little noise ingredient turns a simple solving-linear-equation task into an NP-hard problem, for which only sub-exponential algorithms are discovered. Since breaking the LCMQ problem requires a chain of solving the LPN and MQ problems, its computational complexity is equal to the product of the complexity of those two hard problems. In fact, the LCMQ problem of parameter m, with a same noise level, should be harder than the LPN problem of parameter $(m-1)^2$ because noise $\boldsymbol{v}_i$ in the LCMQ problem is encrypted and then is expanded all over $\boldsymbol{z}_i$. Therefore, we are highly confident that there is no sub-exponential algorithm solving the LCMQ problem, and an adversary only can rely on exhaustive search (matching in the middle) to recover the two m-bit keys.

2.4.2 Parameters Recommendation and Comparisons

As we argued above, $m = d+1$ with small noise level η is sufficient for d-bit security in the DET-model. On the other hand, the security proof in the MIM-model demands negligible false rates, ruling out too small choices of m. To be concrete, those false rates should not exceed 2^{-d}. In practice, the LCMQ protocol may use the upper-bounded Bernoulli noise mode, which eliminates the false negative. Although the HB-like protocols with this noise mode are vulnerable to the OOV2 attack (see Section 2.1.2), it can be safely used in the LCMQ protocol, because we have proven the LCMQ protocol's security against this kind of man-in-the-middle attack. Recall the false positive rate $P_{\mathrm{FP}} = \sum_{i=0}^{\tau} \binom{n}{i}$, thus n should always be $m-1$ in practice. Based on all those arguments, we recommend the following protocol parameters for different security levels in Table 2.2.

Using those parameter selections, we compare the LCMQ protocol key size with traditional cryptographic primitives in Table 2.3. The key sizes of RSA and DSA are quoted from *NIST Special Publication 800-57: Recommendation for Key Management, 2007.*

As we we see in Table 2.2, $(m = 163, \eta = 0.08)$ can provide 80-bit security in the LCMQ protocol. In contrast, the LF algorithm [100], as the best algorithm to solving LPN instances by far, renders the HB-like protocols to take $m \geq 512$ with noise level

Table 2.2: Parameters recommendation for the LCMQ protocol with the upper-bounded Bernoulli noise mode

Security Level	80	112	128
η	0.08	0.09	0.10
m	163	269	317
n	162	268	316
τ	18	34	45

Table 2.3: Comparison of the LCMQ protocol with traditional cryptographic primitives

Security Level	80	112	128
Symmetric Key Encryption	Skipjack	3-DES	AES-128
Hash Function	SHA-1	SHA-256	SHA-256
RSA Key Size	1,024	2,048	3,072
DSA Key Size	1,024	2,048	3,072
LCMQ Key Size	326	538	634

$\eta = 0.25$. Therefore, the LCMQ protocol outperforms all HB-like protocols in terms of metrics of storage, tag's computation, communication, and implementation while provably preventing all PPT attacks. All of those make it very tempting as a lightweight, reliable, secure entity authentication for low-cost devices.

2.4.3 Discussions

Two-as-One Variation

If the performance is ridiculously vital for some applications and thus a smaller value of m is desired while the security level is allowed to slightly sacrifice, a variation of LCMQ protocol, by combining two paralleled authentications as one, might be of help. In this variation, the reader sends two vectors $(\boldsymbol{a}_1, \boldsymbol{a}_2)$ as challenge, and then tag responds with $(\boldsymbol{b}_1, \boldsymbol{z}_1, \boldsymbol{b}_2, \boldsymbol{z}_2)$. Consequently, the reader verifies the authentication by checking $\mathsf{Hwt}((\boldsymbol{b}_1 \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{y}_1) + \mathsf{Hwt}((\boldsymbol{b}_2 \circ \mathbf{C}_{\boldsymbol{k}_1}^{[m\times n]}) \oplus \boldsymbol{y}_2) \leq \tau$. A parameter set as low as $(m = 83, n = 82, \eta = 0.08, \tau = 19)$ with the upper-bounded Bernoulli noise mode can be used. This variation does not affect the security proofs in the DET-model, but the security proof in the MIM-model encounters issues. Even thought it is true that $P_{FP} \leq 2^{-80}$ for that parameter set, ϵ in Theorem 2.1 is no longer equal to $\max(P_{\mathrm{FN}}, P_{\mathrm{FP}})$, but notably bigger than it, as an adversary may manipulate only one set of $(\boldsymbol{a}_1, \boldsymbol{b}_1, \boldsymbol{z}_1)$ and $(\boldsymbol{a}_2, \boldsymbol{b}_2, \boldsymbol{z}_2)$. Nevertheless, the variation may be attractive for extremely resource-constrained systems, and we recommend

$(m = 107, n = 106, \eta = 0.1, \tau = 30)$ for 80-bit security.

Different Lengths of $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$

Technically, the two keys $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ can have different lengths, say m_1 and m_2 bits respectively, as long as m_1 and m_2 are P2 numbers and $n < \min(m_1, m_2)$. However, from the perspective of implementation, it is more convenient to make them same so that only one dimensional matrix multiplication is required rather than two.

Embedding ID into Challenge

From the security proofs of the LCMQ protocol, we can see that challenge $\boldsymbol{a}$ does not need to be random; instead, being unique would suffice for $\boldsymbol{a}$. Thus a reader can use a nonce (number used once) as a challenge. In addition, the identity of the tag to be verified can be implicitly embedded as part of a challenge vector, which is very useful in practice.

2.5 Conclusions

In this chapter, we have presented a lightweight, efficient, practical, and secure entity authentication protocol for RFID systems. Built upon the learning parity with noise problem, a special type of circulant matrix named circulant-P2 matrix, and the multivariate quadratic problem, the proposed LCMQ protocol outweighs all HB-like protocols in terms of the provable security in a general man-in-the-middle model and the tag's computation, storage, and communication costs. As a technique core of this proposal, the vector linear independence, gentle properties, and efficient algorithms on matrix operation of circulant-P2 matrix may also be used to construct other cryptographic primitives and secure protocols.

Chapter 3

Mutual Authentication in Wireless Sensor Networks

The previously proposed LCMQ protocol is greatly suitable for low-cost RFID tags and sensor nodes to authenticate their identities in front of ordinary devices. However, if tags and nodes need to verify others entity authentication, the requirement of performing extended Euclidean algorithm, as mild as it is, may not be fulfilled by some extremely computationally-constrained devices. Generally, this is not an issue for RFID tags, since they are mainly for one-way authentication. In contrast, mutual authentication is a common security obligation in wireless sensor networks; thus entity authentication schemes that merely rest on bit-operations will be of help. This chapter embraces this challenge.

This work is still based on the learning parity with noise problem and circulant-P2 matrices, but adopts the HB-like authentication structure to preserve the bit-operation-only property with necessary enhancements. In order to gradually and clearly demonstrate our proposals and for the purpose of fair comparisons, we first in Section 3.1 present an HB-like protocol named $\text{HB}^{\mathcal{C}}$, which employs circulant-P2 matrices in the three-pass blinding/challenge/response authentication structure to reduce key storage. However, all HB-like protocols, including $\text{HB}^{\mathcal{C}}$, suffer from the OOV attack. In Section 3.2 we provide two mechanisms to protect the protocols from the attack and also improve protocol performance. Applying those techniques to an HB-like one-way protocol results in the enhanced version of authentication, and the parameters selections are stated and compared among enhanced HB-like protocols. Afterwards, in Section 3.3, we propose a mutual authentication framework, which may be instantiated by circulant-P2 matrices as the $\text{HB}^{\mathcal{C}}$-M mutual authentication scheme. Its application scenarios in sensor networks are discussed in Section 3.4. Section 3.5 draws a conclusion of this chapter. Partial contents of this work have been published in [101, 106].

In this chapter, for one-way authentication, we remain to use RFID tag and reader

Tag ($\mathbf{K}$)		Reader ($\mathbf{K}$)
$\boldsymbol{b} \in_R \{0,1\}^{m_b}$	$\xrightarrow{\boldsymbol{b}}$	
	$\xleftarrow{\boldsymbol{a}}$	$\boldsymbol{a} \in_R \{0,1\}^{m_a}$
$\boldsymbol{v} \in_R \{\{0,1\}^n \mid \mathsf{Pr}(\boldsymbol{v}[i]=1)=\eta,$ where $0 \le i \le n-1\}$;		
$\boldsymbol{y} = ((\boldsymbol{a}\|\boldsymbol{b}) \circ \mathbf{K}) \oplus \boldsymbol{v}$	$\xrightarrow{\boldsymbol{y}}$	
		Accept iff $\mathsf{Hwt}(((\boldsymbol{a}\|\boldsymbol{b}) \circ \mathbf{K}) \oplus \boldsymbol{y}) \le \tau$

Figure 3.1: Revised HB$^{\#}$ authentication protocol, where $\mathbf{K}$ is an $(m \times n)$ Toeplitz matrix, $m_a + m_b = m$, noise level $\eta \in (0, \frac{1}{2})$, pass-threshold $\tau \in (\eta n, \frac{n}{2})$

as the two authentication participants. When we move to mutual authentication, we will name the two parties as Alice and Bob, because they are essentially equal roles in our proposed mutual authentication.

3.1 HB$^{\mathcal{C}}$ One-Way Authentication Protocol

Among previous HB-like protocols, the HB$^{\#}$ Protocol is the most efficient one in terms of key storage and communication cost. However, there are still some practical and provably secure concerns about HB$^{\#}$, which motivate us to propose the HB$^{\mathcal{C}}$ protocol. First, we take a brief review on HB$^{\#}$.

3.1.1 Review of HB$^{\#}$ Protocol

The authentication key in the HB$^{\#}$ protocol [76] consists of two independent Toeplitz matrices, one $(m_a \times n)$, the other $(m_b \times n)$; thus the total memory cost is $m + 2n - 2$ bits, where $m = m_a + m_b$. We notice that it can be improved as the revised HB$^{\#}$ protocol depicted in Figure 3.1 by changing to one $(m \times n)$ Toeplitz matrix. The original HB$^{\#}$ security proof in [76] still holds for the revised HB$^{\#}$. Since it reduces the memory cost to $m + n - 1$ bits while maintaining all other performance (and subject to all constraints), the revised HB$^{\#}$ protocol is preferred to the original HB$^{\#}$ protocol.

Gilbert, Robshaw, and Seurin [76] introduced a GRS-MIM-model, in which the MIM adversary is only allowed to manipulate the challenges from the reader to the tag, to prove that Random-HB$^{\#}$ and HB$^{\#}$ [76] prevent the GRS attack. Despite the elegant security reduction proofs in [76], the GRS-MIM-model does not completely simulate the full GRS attack. It is easy to see that the GRS attack strategy applies to manipulating blinding

vectors for recovering $\boldsymbol{k}_b$ in the HB$^+$ protocol as well. With $\boldsymbol{k}_b$, an active adversary can easily retrieve $\boldsymbol{k}_a$ by querying a tag with identical challenges. Therefore, in a full GRS attack model, the adversary should be allowed to manipulate both blinding vectors and challenge vectors. Unfortunately, the Random-HB$^\#$ and HB$^\#$ protocols security proofs cannot be extended to such a model, because the condition necessary for Theorems 2 and 3 in [76] to hold cannot be satisfied for practical parameters. Even though there is no realistic ground to doubt whether Random-HB$^\#$ and HB$^\#$ are vulnerable to the full GRS attack, a protocol is still more desirable that can provably resist the full GRS attack. In addition, there is a practical issue about the HB$^\#$'s key storage cost. Even for the revised HB$^\#$ protocol, a typical authentication key incorporates above one thousand bits, which is comparable to asymmetric primitive key size, and may not be suitable for low-cost devices.

Intended to overcome these two deficiencies of HB$^\#$, we propose our HB-like approach by replacing Toeplitz matrix in HB$^\#$ with circulant-P2 matrix. The resulting HB$^{\mathcal{C}}$ protocol is provably resistant to the full GRS attack except one trivial case, and consumes roughly half of key storage as the revised HB$^\#$ protocol, without degrading other protocol performance.

3.1.2 HB$^{\mathcal{C}}$ Protocol Description

The HB$^{\mathcal{C}}$ protocol with the Bernoulli noise mode is illustrated in Figure 3.2. In contrast to the parallel HB$^+$ protocol employing matrix challenges and vector key and the HB$^\#$ protocol requiring vector challenge and Toeplitz matrix key, the HB$^{\mathcal{C}}$ protocol makes use of circulant-P2 matrix challenge and vector key, or equivalently vector challenge and circulant-P2 matrix key. Benefited from Lemma 2.3, the HB$^{\mathcal{C}}$ protocol shows advantages over the other two in terms of provable security against the full GRS attack and key storage consumption.

3.1.3 Security Models Definitions

Similarly, for the sake of security formalization, we denote a tag-reader HB$^{\mathcal{C}}$ authentication system by a pair of probabilistic functions $(\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}, \mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau})$, and two models are defined below.

Definition 3.1 (DET-model)**.** *In the DET-model, which is essentially equivalent to the DET-model in the previous chapter, adversary $\mathcal{A}$ attacks the protocol in two phases:*

- Phase 1*: Adversary $\mathcal{A}$ interacts q times with $\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}$. On the ith invocation, $\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}$ internally generates a random blinding vector $\boldsymbol{b}_i$, takes a challenge vector $\boldsymbol{a}_i$ from $\mathcal{A}$ as input, and outputs $\boldsymbol{y}_i = (\mathbf{C}^{[n\times m]}_{\boldsymbol{a}_i\|\boldsymbol{b}_i} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}_i$ to $\mathcal{A}$. This simulates an active adversary querying the tag.*

Tag ($\boldsymbol{k}$)		Reader ($\boldsymbol{k}$)
$\boldsymbol{b} \in_R \{0,1\}^{m_b}$	$\xrightarrow{\boldsymbol{b}}$	
	$\xleftarrow{\boldsymbol{a}}$	$\boldsymbol{a} \in_R \{0,1\}^{m_a}$
$\boldsymbol{v} \in_R \{\{0,1\}^n \mid \mathsf{Pr}(\boldsymbol{v}[i]=1)=\eta,$ where $0 \le i \le n-1\}$;		
$\boldsymbol{y} = (\mathbf{C}^{[n\times m]}_{\boldsymbol{a}\|\boldsymbol{b}} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}$	$\xrightarrow{\boldsymbol{y}}$	
		Accept iff $\mathsf{Hwt}((\mathbf{C}^{[n\times m]}_{\boldsymbol{a}\|\boldsymbol{b}} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{y}) \le \tau$

Figure 3.2: $\mathrm{HB}^{\mathcal{C}}$ one-way authentication protocol, where $\boldsymbol{k} \in \mathbb{S}_m$, m is a P2 number, $m_a + m_b = m$, n is the interaction expansion and $n < m$, noise level $\eta \in (0, \frac{1}{2})$, integer pass-threshold $\tau \in (\eta n, \frac{n}{2})$

- Phase 2*: Adversary $\mathcal{A}$ interacts with $\mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}$, intending to impersonate the tag, achieving advantage*

$$\mathsf{Adv}^{\mathrm{DET}}_{\mathcal{A},\mathrm{C}}(m,\eta,n,\tau) \stackrel{def}{=} \mathsf{Pr}[\boldsymbol{k} \xleftarrow{\$} \mathbb{S}_m, \mathcal{A}^{\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}}(1^m) : \langle \mathcal{A}, \mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}\rangle = \mathrm{ACCEPT}] - P_{\mathrm{FP}} \ .$$

Definition 3.2 (GRS-model)**.** *In the GRS-model, the GRS attack is carried out in two phases:*

- Phase 1*: Adversary $\mathcal{A}$ manipulates any blinding vectors from $\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}$ to $\mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}$ and any challenge vectors from $\mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}$ to $\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}$ for q executions. on the ith invocation, $\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}$ internally generates a random blinding vector $\boldsymbol{b}_i$, and sends it to adversary $\mathcal{A}$. Then $\mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}$ receives a modified blinding vector $\boldsymbol{b}'_i$ from $\mathcal{A}$, generates a random challenge vector $\boldsymbol{a}_i$, and gives it to $\mathcal{A}$. $\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}$ takes a modified challenge vector $\boldsymbol{a}'_i$ from $\mathcal{A}$, and sends $\boldsymbol{y}_i = (\mathbf{C}^{[n\times m]}_{\boldsymbol{a}'_i\|\boldsymbol{b}_i} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}_i$ to $\mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}$, which then checks if $\mathsf{Hwt}((\mathbf{C}^{[n\times m]}_{\boldsymbol{a}_i\|\boldsymbol{b}'_i} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{y}_i) \le \tau$. If it holds, $\mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}$ outputs "ACCEPT" to $\mathcal{A}$; otherwise, it outputs "REJECT". This simulates a full GRS attacker. We define interference vector $\boldsymbol{\psi}_i$ by $(\boldsymbol{a}_i \oplus \boldsymbol{a}'_i)\|(\boldsymbol{b}_i \oplus \boldsymbol{b}'_i)$. To simplify the security proof for $\mathrm{HB}^{\mathcal{C}}$, we rule out a trivial case of $\boldsymbol{\psi}_i = \mathbf{1}_m$.*

- Phase 2*: Adversary $\mathcal{A}$ interacts with $\mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}$, intending to impersonate the tag, achieving advantage*

$$\mathsf{Adv}^{\mathrm{GRS}}_{\mathcal{A},\mathrm{C}}(m,\eta,n,\tau) \stackrel{def}{=} \mathsf{Pr}[\boldsymbol{k} \xleftarrow{\$} \mathbb{S}_m, \mathcal{A}^{\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n},\mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}}(1^m) : \langle \mathcal{A}, \mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}\rangle = \mathrm{ACCEPT}] - P_{\mathrm{FP}} \ .$$

In the GRS-MIM-model defined in [76], the adversary is only permitted to change the challenges from the reader to the tag; so the GRS-model includes the GRS-MIM-model,

and actually simulates the full GRS attack except the trivial case $\boldsymbol{\psi}_i = \mathbf{1}_m$.

In the GRS-model, since

$$(\mathbf{C}_{\boldsymbol{a}_i\|\boldsymbol{b}'_i}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{y}'_i = ((\mathbf{C}_{\boldsymbol{a}_i\|\boldsymbol{b}'_i}^{[n\times m]} \oplus \mathbf{C}_{\boldsymbol{a}'_i\|\boldsymbol{b}_i}^{[n\times m]}) \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}_i = (\mathbf{C}_{\boldsymbol{\psi}_i}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}_i \ ,$$

the authentication result is equivalently decided by

$$\mathsf{Hwt}((\mathbf{C}_{\boldsymbol{\psi}_i}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}_i) \leq \tau \ . \tag{3.1}$$

Security Intuitions

If a GRS adversary imposes an inference vector $\boldsymbol{\psi}_i \in \mathbb{S}_m$ on the HB$^{\mathcal{C}}$ protocol without knowledge of $\boldsymbol{k}$, the resulting error vector $(\mathbf{C}_{\boldsymbol{\psi}_i}^{[n\times m]} \circ \boldsymbol{k}^T)^T$ will be random as all row vectors in $\mathbf{C}_{\boldsymbol{\psi}_i}^{[n\times m]}$ are linearly independent by Lemma 2.3; thus the manipulated authentication will fail with an overwhelming probability, revealing no useful information—the protocol is resistant to the GRS attack. The formal reduction proof is given as follows.

3.1.4 HB$^{\mathcal{C}}$ Security in the GRS-model

Theorem 3.1 (HB$^{\mathcal{C}}$ Security in the GRS-model)**.** *Let $\mathcal{A}$ be an adversary attacking the HB$^{\mathcal{C}}$ protocol in the GRS-model, modifying at most q executions of the protocol between an honest tag and an honest reader, running in time t, and achieving $\mathsf{Adv}^{\mathrm{GRS}}_{\mathcal{A},\mathrm{C}}(m,\eta,n,\tau) \geq \delta$. Then there exists an adversary $\mathcal{A}'$ attacking the HB$^{\mathcal{C}}$ protocol in the DET-model, interacting at most q oracle queries, running in time $O(t)$, and achieving $\mathsf{Adv}^{\mathrm{DET}}_{\mathcal{A}',\mathrm{C}}(m,\eta,n,\tau) \geq \delta - q\epsilon(P_{\mathrm{FP}} + \delta)$ for some negligible function ϵ. Hence, assuming HB$^{\mathcal{C}}$ is secure in the DET-model, HB$^{\mathcal{C}}$ is provably secure in the GRS-model.*

Proof. In Phase 1, $\mathcal{A}'$ can readily simulate thetag to $\mathcal{A}$ since $\mathcal{A}'$ has access to $\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}$. The main challenge lies on how to simulate $\mathcal{R}^{\mathrm{C}}_{\boldsymbol{k},n,\tau}$. Similar to the proof method for the Random-HB$^{\#}$ protocol [76], $\mathcal{A}'$ launches Phase 1 of adversary $\mathcal{A}$, and simulates the tag and the reader for q times as follows:

1. $\mathcal{A}'$ obtains a blinding vector $\boldsymbol{b}_i$ from the tag $\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}$, and sends $\boldsymbol{b}_i$ as the blinding vector of the simulated tag; $\mathcal{A}$ modifies it into $\boldsymbol{b}'_i$, and sends $\boldsymbol{b}'_i$ to the simulated reader.
2. $\mathcal{A}'$ sends a random vector $\boldsymbol{a}_i$ as the challenge of the simulated reader. $\mathcal{A}$ modifies it into $\boldsymbol{a}'_i$; $\mathcal{A}'$ forwards $\boldsymbol{a}'_i$ to the real tag.
3. The real tag responds with $\boldsymbol{y}_i = (\mathbf{C}_{\boldsymbol{a}'_i\|\boldsymbol{b}_i}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}_i$ to $\mathcal{A}'$, which uses it as the answer of the simulated tag to the simulated reader.

4. Recall $\boldsymbol{\psi}_i = (\boldsymbol{a}_i \oplus \boldsymbol{a}'_i)\|(\boldsymbol{b}_i \oplus \boldsymbol{b}'_i)$. If $\boldsymbol{\psi}_i = \mathbf{0}_m$, $\mathcal{A}'$ outputs "ACCEPT" to $\mathcal{A}$ as the authentication result of the simulated reader; otherwise, it outputs "REJECT".

After Phase 1, $\mathcal{A}'$ launches Phase 2 of $\mathcal{A}$. Since Phase 2 in the DET-model is identical to that in the GRS-model, $\mathcal{A}'$ just replicates $\mathcal{A}$'s behavior with the real reader, perfectly simulating the tag $\mathcal{T}^{\mathrm{C}}_{\boldsymbol{k},\eta,n}$. Therefore, if $\mathcal{A}$ achieves $\mathsf{Adv}^{\mathrm{GRS}}_{\mathcal{A},\mathrm{C}}(m,n,\eta,u) \geq \delta$, then the probability of $\mathcal{A}'$ successfully impersonating a valid tag is the same as the success probability of $\mathcal{A}$, i.e., $P_{\mathrm{FP}} + \delta$, on the condition that the reader is correctly simulated by $\mathcal{A}'$ in Phase 1.

Now we need to calculate the probability of $\mathcal{A}'$ successfully simulating the reader for $\mathcal{A}$ in Phase 1. Consider one execution of the protocol in Phase 1, based on Equation (3.1).

- When $\boldsymbol{\psi}_i = \mathbf{0}_m$, $\mathcal{A}'$ fails at simulating the reader with a probability equal to the false negative rate P_{FN}.
- If $\boldsymbol{\psi}_i \neq \mathbf{0}_m$, since we suppose $\boldsymbol{\psi}_i \neq \mathbf{1}_m$, thus $\boldsymbol{\psi}_i \in \mathbb{S}_k$. By Lemma 2.3, all row vectors in $\mathbf{C}^{[n\times m]}_{\boldsymbol{\psi}_i}$ are linearly independent. Let $\boldsymbol{\gamma}_i = \mathbf{C}^{[n\times m]}_{\boldsymbol{\psi}_i} \circ \boldsymbol{k}$ denote the error vector added by $\mathcal{A}$. Following the same argument in Theorem 2 of [93], $\boldsymbol{\gamma}_i$ is uniformly distributed over $\{0,1\}^n$, as the row vectors of $\mathbf{C}^{[n\times m]}_{\boldsymbol{\psi}_i}$ are linearly independent. Since the adversary does not extra knowledge about the Bernoulli noises introduced by the tag, $\boldsymbol{v}_i$ is independent of $\boldsymbol{\gamma}_i$. Thus the resulting error vector $\boldsymbol{\gamma}_i \oplus \boldsymbol{v}_i$ follows the uniform distribution over $\{0,1\}^n$. As a result, the probability of $\mathcal{A}'$ wrongly outputting "REJECT" is exactly the same as the false positive rate P_{FP}.

Overall, $\mathcal{A}'$ fails at simulating the reader in one execution at most with probability $\epsilon = \max(P_{\mathrm{FN}}, P_{\mathrm{FP}})$. The probability of $\mathcal{A}'$ correctly simulating the reader in Phase 1 would be not less than $1 - q\epsilon$, and adversary $\mathcal{A}'$ can impersonate a valid tag with success probability not less than $(P_{\mathrm{FP}} + \delta)(1 - q\epsilon)$.

Therefore, $\mathcal{A}'$ can achieve advantage

$$\mathsf{Adv}^{\mathrm{DET}}_{\mathcal{A}',\mathrm{C}}(m,\eta,n,\tau) \geq (P_{\mathrm{FP}} + \delta)(1 - q\epsilon) - P_{\mathrm{FP}} = \delta - q\epsilon(P_{\mathrm{FP}} + \delta) \ .$$

If δ is non-negligible, then $q\epsilon(P_{\mathrm{FP}} + \delta) \leq \frac{\delta}{2}$ for m big enough, and $\mathsf{Adv}^{\mathrm{DET}}_{\mathcal{A}',\mathrm{C}}(m,\eta,n,\tau) \geq \frac{\delta}{2}$ is non-negligible. Thus if $\mathrm{HB}^{\mathcal{C}}$ is secure in the DET-model, $\mathrm{HB}^{\mathcal{C}}$ is secure in the GRS-model. □

3.1.5 $\mathrm{HB}^{\mathcal{C}}$ Security in the DET-model

Similar to $\mathrm{HB}^{\#}$, we cannot present a strict reduction from the LPN problem to $\mathrm{HB}^{\mathcal{C}}$ security in the DET-model currently. Instead, we conjecture that the $\mathrm{HB}^{\mathcal{C}}$ protocol is secure in the DET-model.

Claim 3.1. *In the DET-model, the HB$^{\mathcal{C}}$ protocol is as secure as the parallel HB^{+} protocol.*

Justification. Let's recall the parallel HB^{+} protocol, which is provably secure in the DET-model [91, 93]. The tag first generates a random $(n \times m_b)$ blinding matrix $\mathbf{B}$ and sends it to the reader; then the reader selects an $(n \times m_a)$ challenge matrix $\mathbf{A}$ at random. Let $\mathbf{A}\|\mathbf{B}$ stand for concatenation of matrices $\mathbf{A}$ and $\mathbf{B}$ (with the same number of rows), that is, each row in the resulting matrix is the concatenation of two corresponding rows in $\mathbf{A}$ and $\mathbf{B}$. After receiving $\mathbf{A}$, the tag computes and sends the n-bit response vector $\boldsymbol{y} = ((\mathbf{A}\|\mathbf{B}) \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}$. Let $\boldsymbol{k}_a\|\boldsymbol{k}_b = \boldsymbol{k}$, where vector $\boldsymbol{k}_a$ is of m_a bits and vector $\boldsymbol{k}_b$ is of m_b bits. Then the response vector is equivalently computed by

$$\boldsymbol{y} = (\mathbf{A} \circ \boldsymbol{k}_a^T)^T \oplus (\mathbf{B} \circ \boldsymbol{k}_b^T)^T \oplus \boldsymbol{v} \ . \tag{3.2}$$

For the HB$^{\mathcal{C}}$ protocol, we define $\bar{\boldsymbol{a}} = \boldsymbol{a}\|\mathbf{0}_{m_b}$, and $\bar{\boldsymbol{b}} = \mathbf{0}_{m_a}\|\boldsymbol{b}$. Then the response vector is equivalently computed by

$$\boldsymbol{y} = (\mathbf{C}_{\bar{\boldsymbol{a}}}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus (\mathbf{C}_{\bar{\boldsymbol{b}}}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v} \ . \tag{3.3}$$

In Theorem 2 of [93], to prove the parallel HB^{+} protocol's security in the DET-model, it requires 2^{n-m_a} to be negligible such that the row vectors in the random challenge matrix $\mathbf{A}$ are linearly independent. Interestingly, the security proof of Random-HB$^{\#}$ in the GRS-MIM-model (Theorem 2 of [76]) demands 2^{m_a-n} to be negligible. In contrast, the HB$^{\mathcal{C}}$ protocol does not impose such limitations. The row vectors in circulant-P2 matrix $\mathbf{C}_{\bar{\boldsymbol{a}}}^{[n\times m]}$ are linearly independent with the overwhelming probability of $1-(\frac{1}{2})^{m_a}$, because $\bar{\boldsymbol{a}} \in \mathbb{S}_m$ if $\boldsymbol{a} \neq \mathbf{0}_{m_a}$. Similarly, the row vectors in circulant-P2 matrix $\mathbf{C}_{\bar{\boldsymbol{b}}}^{[n\times m]}$ are linearly independent with probability $1-(\frac{1}{2})^{m_b}$.

The HB$^{\mathcal{C}}$ and parallel HB^{+} protocols are resembling except that matrix $(\mathbf{A}\|\mathbf{B})$ in parallel HB^{+} is random and $\mathbf{C}_{\boldsymbol{a}\|\boldsymbol{b}}^{[n\times m]}$ in HB$^{\mathcal{C}}$ is a circulant-P2 matrix. Since the major operation in the authentication is inner product, the linear independence of the row vectors in $\mathbf{C}_{\boldsymbol{a}\|\boldsymbol{b}}^{[n\times m]}$ actually enhances the protocol's security.

Therefore, we claim that HB$^{\mathcal{C}}$ is as secure as the parallel HB^{+} protocol in the DET-model. □

Storage Benefit

Since HB$^{\#}$ employs Toeplitz matrix, compared to m-bit key in HB$^{\mathcal{C}}$ protocol, the keys in HB$^{\#}$ and revised HB$^{\#}$ consume $m+2n-2$ bits and $m+n-1$ bits respectively. As we derive later, to achieve negligible false rate, n is usually several hundred and is at the same

order of m, thus the HB$^{\mathcal{C}}$ protocol significantly reduces the storage requirement roughly by half.

3.2 Techniques to Enhance HB-like Protocols

Now we have three HB-like protocols that resist the GRS attack: Random-HB$^{\#}$, HB$^{\#}$, and HB$^{\mathcal{C}}$ which employ random, Toeplitz, and circulant-P2 key matrices respectively. Unfortunately, those protocols are still subject to the OOV attack. We here provide two techniques to protect HB-like protocols against the attack, and also improve protocol performance in terms of false rates and key length. These techniques are demonstrated through the HB$^{\mathcal{C}}$ protocol later, but we stress that they can apply to any HB-like protocol.

3.2.1 New Noise Mode to Prevent OOV Attack

We propose a new noise mode for all HB-like protocols to thwart the OOV attack. A protocol with this noise mode will naturally be false-negative-free, while robust to the OOV2 attack. Let $t = \lfloor \eta n \rfloor$, for this noise mode, the tag generates a noise $\boldsymbol{v}$ such that $\mathsf{Hwt}(\boldsymbol{v})$ is equal to t or $t+1$; the reader accepts the tag's response $\boldsymbol{y}$ if and only if $\mathsf{Hwt}((\mathbf{C}^{[n\times m]}_{\boldsymbol{a}\|\boldsymbol{b}} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{y}) = t$ or $t+1$.

Let's review the OOV attack strategy in the context of the proposed noise mode. Algorithm 3 of [119], an important optimization for the OOV basic attack, no longer takes effect. Thus the attacker only can launch the basic attack. If an OOV attacker uses $(\boldsymbol{b}', \boldsymbol{a}', \boldsymbol{y}')$ satisfying $\mathsf{Hwt}((\mathbf{C}^{[n\times m]}_{\boldsymbol{a}'\|\boldsymbol{b}'} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{y}') = w$ to interfere with the authentication of the HB$^{\mathcal{C}}$ protocol with the new noise mode, the success probability will be

$$p(w) = \frac{\binom{w}{\lfloor \frac{w}{2} \rfloor}\binom{n-w}{t-\lfloor \frac{w}{2} \rfloor} + \binom{w}{\lceil \frac{w}{2} \rceil}\binom{n-w}{t+1-\lceil \frac{w}{2} \rceil}}{\binom{n}{t} + \binom{n}{t+1}} = \frac{\binom{w}{\lfloor \frac{w}{2} \rfloor}\left(\binom{n-w}{t-\lfloor \frac{w}{2} \rfloor} + \binom{n-w}{t+1-\lceil \frac{w}{2} \rceil}\right)}{\binom{n+1}{t+1}} \quad . \tag{3.4}$$

Through an observation of one iteration of the protocol, the attacker obtains a tuple $(\widehat{\boldsymbol{b}}, \widehat{\boldsymbol{a}}, \widehat{\boldsymbol{y}})$ such that $\mathsf{Hwt}((\mathbf{C}^{[n\times m]}_{\widehat{\boldsymbol{a}}\|\widehat{\boldsymbol{b}}} \circ \boldsymbol{k}^T)^T \oplus \widehat{\boldsymbol{y}}) = w$, where $w = t$ or $t+1$. As a result, for $w = [0, t]$, the probability of the attacker correctly generating such a tuple $(\boldsymbol{b}', \boldsymbol{a}', \boldsymbol{y}')$ cannot exceed $\frac{\binom{t}{t-w}}{\binom{n}{t-w}}$. Therefore, the success probability of an OOV manipulated authentication is upper-bounded as

$$P_{\text{OOV}} \leq \max_{0\leq w\leq t} \left(\frac{\binom{t}{w}\binom{w}{\lfloor \frac{w}{2} \rfloor}\left(\binom{n-w}{t-\lfloor \frac{w}{2} \rfloor} + \binom{n-w}{t+1-\lceil \frac{w}{2} \rceil}\right)}{\binom{n}{t-w}\binom{n+1}{t+1}} \right) \quad . \tag{3.5}$$

For specific protocols, we can carefully choose practical parameters (n, t) such that P_{OOV} is negligible. Therefore, the OOV attack cannot succeed under polynomial bounds. As for the OOV2 attack, in which the attacks flips k bits of $\boldsymbol{y}$ for one iteration of protocol, the proposed noise mode prevents the attacker from learning a firm equation $(\mathbf{C}^{[n\times m]}_{\widehat{\boldsymbol{a}}\|\widehat{\boldsymbol{b}}} \circ \boldsymbol{k}^T)[j] =$ 1 or 0 from the authentication result. Thus the OOV2 attack does not work either.

Technically, our proposed new noise mode is a special case of a noise mode implicitly mentioned in [93], which we name the *upper and lower bounded Binomial noise mode*. In this noise mode, there are two distinct integer thresholds μ and τ satisfying that $\mu \leq \eta n \leq \tau$ and additionally we specify an important condition of $\mathsf{Pr}[\mathsf{Hwt}(\boldsymbol{v}) = \mu] \approx \mathsf{Pr}[\mathsf{Hwt}(\boldsymbol{v}) = \tau]$ to prevent the OOV2 attack; the tag always selects a random noise vector $\boldsymbol{v}$ with upper and lower bounded Hamming weight, namely, $\mu \leq \mathsf{Hwt}(\boldsymbol{v}) \leq \tau$; and the reader accepts the tag's authentication if and only if $\mu \leq \mathsf{Hwt}((\mathbf{C}^{[n\times m]}_{\boldsymbol{a}\|\boldsymbol{b}} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{y}) \leq \tau$. It is clear that setting $\mu = t$ and $\tau = t + 1$ renders minimal P_{OOV}, thus we choose this special case as our proposed new noise mode. Even though it is not provably affirmative about the hardness of the LPN variant with the upper and lower bounded Binomial noise mode noise model as well as our proposed new noise mode, we believe that it is asymptotically as hard as the original LPN problem.

After changing the Bernoulli noise mode to the new noise mode, the reduction procedures for all HB-like protocols, including Theorem 3.1 for $\mathrm{HB}^{\mathcal{C}}$, still hold. Even better, the inherent zero false negative bridge a security proof gap in many reduction proofs, e.g., Theorem 2 of [76], which requires the false negative rate P_{FN} to be negligible, while a non-negligible $P_{\mathrm{FN}} = 2^{-40}$ is recommended for practical concerns in [76]. The false positive rate in the HB-like protocol with this new noise mode is determined by

$$P'_{\mathrm{FP}} = \sum_{i=t}^{t+1} \binom{n}{i} 2^{-n} = \binom{n+1}{t+1} 2^{-n} \ , \tag{3.6}$$

and the number of different noise instances is

$$\rho = \sum_{i=t}^{t+1} \binom{n}{i} = \binom{n+1}{t+1} \ . \tag{3.7}$$

Insecurity of Another Node

Some may suggest a noise mode in which the Hamming weights of noise vectors are a constant integer t. However, the LPN variant problem on this noise mode is not hard at all. From each instance $(\mathbf{A}_i, \boldsymbol{y}_i = (\mathbf{A}_i \circ \boldsymbol{k}) \oplus \boldsymbol{v}_i)$ where $\mathsf{Hwt}(\boldsymbol{v}_i) = t$, an attacker, as pointed

out in [119], can learn

$$\bigoplus_{j=0}^{n-1}(\mathbf{A}_i \circ \boldsymbol{k})[j] = \bigoplus_{j=0}^{n-1} \boldsymbol{y}_i[j] \oplus \begin{cases} 1 & \text{if } t \text{ is odd} \\ 0 & \text{if } t \text{ is even} \end{cases} .$$

After gathering enough instances, the attacker can completely recover $\boldsymbol{k}$.

3.2.2 Masking for Increasing Noise Level

It is easy to see that the key length in LPN-based protocols is the most important parameter because it directly affects or even determines protocols computation, communication and storage costs. Since the key length and the noise level together dominate the security level, the noise level is as important as the key length. Intuitively, the lower η is, the easier an adversary can overcome noise; subsequently larger key length is required. Since smaller key length is preferred all the time, higher noise level is desired in practice. Unfortunately, larger η may result in unacceptable false rates. To improve protocol's performance, we introduce a simple but effective mechanism to substantially increase noise level from the adversary's perspective (and subsequently reduce key length) without undermining false rates.

Masking

Note that in the three HB-like protocols, the bit-length of response vector $\boldsymbol{y}$ is always less than the key size. Let $\boldsymbol{k}_n$ be a vector consisting of the leftmost n bits in the secret key. The tag uses $\boldsymbol{k}_n$ to mask LPN response $\boldsymbol{y}$ in the HB$^{\mathcal{C}}$ protocol, that is, calculates $\boldsymbol{z} = \boldsymbol{y} \oplus \boldsymbol{k}_n = (\mathbf{C}_{\boldsymbol{a}\|\boldsymbol{b}}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v} \oplus \boldsymbol{k}_n$, and accordingly the reader checks if $\mathsf{Hwt}((\mathbf{C}_{\boldsymbol{a}\|\boldsymbol{b}}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{z} \oplus \boldsymbol{k}_n) \leq \tau$. The masking mechanism does not affect the security proofs of HB$^{\mathcal{C}}$ in the GRS-model and Random-HB$^{\#}$/HB$^{\#}$ in the GRS-MIM-model.

Impact in the DET-model

By masking, a DET attacker cannot directly get a valid LPN instance from one authentication. Instead, to get rid of the masking effect of $\boldsymbol{k}_n$, the adversary has to combine two authentication interactions, say $(\boldsymbol{b}_1, \boldsymbol{a}_1, \boldsymbol{z}_1 = (\mathbf{C}_{\boldsymbol{a}_1\|\boldsymbol{b}_1}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}_1 \oplus \boldsymbol{k}_n)$ and $(\boldsymbol{b}_2, \boldsymbol{a}_2, \boldsymbol{z}_2 = (\mathbf{C}_{\boldsymbol{a}_2\|\boldsymbol{b}_2}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}_2 \oplus \boldsymbol{k}_n)$, and retrieves an LPN instance $((\boldsymbol{a}_1 \oplus \boldsymbol{a}_2)\|(\boldsymbol{b}_1 \oplus \boldsymbol{b}_2), \boldsymbol{z}' = (\mathbf{C}_{(\boldsymbol{a}_1\oplus\boldsymbol{a}_2)\|(\boldsymbol{b}_1\oplus\boldsymbol{b}_2)}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v}_1 \oplus \boldsymbol{v}_2)$, which effectively exposes noise level $\frac{3}{2}\eta$ to the adversary.

Tag ($\boldsymbol{k}$)		Reader ($\boldsymbol{k}$)
$\boldsymbol{b} \in_R \{0,1\}^{m_b}$	$\xrightarrow{\quad \boldsymbol{b} \quad}$	
	$\xleftarrow{\quad \boldsymbol{a} \quad}$	$\boldsymbol{a} \in_R \{0,1\}^{m_a}$
$\boldsymbol{v} \in_R \{\{0,1\}^n \mid \mathsf{Hwt}(\boldsymbol{v}) \in [t, t+1]\}$;		
$\boldsymbol{z} = (\mathbf{C}_{\boldsymbol{a}\|\boldsymbol{b}}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{v} \oplus \boldsymbol{k}_n$	$\xrightarrow{\quad \boldsymbol{y} \quad}$	
		? $\mathsf{Hwt}((\mathbf{C}_{\boldsymbol{a}\|\boldsymbol{b}}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \boldsymbol{z} \oplus \boldsymbol{k}_n) \in [t, t+1]$

Figure 3.3: Enhanced HB$^{\mathcal{C}}$ one-way authentication protocol, where $\boldsymbol{k} \in \mathbb{S}_m$, m is a P2 number, $m_a + m_b = m$, n is the interaction expansion and $n < m$, noise level $\eta \in (0, \frac{1}{2})$, $t = \lfloor \eta n \rfloor$

Impact on the OOV Attack

We still need the new noise mode to thwart the OOV attack, but the marking mechanism does impair the advantage of the OOV attacker, since now he prefers to combine two interactions into one interference as well. Otherwise, if the adversary only uses one interaction to launch the OOV manipulation, then the Hamming weight of resulting error vector affected by $\boldsymbol{k}_n$ will be greater than that in the case of two-instance combination.

By merging two protocol interactions, the OOV attacker obtains a tuple $(\widehat{\boldsymbol{b}}, \widehat{\boldsymbol{a}}, \widehat{\boldsymbol{y}})$ such that $\mathsf{Hwt}((\mathbf{C}_{\widehat{\boldsymbol{a}}\|\widehat{\boldsymbol{b}}}^{[n\times m]} \circ \boldsymbol{k}^T)^T \oplus \widehat{\boldsymbol{y}}) = w$, where the expected value of w is roughly $\lfloor 3t/2 \rfloor$. Let $t' = \lfloor 3t/2 \rfloor$. Therefore, when combining masking and the new noise mode, the success probability of an OOV manipulation authentication is bounded as

$$P'_{\mathrm{OOV}} \leq \max_{0 \leq w \leq t'} \left(\frac{\binom{t'}{w} \binom{w}{\lfloor \frac{w}{2} \rfloor} \left(\binom{n-w}{t-\lfloor \frac{w}{2} \rfloor} + \binom{n-w}{t+1-\lceil \frac{w}{2} \rceil} \right)}{\binom{n}{t'-w} \binom{n+1}{t+1}} \right) . \tag{3.8}$$

Enhanced HB$^{\mathcal{C}}$ One-Way Authentication Protocol

We refer to the resulting edition of applying those two mechanisms to an HB-like protocol as its *enhanced version*. The enhanced HB$^{\mathcal{C}}$ one-way authentication is depicted in Figure 3.3.

3.2.3 Parameters Selections

For enhanced HB-like protocols, to provide d-bit security, the parameters should satisfying the following conditions:

1. $m_a \geq d$. As pointed out in [100], the hardness of HB-like protocols against a DET adversary only relies on the m_b-bit LPN instances, because an active adversary always can set $\boldsymbol{a}_i = \mathbf{0}_{m_a}$, while $m_a \geq d$ suffices for d-bit security.

2. The selections of η and m_b jointly ensure that the best known algorithm (At present, it is LF algorithm [100].) cannot solve m_b-bit LPN instances of noise level η in less than 2^d steps.

3. Under a specified η, the selection of n guarantees that

   (a) $P'_{\mathrm{FP}} \leq 2^{-d}$, by Equation (3.6). This is due to the security proofs requirements and from practical concerns.

   (b) $\rho \geq 2^d$, by Equation (3.7). In other words, there should be more than 2^d noise instances.

   (c) $P'_{\mathrm{OOV}} \leq 2^{-d}$, by Equation (3.8). This is to prevent the OOV attack.

For the HB$^{\mathcal{C}}$ protocol, there are two additional requirements for parameters. First, since we exclude the case of $\boldsymbol{\psi}_i = \mathbf{1}_m$ in the GRS-model, and the interference vector $\mathbf{1}^m$ in the GRS attack does reveal the parity of $\boldsymbol{k}$, key length m in the HB-CM protocol should be increased by 1. Second, m should be a P2 number, which we can look up in Table 2.1.

Table 3.1: Enhanced HB-like protocols parameters selections, key storage consumption, and communication cost

Security Level d		80	112	128
Noise Level η		0.1667	0.1667	0.1667
Minimal m_b		512	680	768
Minimal m_a		80	112	128
Minimal n		467	656	751
Minimal $m_a + m_b$		592	792	896
Key Size	HB$^{\#}$	1524	2102	2396
	Revised HB$^{\#}$	1058	1447	1646
	HB$^{\mathcal{C}}$	613	797	907
Transmitted Bits	HB$^{\#}$	1059	1448	1647
	Revised HB$^{\#}$	1059	1448	1647
	HB$^{\mathcal{C}}$	1080	1453	1658

According to the LF algorithm [100], for noise level 0.25, $m_b = 512$, 680, and 768 provide 80, 112, and 128 security levels respectively. Due to the effect of masking, the actual noise level used in an enhance HB-like protocol can be $\eta = 2 * 0.25/3 = 0.1667$

Alice ($\mathbf{K}$)		Bob ($\mathbf{K}$)
$\boldsymbol{r}_1 \in_R \{0,1\}^{m_b}$	$\xrightarrow{\boldsymbol{r}_1}$	
	$\xleftarrow{\boldsymbol{r}_2}$	$\boldsymbol{r}_2 \in_R \{0,1\}^{m_a}$
$\boldsymbol{v}_1 \in_R \{\{0,1\}^n \mid \mathsf{Hwt}(\boldsymbol{v}_1) \in [t, t+1]\}$;		
$\boldsymbol{z}_1 = ((\boldsymbol{r}_1 \Vert \boldsymbol{r}_2 \Vert \mathsf{id}_a) \circ \mathbf{K}) \oplus \boldsymbol{v}_1 \oplus \boldsymbol{k}_n$;		
$\boldsymbol{r}'_3 \in_R \{0,1\}^{m_b - n}$	$\xrightarrow{\boldsymbol{z}_1, \boldsymbol{r}'_3}$	
		? $\mathsf{Hwt}((\boldsymbol{r}_1 \Vert \boldsymbol{r}_2 \Vert \mathsf{id}_a) \circ \mathbf{K}) \oplus \boldsymbol{z}_1 \oplus \boldsymbol{k}_n)$ $= t$ or $t+1$;
		$\boldsymbol{v}_2 \in_R \{\{0,1\}^n \mid \mathsf{Hwt}(\boldsymbol{v}_2) \in [t, t+1]\}$;
		$\boldsymbol{r}_3 = \boldsymbol{z}_1 \Vert \boldsymbol{r}'_3$
	$\xleftarrow{\boldsymbol{z}_2}$	$\boldsymbol{z}_2 = ((\boldsymbol{r}_2 \Vert \boldsymbol{r}_3 \Vert \mathsf{id}_b) \circ \mathbf{K}) \oplus \boldsymbol{v}_2 \oplus \boldsymbol{k}_n$
$\boldsymbol{r}_3 = \boldsymbol{z}_1 \Vert \boldsymbol{r}'_3$		
? $\mathsf{Hwt}((\boldsymbol{r}_2 \Vert \boldsymbol{r}_3 \Vert \mathsf{id}_b) \circ \mathbf{K}) \oplus \boldsymbol{z}_2 \oplus \boldsymbol{k}_n)$ $= t$ or $t+1$		

Figure 3.4: HB-M mutual authentication framework, where $\mathbf{K}$ is a secret $(n \times m)$-binary matrix, $m = m_a + m_b + m_c$, id_a and id_b are the m_c-bit identities of Alice and Bob respectively; noise level $\eta \in (0, \frac{1}{2})$, and noise degree $t = \lfloor \eta n \rfloor$.

for those m_b's and security levels. We then calculate the minimal values of n that ensure $P'_{\text{FP}} \leq 2^{-d}$, $\rho \geq 2^d$, and $P'_{\text{OOV}} \leq 2^{-d}$. The results of parameters and protocols performance in terms of key storage and communication cost for different security level are showed in Table 3.1. From this table, we can see that compared to HB$^{\#}$, the HB$^{\mathcal{C}}$ protocol reduces the key storage consumption almost by half, while the communication cost is only slightly increased, due to the extra requirement of m being a P2 number.

3.3 Mutual Authentication Framework

3.3.1 Framework Description

Now we extend enhanced HB-like one-way authentication protocols to mutual authentication. Those protocols can be unified under one HB-M authentication framework, which is illuminated in Figure 3.4. According to the different type of key matrix $\mathbf{K}$, this mutual authentication framework can be instantiated as:

1. Random-HB$^{\#}$-M mutual authentication protocol, if $\mathbf{K}$ is a random matrix.

2. HB$^{\#}$-M mutual authentication protocol, if $\mathbf{K}$ is a Toeplitz matrix.

3. HB$^{\mathcal{C}}$-M mutual authentication protocol, if $\mathbf{K}$ is a circulant-P2 matrix. In this case, $n < m$.

3.3.2 Analysis and Discussions

Roles of $\boldsymbol{z}_1$ and Construction of $\boldsymbol{r}_3$

In the HB-M mutual authentication framework, n-bit $\boldsymbol{z}_1$ has two roles. First, it serves as the authentication response of Alice to Bob. Second, it is used to construct the m_b-bit challenge vector $\boldsymbol{r}_3$ from Alice. Of course, it is rare that $n = m_b$. When $m_b < n$, $\boldsymbol{z}_1$ should be truncated to serve as $\boldsymbol{r}_3$. If $m_b > n$, Alice needs to provide an $(m_b - n)$-bit vector that is concatenated with $\boldsymbol{z}_1$ to output $\boldsymbol{r}_3$, which is the case adopted in Figure 3.4.

According to Lemma 1 in [93], the hardness of the LPN problem implies the pseudo-randomness of LPN instances. Therefore, if the underling HB-like protocol is secure in the DET-model, then an adversary impersonating Bob cannot distinguish $\boldsymbol{z}_1$ with a true random vector; thus $\boldsymbol{z}_1$ can be used to construct the challenge $\boldsymbol{r}_3$ from Alice.

As for a man-in-the-middle adversary, which inherently has the ability to change $\boldsymbol{r}_3$, the approach of replacing $\boldsymbol{r}_3$ by $\boldsymbol{z}_1$ does show a certain advantage, since Bob will use it only if the interactions $(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{z}_1)$ pass his verification; thus the adversary cannot change $\boldsymbol{z}_1$ as freely as manipulate $\boldsymbol{r}_3$. In this regard, $\boldsymbol{z}_1$ is like a certificated random sequence.

For an adversary that impersonates Alice, similarly, he cannot select the value of $\boldsymbol{v}_3$ freely because $\boldsymbol{v}_3$ is constructed by the certified $\boldsymbol{z}_1$. Therefore, in the second half authentication of Alice verifying Bob's response, the blinding vector can just be m_a bits, rather than m_b bits.

Parameter Selections

Compared to one-way authentication, the communication cost of the mutual authentication only increases roughly by n bits, since $n \approx m_b$ according to Table 3.1. The parameters determination guidelines and recommendations for η, m_a, m_b, n in Section 3.2.3 still hold in the HB-M mutual authentication.

Reflection Attack

A basic requirement for mutual authentication is to prevent the reflection attack, in which an adversary conducts two authentication procedures with one legitimate participant in

both directions simultaneously, of which one is the adversary as Alice and the participant as Bob, and the other is the adversary as bob and the participant as Alice, trying to successfully complete one of the authentications. As a common approach to thwart this attack, the participants identities are included in the HB-M mutual authentication. The formal analysis on the HB-M authentication preventing the reflection attack is given below.

Claim 3.2. *An HB-M mutual authentication protocol is secure against the reflection attack if its one-way authentication version can prevent the GRS attack.*

Justification. Suppose there is a legitimate user with identity id_α and an adversary wants to impersonate a user with identity id_β with user id_α by launching a reflection attack. During the interactions, the adversary merely forwards all the responses of user id_α in one procedure back to user id_α in the other and tries to pass the verification of user id_α in the second authentication procedure with a non-negligible probability.

Suppose the first three interactions in the both procedures are $(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{z}_1)$, then the following equation holds

$$\boldsymbol{z}_1 = ((\boldsymbol{r}_1||\boldsymbol{r}_2||\mathsf{id}_\alpha) \circ \mathbf{K}) \oplus \boldsymbol{v}_1 \oplus \boldsymbol{k}_n \ ,$$

where $\mathsf{Hwt}(\boldsymbol{v}) = t$ or $t+1$; and then in the second authentication procedure the adversary successfully impersonates user id_β if and only if

$$((\boldsymbol{r}_1||\boldsymbol{r}_2||\mathsf{id}_\beta) \circ \mathbf{K}) \oplus \boldsymbol{z}_1 \oplus \boldsymbol{k}_n = t \text{ or } t+1 \ .$$

The equation above is equivalent to

$$(((\mathbf{0}_{m_a+m_b}||(\mathsf{id}_\alpha \oplus \mathsf{id}_\beta))) \circ \mathbf{K}) \oplus \boldsymbol{v}_1 = t \text{ or } t+1 \ .$$

Now it is clear that the reflection attack is, in fact, a variant of the GRS attack in the proposed mutual authentication framework. If an enhanced HB-like protocol can prevent the GRS attack, then the probability of that equation holding is negligible, and subsequently the corresponding HB-M mutual authentication protocol is secure against the reflection attack. □

Participants Identities

In the HB-M authentication framework, the identities of Alice and Bob do not explicitly appear in the interactions. This arrangement is due to two considerations.

- Alice and Bob may have known the identities of each other prior to authentication; then they launch the authentication procedure to assure that the other party is really whom she/he claims to be.

- It is required for the privacy-preservation scenario, where one or both of the participants do not want to leak real identities to a passive eavesdropper. In this occasion, a central server takes the role of Bob, and there are many sensor nodes that share different pairwise keys with the server and act as Alice. Sensor nodes are aware of the identity of the server and perform the authentication accordingly; in contrast, the server only knows that the other party is one of sensor nodes, and then during the verification phase of Bob, the server enumerates all possible keys of sensor nodes, and tries the verification with each one of the keys until finding a pass. In this way, the server can successfully locate the real identity of "Bob" and then continue the procedure. Since the computation in the HB-M framework is lightweight, many rounds of verification computation would not incur formidable burdens on a powerful server, and thus our proposed HB-M framework provides a practical privacy-preservation approach.

3.4 Application Scenarios in Sensor Networks

To effectively use the mutual authentication in wireless sensor networks, nodes should have mechanisms to share authentication keys. According to different applications, there are four scenarios for authentication key establishment.

Scenario 1: Single global key shared by all nodes

This is the simplest but very useful scenario for sensor networks due to its low memory cost for sensor nodes. An invalid node, through receiving-forwarding transfer in two authentication procedures with two legitimate nodes id_α and id_β, still cannot impersonate any other legitimate node. The main disadvantage of this case is that an adversary who captures any node and extracts the global authentication key will compromise the security of all other nodes. Tamper-proof memory for the global key is a solution to the physical attack.

Scenario 2: Distinct key that each node shares with base station

This scenario is useful for authentication of users who want to query individual sensor's data. As a matter of fact, many proposed entity authentication schemes in wireless sensor networks mainly consider this scenario. As a typical example of this scenario, a nurse tries to query patients' physical information from body sensor networks using handset devices, which are defined as users of the sensor networks. The users, usually being high-capability devices with enough memory to hold all keys with nodes, move around to issue

task commands to the sensor networks and collect information. The authentication of users guarantees the patients' privacy, and the authentication of sensor nodes makes sure that users retrieve data from legitimate nodes. This scenario can be combined with the previous one to provide overall authentication for the whole sensor network.

Scenario 3: Pairwise key in every two nodes

If sensor nodes are subject to physical compromise in some application, then every two neighbor nodes should share different pairwise keys. If the network topology can be determined during the key pre-distribution stage, any two neighbor nodes can be preloaded with a pairwise key and this is a practical approach. However, sensor nodes are randomly deployed at most cases, then we have to ask every two nodes to be preloaded with a distinct pairwise key. Then the storage requirement in a single node is proportional to the number of all nodes in a network. Consequently, this approach is only suitable for small scale sensor networks.

Alternatively, we may use an initial-trust model that has been advocated in several WSN security proposals, such as initial-trust in [159] and smart trust in [12]. Upon deployment, every node explores its adjacent nodes, trusts all neighbors which are discovered, and creates pairwise keys between neighbors by negotiating them in plaintext within a short time slot T_{min}, because it is assumed that adversaries cannot launch any attack during the time slot T_{min}. After time T_{min}, any node, before any operations that would affect others, shall launch one HB-hybrid protocol by those pairwise keys with its neighbors. As argued in [12], due to the random employment characteristic of wireless sensor networks, this initial-trust model is relatively practical in some applications.

Scenario 4: Random predistributed keys in nodes

Random key predistribution approaches [59, 41, 111, 54] are one of the most prevalent techniques for key establishment in sensor networks. This technique can be directly applied to authentication key distribution between nodes, in order to increase resilience against physically compromising sensor nodes. Interestingly, we can only count on the basic Eschenauer-Gligor scheme [59] or its q-composite mode [41] since the motivation of our proposal is to design bit-operation-based-only approaches while other schemes [111, 54] require more complex calculations. In this application scenario, an offline server first generates a key pool of a large number of keys. Then every node is randomly preloaded with some keys out of the key pool before deployment. If two adjacent nodes happen to share at least one common key (or q common keys for the q-composite mode), they can directly verify the authenticity of the XORing result of all common keys by the proposed mutual authentication protocol. If they do not, they can rely on other nodes to facilitate their

authentication in a multi-hop manner. According to random graph theory [59], two nodes will almost be able to find an authentication path through multi-hop links if any two nodes share at least one common key with certain probability. Then these two nodes, finding this authentication path with overwhelming probability, verify each other via one-by-one verification relay in the path. Technically, our proposed protocol in this scenario is used to verify the authenticity of keys, rather than to authenticate node identities. Since nodes might be compromised and then their keys are leaked to the adversary, the success of verification only implies that participating nodes hold claimed keys.

3.5 Conclusions

In this chapter, we have proposed a concrete construction of HB-like protocol using circulant-P2 matrices, which overcome two inefficiencies in the HB$^{\#}$ protocol. In addition, we have introduced two enhancement techniques to prevent the OOV attack and improve protocol performance for HB-like protocols. Moreover, we have presented a mutual authentication framework for low-cost devices and its application scenarios in wireless sensor networks have been discussed at length.

Chapter 4

On the Distributed Detection of Node Clone

In this chapter, we present two innovative, practical node clone detection protocols with difference tradeoffs on network conditions and performance. The first protocol is based on the *distributed hash table* (DHT) [17], by which a fully decentralized, key-based caching and checking system is constructed to effectively catch cloned nodes. The protocol performance on security level as well as memory consumption is theoretically deducted through a probability model, and the resulting equations, with necessary adjustment for real application, are supported by the simulations. In accordance with our analysis, the comprehensive experimental results show that the DHT-based protocol can detect node clone with high security level and holds strong resistance against adversary's attacks.

Our second protocol, named *randomly directed exploration*, is intended to provide highly efficient communication performance with adequate detection probability for dense sensor networks. In this protocol, initially nodes send claiming messages containing neighbor-list with a maximum hop limit to randomly selected neighbors; then the subsequent message transmission is guided by a *probabilistic directed* technique to both roughly maintain a line property through the network and provide sufficient randomness for better performance on communication and resilience against the adversary. In addition, a border determination mechanism is proposed to further reduce communication payload. During forwarding messages, intermediate nodes explore claiming messages for node clone detection. By design, this protocol consumes almost minimal memory, and the simulations demonstrate that it outperforms all other detection protocols in terms of communication cost, while the detection probability is competitive.

The rest of the chapter is organized as follows. First we discuss preliminaries on network model, detection guidelines, performance metrics, and adversary model in Section 4.1. Afterwards, we present the DHT-based detection protocol and analyze its performance in

Section 4.2 and Section 4.3 respectively. Its simulation results are provided in Section 4.4. Then the randomly directed exploration protocol is detailed in Section 4.5, and its supportive experimental results are illustrated in Section 4.6. The conclusions are given in Section 4.7. Earlier versions of those two detection protocols can be found in [102, 103].

4.1 Preliminaries

As preliminaries of our proposals, we set up an appropriate network model, present general detection guidelines, define critical metrics to evaluate protocol performance, and discuss the adversary model.

4.1.1 Network Model

We consider a homogeneous sensor network consisting of n resource-constrained sensor nodes. The number of nodes n can be the order of hundreds and thousands. Consequently, the proposed sensor network protocols should scale to network size. In most of time, sensor nodes operate without any supervision, and they can function correctly in a dynamic network, where new nodes are added, or old nodes disappear. In addition to neighbor nodes, sensor nodes may be aware of some other distant nodes. However, it is not demanded for sensor nodes to realize the network geographic outline. The average node degree d varies with networks.

Analogous to the previous distributed detection approaches, we assume that an identity-based public-key cryptography facility [137] is available in the sensor network. Prior to deployment, each legitimate node is allocated a unique ID and a corresponding private key by a trusted third party. The public key of a node is its ID, which is the essence of identity-base cryptosystem. Consequently, no node can lie to others about its identity. Moreover, anyone is able to verify messages signed by a node using the identity-based key. Let K_α and K_α^{-1} denote the public and private keys of node α respectively, and $\{M\}_{K_\alpha^{-1}}$ represent the signature of M signed by node α.

It is true that conventional public key cryptosystems can satisfy the requirements of our protocols as well. However, the identity-based systems alleviate the heavy burden of public key certificates. In addition, our protocols, which will be detailed later, only use this cryptographic primitive for authentication rather than for encryption. Subsequently, it is not necessary for the identity-based cryptosystem here to be computationally-intense pairing-based schemes. Even though the usage of asymmetric primitives in sensor networks might be doubted several years ago, there are more and more WSNs security protocols relying on public key systems. In fact, some applicable implementation of such cryptosystems in typical sensor nodes platforms have been addressed in [109, 141, 118].

We also assume that every sensor node can determine its geographic location L via a secure localization mechanism. A number of those mechanisms have been proposed, which can be referred to in [127]. We do not specify the particular selection of secure localization protocols for our protocols, since it is comparatively irrelevant to our proposals. During the node clone detection protocol procedure, we suppose the sensor network to be stationary, where nodes locations remain unchanged.

There might be or not be a powerful base station in our modeled network, but there should exist a trusted role named *initiator* that is responsible for initiating a distributed detection procedure. Otherwise, an adversary can readily launch a denial-of-service (DoS) attack to the system by repeatedly mobilizing the sensor network to conduct the clone detection protocol and exhausting nodes energy. The requirement is easily satisfied in practice as the initiator can be the base station if one exists, or can be selected among all nodes via a distributed leader election, such as one in [45].

4.1.2 General Detection Guidelines

Relying on the identity-based cryptography and secure localization used in our network model, node clone in the stationary sensor network can be determined by the occurrence of nodes with same ID appearing at different locations. Specifically, at the beginning of a round of detection protocol, the information regarding the ID and location of every node is claimed by its neighbors for clone detection. In this sense, the neighbors of a node are its *observers*. Subsequently, some nodes will be selected as *inspectors* to examine claiming messages for the purpose of clone detection. If an inspector successfully finds clone, it becomes a *witness*, which will broadcast necessary evidence pertaining to cloned nodes to inform all connected nodes revoking the malicious nodes. While the initiator is presumably trusted, the other roles (observer, inspector, and witness) might be compromised by the adversary and behavior maliciously. The four roles in our protocols are summarized in Table 4.1.

Table 4.1: Four roles in the proposed detection protocols

Roles	Trusted	Duty
Initiator	Yes	Start a round of detection
Observer	No	Claim neighbors IDs and locations
Inspector	No	Buffer and check messages for detection
Witness	No	Broadcast detection evidence

4.1.3 Performance Metrics

The following metrics are used to measure the protocol performance and evaluate its practicability.

- **Detection Probability and Security Level**: As a primary security requirement, a practical detection scheme should detect the occurrence of the attack with high probability. Thus the *detection probability* is the most important security metric for a *probabilistic* clone detection scheme. On the other hand, if a detection protocol is *deterministic* in the sense that cloned nodes are always caught by witnesses, and it is also a fully *symmetric* approach in which nodes are equally likely to become witnesses prior to a round of detection procedure, we will use the number of witnesses to evaluate the security level, because more witnesses improve protocol resilience against the adversary's potential attacks to witnesses. .
- **Communication Cost**: In general, communication cost is always a crucial performance metric for sensor network protocols, because usually energy is the most valuable resource for sensor nodes, and message transmission consumes at least one order of magnitude power than any of other operations [10]. For simplicity, we use the average number of messages sent per node to represent a protocol's communication cost.
- **Storage Consumption**: Ordinary low-cost sensor nodes are only equipped with a limited amount of memory; thus any schemes requiring high storage will be considered as impractical. The protocol memory requirement is another performance metric for efficiency.
- **Balance**: In a homogeneous sensor network, schemes are supported to consume the energy and memory in a balanced way. It should be avoided to create hot nodes which would be buffer-overflowed or die away quickly.

4.1.4 Adversary Model

We consider a threat model in which sensor nodes are deployed in a hostile environment and are subject to capture and complete control by an adversary, but the adversary only can compromise a limited number of sensor nodes and then the adversary uses the compromised nodes to clone many nodes and deploys the replicas in places that are intelligently decided. To be concrete, we assume that the adversary is only allowed to control up to 10% sensor nodes and each cloned node has at least one neighbor that remains intact.

The adversary definitely wants to conceal the existence of clone. In our settings, this enemy is allowed to interfere with a detection protocol in the following three ways.

1. The cloned nodes may not participate in the regular detection procedures.
2. The nodes controlled by the adversary may fake, drop, or manipulate claiming messages which they forward.
3. The adversary can capture some nodes accordingly, but it would take time, and the total number of nodes that an adversary can compromise is limited. All nodes that are not controlled by the adversary will be referred to as *integrity nodes*.

Lastly, the adversary has the intention to launch DoS attacks to the sensor network by exploiting vulnerabilities of detection protocols.

4.2 DHT-Based Detection Protocol

The principle of our first distributed detection protocol is to make use of the distributed hash table (DHT) mechanism to form a decentralized, key-based caching and checking system that can effectively detect cloned nodes. In essence, this kind of system shares many common characteristics with P2P indexing services, which enormously benefit from DHT in the recent years. Indeed, DHT enables sensor nodes to distributively construct an *overlay network* upon a physical sensor network and provides an efficient key-based routing within the overlay network. A message associated with a key will be transmitted through the overlay network to reach a *destination* node that is solely determined by the key and will buffer the data for indexing services; the source node does not need to specify or know which node a message's destination is—the DHT key-based routing takes care of transportation details by the message's key. More importantly, messages with a same key will be stored in a unique destination node. Those properties inspire and build the foundation for our first detection protocol.

As an initialization of our proposed DHT-based clone detection, the initiator broadcasts an action message including a random *seed*. Then every observer generates a claiming message for each neighbor node, which is referred to as an *examinee* of the observer and the message, and sends the message with probability p_c independently. The introduction of the claiming probability p_c is intended to reduce the communication overwork in case of a high node degree network. In the protocol, a message's DHT key that decides its routing as well as its destination is the hash value of concatenation of the seed and the examinee's ID. By means of the DHT mechanism, a claiming message will eventually be transmitted to a deterministic destination node, which will save the examinee's ID-location information from the message and check for node clone detection, acting as an inspector. In addition, some intermediate nodes also behave as inspectors to enlarge the average number of witnesses in an efficient way. In order to detect clone, an inspector maintains a *cache-table* of dynamic

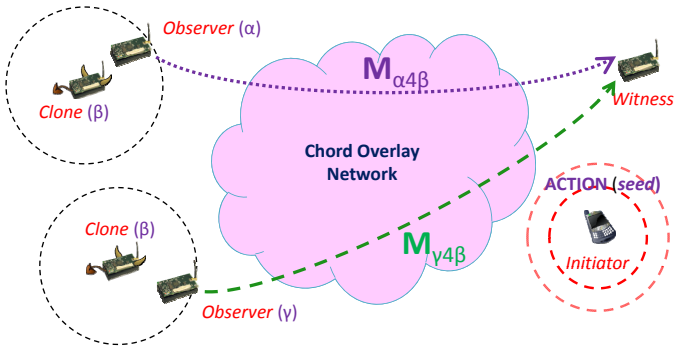


Figure 4.1: Outline of the DHT-based detection

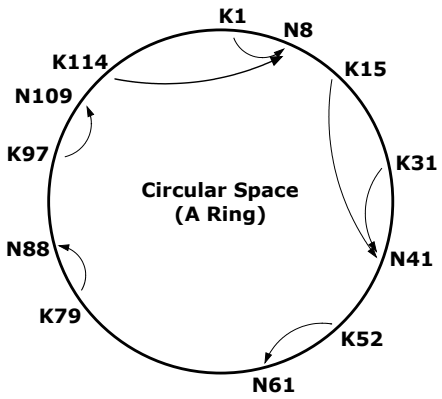
number of records, each of which contains ID-location information extracted from claiming messages that the inspector received and examined. We emphasize that records in a cache-table have different examinees IDs. Since the cache-table constitutes the main memory consumption of the DHT-based detection protocol, we measure the protocol's storage cost by the average value of final sizes of cache-tables. The DHT-based detection is outlined in Figure 4.1.

4.2.1 Distributed Hash Table

Before diving into the detection protocol, we briefly introduce distribute hash table techniques. In principle, a distribute hash table is a decentralized distributed system that provides a key-based lookup service similar to a hash table: (key, record) pairs are stored in the DHT, and any participating node can efficiently store and retrieve records associated with specific keys. By design, DHT distributes responsibility of maintaining the mapping from keys to records among nodes in an efficient and balanced way, which allows DHT to scale to extremely large networks and suitable to serve as a facility of distributed node clone detection. In fact, DHT techniques have been utilized to provide data-central storage and indexing services for distributed sensor networks, such as object-tracking in [53], and to design DHT-driven sensor network routing protocols [68, 38, 16].

There are several different types of DHT proposals, such as CAN [131], Chord [140], Pastry [136]. Generally, CAN has worse performance than others in terms of communication cost and scalability, and it is rarely employed in real systems. By contrast, Chord is probably the most widely implemented DHT, and we choose Chord as a DHT implementation to demonstrate our protocol. However, our protocol can easily migrate to build upon Pastry and present similar security and performance results.

The technical core of Chord [140] is to form a massive virtual ring in which every node is located at one point, owning a segment of the periphery. To achieve pseudo-randomness



Node	Key
N8	K114
	K1
N40	K15
	K31
N61	K52
N88	K79
N109	K97

N8's Finger Table

t	Key	Node
1	72	N88
2	40	N41
3	24	N41

Figure 4.2: A Chord network example, where the key space is 7-bit ($b = 7$), and seven records with different keys are stored in five nodes

on output, a hash function H is used to map an arbitrary input into a b-bit space, which can be conceived as a ring. Each node is assigned with a Chord coordinate upon joining the network. Practically for our protocol, a node's Chord point's coordinate can be the hash value of the node's MAC address. All n nodes divide the ring into n segments by their Chord points. Likewise, the key of a record is the result of the hash function. Every node is responsible for one segment which ends at the node's Chord point, and all records whose keys fall into that segment would be transmitted to and stored in that node.

For the purpose of efficient key-based routing, every node maintains a *finger table* of size $t = O(\log n)$ to facilitate a binary-tree search. To be specific, the finger table for a node with Chord coordinate y contains information of t nodes that are respectively responsible for holding the t keys: $(y + 2^{b-i}) \mod 2^b$ for $i \in [1, t]$. A demonstration example of a Chord system with small parameters is given in Figure 4.2. In this system, if node $N8$ wants to look up a record with key 97, after checking the finger table, it will contact node $N88$, which serves as a Chord intermediate node to transfer this request.

4.2.2 Protocol Details

As a preparation for subsequent detection procedures, all nodes cooperatively build a Chord overlay network over the sensor network. The construction of the overlay network is independent of node clone detection. As a result, nodes possess the information of their direct predecessor and successor in the Chord ring. In addition, each node caches information of its g consecutive successors in its *successors table*. Many Chord systems utilize this kind of cache mechanism to reduce the communication cost and enhance systems robustness. More importantly in our protocol, the facility of the successors table contributes to the economical selection of inspectors.

One round of DHT-based node detection consists of three stages.

Stage 1: Initialization

To activate all nodes starting a new round of node clone detection, the initiator uses a broadcast authentication scheme to release an *action message* including a monotonously increasing nonce and a random round seed. The nonce is intended to prevent adversaries from launching a denial of service (DoS) attack by repeating broadcasting action messages. The action message is defined by

$$M_{\text{ACT}} = nonce, seed, \{nonce \mid\mid seed\}_{K^{-1}_{\text{initiator}}} \ .$$

Stage 2: Claiming neighbors information

Upon receiving an action message, a node verifies if the message nonce is greater than last nonce and if the message signature is valid. If both pass, the node updates the nonce, stores the seed, and starts to operate as an observer that generates a claiming message for each neighbor (examinee) and transmits the message through the overlay network with respect to the claiming probability p_c. The claiming message by observer α for examinee β is constructed as follows:

$$M_{\alpha 4\beta} = \mathsf{id}_\beta, L_\beta, \mathsf{id}_\alpha, L_\alpha, \{\mathsf{id}_\beta \mid\mid L_\beta \mid\mid \mathsf{id}_\alpha \mid\mid L_\alpha\}_{K^{-1}_\alpha} \ .$$

Stage 3: Processing claiming messages

A claiming message will be forwarded to its destination node via several Chord intermediate nodes. Only those nodes in the overlay network layer (i.e., the source node, Chord intermediate nodes, and the destination node) need to process a message, whereas other nodes along the path simply route the message to temporary targets. Algorithm 4.1 for handling a message is the kernel of our DHT-based detection protocol. If the algorithm returns NIL, then the message has arrived at its destination. Otherwise, the message will be subsequently forwarded to the next node with the ID that is returned by Algorithm 4.1.

Criterions of determining inspectors: During handling a message in Algorithm 4.1, the node will act as an inspector if one of the following conditions is satisfied.

1. This node is the destination node of the claiming message.
2. The destination node is one of the g successors of the node. In other words, the destination node will be reached in one Chord hop.

Algorithm 4.1 *dht_handlemessage*($M_{\alpha 4\beta}$): handle a message in the DHT-based detection, where y is the current node's Chord coordinate, $finger[i]$ is the first node on the ring that succeeds key $((y + 2^{b-i}) \mod 2^b), i \in [1, t]$, $successors[j]$ is the next jth successor, $j \in [1, g]$

Output: NIL if the message has arrived at its destination; otherwise, it is the ID of the next node that continues receiving the message in the Chord overlay network

1: $key \Leftarrow \mathsf{H}(seed \,||\, \mathsf{id}_\beta)$
2: **if** $key \in (predecessor, y]$ **then** {has reached destination}
3: $inspect(M_{\alpha 4\beta})$ {act as an inspector, see Algorithm 4.2}
4: **return** NIL
5: **for** $i = 1$ to g **do**
6: **if** $key \in (y, successors[i]]$ **then** {destination is in the next Chord hop}
7: $inspect(M_{\alpha 4\beta})$ {act as an inspector, see Algorithm 4.2}
8: **return** $successors[i]$
9: **for** $j = 1$ to t **do** {for normal DHT routing process}
10: **if** $key \in [finger[j], y)$ **then**
11: **return** $finger[j]$
12: **return** $successors[g]$

While the first criterion is intuitive for a DHT-based indexing system, the second criterion is subtle and it results in an efficient protocol design. Since most of claiming messages related to a same examinee's ID would go through one of the g predecessors to reach the destination, those g nodes will be much more likely to be able to detect a clone than elsewhere randomly selected inspectors. As a result, this criterion to decide inspectors can increase the average number of witnesses at a little extra memory cost. We will theoretically quantify those performance measurements later.

In Algorithm 4.1, to examine a message for node clone detection, an inspector will invoke Algorithm 4.2, which compares the message with previous inspected messages that are buffered in the cache-table. Naturally, all records in the cache-table should have different examinee IDs, as implied in Algorithm 4.2. If detecting a clone, which means that there exist two messages $M_{\alpha 4\beta}$ and $M_{\alpha' 4\beta'}$ satisfying $\mathsf{id}_\beta = \mathsf{id}_{\beta'}$ and $L_\beta \neq L_{\beta'}$, the witness node then broadcasts the evidence, simply composed of the two messages, to notify the whole network.

$$M_{evidence} = M_{\alpha 4\beta}, M_{\alpha' 4\beta'}$$

It is worth noting that messages $M_{\alpha 4\beta}$ and $M_{\alpha' 4\beta'}$ are authenticated by observers α and α' respectively. Therefore, the witness does not need to sign the evidence message. If a malicious node tries to launch a DoS attack by broadcasting a bogus evidence message, the next integrity node receiving it can immediately detect the wicked behavior by verifying

Algorithm 4.2 *inspect*($M_{\alpha 4\beta}$): Inspect a message to check for clone detection in the DHT-based detection protocol

```
1: verify the signature of M_{α4β}
2: if id_β found in cache-table then
3:    if id_β has two distinct locations then {found clone, become a witness}
4:       broadcast the evidence
5: else
6:    buffer M_{α4β} into cache-table
7: return
```

the signatures of $M_{\alpha 4\beta}$ and $M_{\alpha' 4\beta'}$ before forwarding to other nodes.

All integrity nodes verify the evidence message and stop communicating with the cloned nodes. To prevent cloned nodes from joining the network in the future, a revocation list of compromised nodes IDs may be maintained by nodes individually.

4.2.3 Security Discussions

The identity-based cryptographic system provides reliable identity authentication and message authentication for the DHT-based protocol. As a result, the adversary cannot falsify clone nodes' ID; neither can he modify messages signed by integrity nodes. Moreover, a cloned node cannot lie to its observers about its location since a forged location would be far deviated from the communication range of the observers, which suffices to alert observers. Therefore, the detection guidelines are robust provided observers are honest.

As we discussed before, a witness cannot forge an evidence to attack integrity nodes, since the evidence eventually is composed of claiming messages from different observers, and any nodes can verify them. However, if some malicious observers try to abuse the detection protocol by claiming wrong locations for integrity nodes, those framed nodes may be wrongly removed from the sensor network. To prevent this attack, we may require nodes to buffer evidence messages they received. If nodes discover that an observer accounts for disposal of more nodes than a pre-defined threshold, they simply stop trusting evidences related to that observer. Alternatively, nodes might directly revoke both the observers and their accused "cloned" nodes in the evidence. In fact, we can combine both into an advance processing strategy. Nodes maintain a debit table for observers. When a node is declared as clone in one or several evidences, assume one of its distinct locations is claimed by q different observers, then each of those observers should be debited by $\frac{1}{q}$. If a node's balance in the debit table exceeds a threshold, it will be expelled. In this fashion, this abuse attack is not worthwhile for the adversary.

Technically, the hash functions used in DHT do not need to be cryptographic hash functions. In practice, the cryptographic ones are usually employed in the DHT systems because of their excellent uniformly random distribution of outputs for high protocol performance and additional one-way property to prevent potential abuses. For our protocol, a cryptographic hash function is indeed mandated, because it can restrain the adversary's abilities by its second preimage resistance on $\mathsf{H}(seed \mid\mid \mathsf{id})$ as he cannot distinguish which nodes could more likely become witnesses before a round of detection. For the specific hash function used in the protocol, at present we recommend SHA-1. Even though collisions have been found in SHA-1 [147], which raised serious concerns on its security, by principle it does not damage our protocol, since its security relies on second preimage resistance of underlying hash function, rather than the more strong requirement of collision resistance.

After disclosure of the random seed, the adversary may want to comprise witnesses to thwart detection. However, there are $g + 1$ potential witnesses that are geographically randomly distributed in the network. Determining and capturing all the witnesses will be troublesome for the adversary. Moreover, those witnesses vary round by rounds. Consequently, the adversary cannot stop the detection by trying to capture a few witnesses.

The cloned nodes may discard claiming messages that pass through them. Our protocol is resilient against this attack, due to the characteristic of full distributiveness and balance of the DHT-based protocol. If there are only a few clones nodes, the impact of this malicious action will be insignificant. When the number of cloned nodes increases, more claiming messages will assure sufficient number of witnesses. The simulations later clearly indicate this result. In summary, the DHT-based detection protocol is secure in the adversary model defined in Section 4.1.4.

4.3 Performance Analysis of DHT-Based Protocol

For the DHT-based detection protocol, we use the following specific measurements to evaluate the protocol performance.

- *Average number of messages transmitted by a node*, which measures the protocol communication cost.
- *Average size of node cache-tables*, which represents the protocol storage consumption.
- *Average number of witnesses*, which serves as the protocol security level, because the detection protocol is deterministic and symmetric.

4.3.1 Communication Cost

We denote the average path length between two random nodes by l, which varies from $O(\log n)$ to $O(\sqrt{n})$, dependent on underlying sensor networks. According to the Chord's properties [140], the average Chord-hop of a message, that is, the number of transfers in the Chord overlay network, is $c \log n$, where c is a constant number, usually less than 1. Therefore, the average path hop length of a message is $cl \log n$. There are $p_c dn$ claiming messages in total for a round of detection. Thus the average number of messages sent per node is given by $p_c dcl \log n$. Since the p_c, d, and c are constant, the asymptotic communication cost of the DHT-based protocol is between $O(\log^2 n)$ and $O(\sqrt{n} \log n)$.

4.3.2 Storage Consumption and Security Level

Now we theoretically analyze the average cache-table size denoted by s and the average witness number denoted by w. For simplicity, we hereby assume that all nodes, including compromised ones, abide by the detection protocol. Later in the next section of simulations, we will see that the malicious behaviors such as discarding claiming messages only slightly affect those performance measurements.

In the DHT-based detection protocol, claiming messages associated with a same examinee's ID will be transported to one destination node. Because there are n examinees and n potential destinations, and due to the well pseudo-randomness of the Chord system, on average, every node stores one record in its cache-table associated with one examinee's ID as its destination, regardless of the number of claiming messages per examinee. In addition, for a designated examinee, the g predecessor nodes of the destination can act as inspectors; thus they probably hold up to one record related to the examinee.

Assume that there are m independent claiming messages for every examinee, and let p_r denote the probability of a predecessor receiving a specific claiming message, then the probability of a predecessor holding a record for an examinee is $1-(1-p_r)^m$. Consequently, the average cache-table size can be calculated by

$$s = 1 + g(1 - (1 - p_r)^m) \ .$$

If there are δ ($\delta \geq 2$) cloned nodes with a same ID in the network, then their destination node (from the point of view of claiming messages related to those nodes as examinees) will deterministically become a witness to successfully catch the attack, while the g predecessor nodes of the destination may become the witnesses if and only if they receive at least two claiming messages associated with different cloned nodes. It is easy to see that this probability is minimized as $(1 - (1 - p_r)^{2m})$ when $\delta = 2$. Therefore, as a lower-bound for

witness number, we only analyze the case that there are two cloned nodes. In this case, the average witness number can be obtained by

$$w = 1 + g(1 - (1 - p_r)^{2m}) \ .$$

A Continuous Probability Model and Its Distribution

The analysis above conveys the basic idea of performance deductions. However, to achieve practical results of s and w, we need to use a continuously probability model to quantify p_r and conduct the deductions through continuous random variables. Based on the properties of Chord and Algorithm 4.1, a continuous probability model to approximate p_r is defined as follows.

Definition 4.1 (Random line segmentation model)**.** *We start with a line of length 1, of which the starting and ending points coordinates are 0 and 1 respectively. Then we randomly generate g values over $[0, 1]$ as point coordinates, and let random variable X_i represent the ith point, for $i = 1, 2, \cdots, g$. Those g continuous random variables divide the line into $g + 1$ segments. We want to determine the continuous probability distributions of segment lengths.*

Since g intermediate points are randomly selected, all $g + 1$ segment lengths have a same probability distribution. Let R denote the random variable of the length of the first segment, which origins from 0. As a rule, we denote density function by f, cumulative distribution function by F, expected value by E. Since the sample spaces of all random variables in this section are $[0, 1]$, then $0 \leq x \leq 1$ is implied for cumulative distribution functions and density functions. The following theorem states the probability distribution of R.

Theorem 4.1. *The cumulative distribution function of R is*

$$F_R(x) = 1 - (1 - x)^g \ .$$

Accordingly, its density function is

$$f_R(x) = g(1 - x)^{g-1} \ .$$

Proof. According to Definition 4.1, X_i is uniformly distributed over $[0, 1]$ (i.e., $F_{X_i}(x) = \mathsf{Pr}(X_i \leq x) = x$) and is independent of each other. This theorem can be proved by induction.

Basis: Assume $g = 1$, then $R = X_1$, and it holds that $F_R(x, g) = 1 - (1 - x)^g = x = F_{X_1}(x)$.

Inductive step: If $F_R(x,g) = 1-(1-x)^g$ holds, then we have

$$\begin{aligned} F_R(x,g+1) &= \mathsf{Pr}(R \le x) = \mathsf{Pr}\left(\min_{i=1,\cdots,g+1}(X_i) \le x\right) \\ &= \mathsf{Pr}\left(\min_{i=1,\cdots,g}(X_i) \le x\right) + \mathsf{Pr}(X_{g+1} \le x) \\ &\quad - \mathsf{Pr}\left(\min_{i=1,\cdots,g}(X_i) \le x\right)\mathsf{Pr}(X_{g+1} \le x) \\ &= F_R(x,g) + x - xF_R(x,g) \\ &= 1-(1-x)^g + x(1-x)^g = 1-(1-x)^{g+1} \quad . \end{aligned}$$

Since both the basis and the inductive step have been proved, it has now been proved that the cumulative distribution function of R is $1-(1-x)^g$. By differentiation, its density function is $f_R(x) = F'_R(x) = g(1-x)^{g-1}$. □

Analytic Formulas for Average Cache-table Size and Average Witness Number

The g intermediate points in the proceeding model are corresponding to the g predecessors, and the ending point represents the destination. For a claiming message, the probability of its destination being its sender is $\frac{1}{n}$. Since the network size n is large-scale, we may safely assume $\frac{1}{n} \approx 0$ for simplicity. In addition, due to pseudorandomness of the Chord system, it is reasonable to assume that one claiming message randomly occurs in that line before the final transmission to the destination. Therefore we can directly adopt Theorem 4.1 as the probability distribution of a predecessor receiving one claiming message.

Theorem 4.2. *In an ideal case where there are m independent claiming messages for each examinee and g is the successors table size, the average size of cache-tables is*

$$s = 1 + \frac{gm}{g+m} \quad , \tag{4.1}$$

and the average witness number, when there are two cloned nodes, is

$$w = 1 + \frac{2gm^2}{(g+2m)(g+m)} \quad . \tag{4.2}$$

Proof. For a specific examinee's ID, let random variable S represent the probability of a predecessor receiving at least one claiming message, which results in one record in its cache-table. By the analysis before, $S = 1-(1-R)^m$; thus we can get the cumulative

function of S by

$$\begin{aligned} F_S(x) &= \mathsf{Pr}(S \le x) = \mathsf{Pr}\left(1-(1-R)^m \le x\right) \\ &= \mathsf{Pr}\left(R \le 1-\sqrt[m]{1-x}\right) = F_R(1-\sqrt[m]{1-x}) \\ &= 1-(1-x)^{g/m} \ , \end{aligned}$$

and the density function is obtained by

$$f_S(x) = F'_S(x) = \frac{g}{m}\left(1-x\right)^{g/m-1} \ .$$

Thus the expected value of S is

$$E(S) = \int_0^1 x f_S(x)\mathrm{d}x = \frac{m}{g+m} \ .$$

When there are two clone nodes, each of them incurs m independent claiming messages. Let random variable W represent the probability of a predecessor becoming a witness. Because $W = (1-(1-R)^m)^2$, we have

$$\begin{aligned} F_W(x) &= \mathsf{Pr}(W \le x) = \mathsf{Pr}\left((1-(1-R)^m)^2 \le x\right) \\ &= \mathsf{Pr}\left(R \le 1-\sqrt[m]{1-\sqrt{x}}\right) = F_R\left(1-\sqrt[m]{1-\sqrt{x}}\right) \\ &= 1-(1-\sqrt{x})^{g/m} \ , \end{aligned}$$

and

$$f_W(x) = \frac{g}{2m\sqrt{x}}\left(1-\sqrt{x}\right)^{g/m-1} \ .$$

Then the expectation of W is

$$E(W) = \int_0^1 x f_W(x)\mathrm{d}x = \frac{2m^2}{(g+2m)(g+m)} \ .$$

Therefore, in the ideal case, the average size of cache-tables is

$$s = 1 + gE(S) = 1 + \frac{gm}{g+m} \ ,$$

and the average witness number is

$$w = 1 + gE(W) = 1 + \frac{2gm^2}{(g+2m)(g+m)} \ .$$

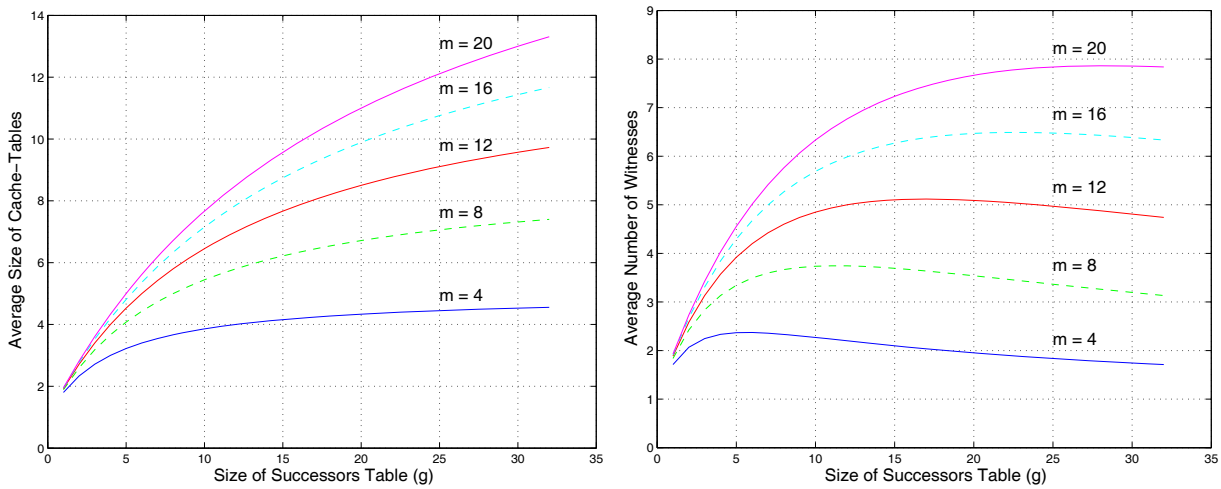


Figure 4.3: Storage cost and security level in the ideal case, where there are m independent claiming messages per examinee, and the witness number is based on two cloned nodes setting

That concludes the proof. □

We implement the experiment of the ideal case on random line segmentation model by Python, and carry out simulations to measure s and w for different values of m and g. The experimental results statistically match the outputs of Equations (4.1) and (4.2).

Figure 4.3 depicts the plots of the average number of witnesses and the average size of cache-tables as functions of the size of successors table with various numbers of independent claiming messages m in the ideal case. It is worth mentioning that if m is constant, the average witness number is maximized when

$$g = \sqrt{2}m \ ,$$

because by Equation (4.2),

$$w = 1 + \frac{2gm^2}{(g+2m)(g+m)} = 1 + \frac{2m^2}{\left(g + \frac{2m^2}{g}\right) + 3m} \ ,$$

and the non-negative denominator takes its minimum value when $g = \sqrt{2}m$. Consequently, the maximum witness number is

$$1 + \frac{2}{2\sqrt{2}+3}m \approx 0.343m \ .$$

Remarks

In the reality of the DHT-based protocol, each examinee, on average, has dp_c claiming messages, thus $m = dp_c$. Unfortunately, the transmissions of claiming messages associated with a same examinee's ID are in fact not independent in the Chord key-based routing system. It is very difficult to analyze performance for the dependent case. Fortunately, through simulation studies, we discover that we may use those equations for the ideal case with some necessary adjustment to approximate the protocol practical performance on storage consumption and security level. Specifically, the message correlation impact can be mimicked by decreasing g to a certain proportion, when we apply Equations (4.1) and (4.2) to estimate the average cache-table size and the average witness number. In other words, even though the actual successors-table size is g, when we calculate performance measurements by those two equations, we use g', which is smaller than g with respect to a certain ratio. We will demonstrate it in the next section of thorough simulations.

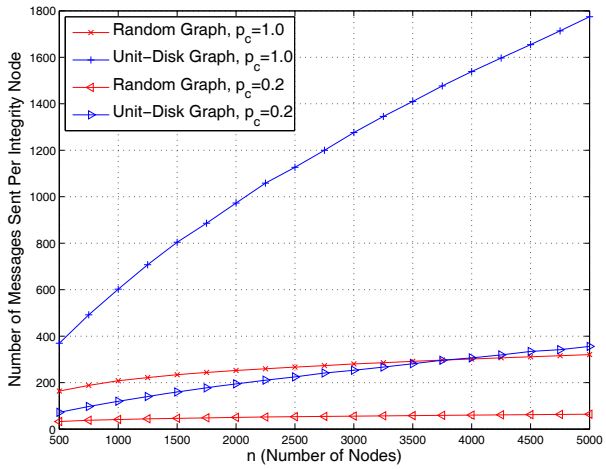
4.4 Simulations for DHT-Based Protocol

We implement the DHT-based detection protocol and run simulations to evaluate performance comprehensively on our extended framework over the OMNeT++ simulator (see Section 1.4). We use two network scenarios: Random Graph $G(n, d)$ as an ideal random network testing scenario and Unit-Disk Graph as a practical one. In the latter scenario, node communication ranges are dynamically adjusted such that the average node degree approximates d.

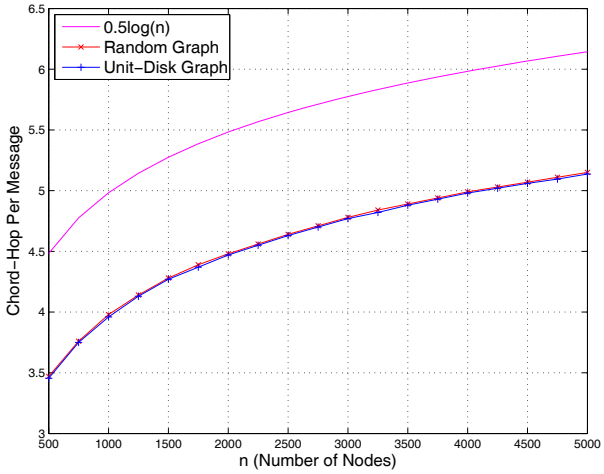
4.4.1 Performance on Varying Network Sizes

The following parameters are used in the simulations: finger table size $t = 16$, successors table size $g = 16$, and node degree $d = 20$. Two different values of claiming probability p_c are used as 1.0 for pro-security and 0.2 for pro-communication cost. Overall, we have four groups: Random Graph with $p_c = 1.0$, Unit-Disk Graph with $p_c = 1.0$, Random Graph with $p_c = 0.2$, and Unit-Disk Graph with $p_c = 0.2$.

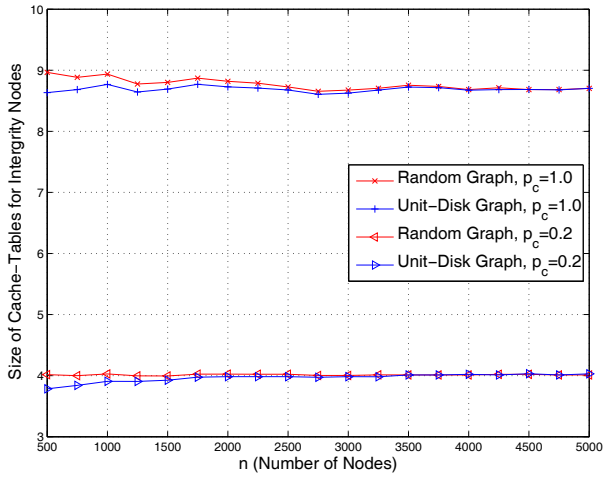
We design and conduct the first experiment to measure the protocol's performance on different network sizes, ranging from 500 to 5000, with a step of 250. Since we have four groups, we need to test $4 \times 19 = 76$ cases. In order to obtain relatively fair and comparable results, for each case, ten different network instances in accordance with the parameter settings are constructed. Each of those simulation executions is quoted as a run; and one run performs 20 rounds of detection. In each of those rounds, a random seed is generated and two nodes are randomly chosen to set the same ID, that is, those two are cloned nodes.



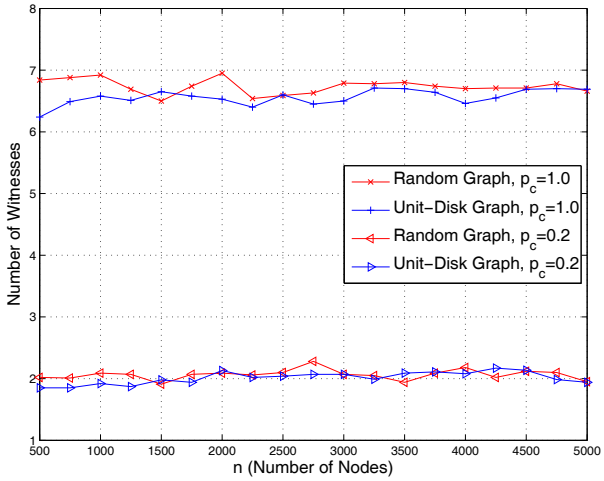
(a) Communication Cost



(b) Chord-Hop Per Message



(c) Storage Consumption



(d) Security Level

Figure 4.4: Simulation results of the DHT-based detection on varying network sizes, where there are two cloned nodes

The average number of messages sent per integrity node in this experiment is plotted in Figure 4.4a. In the graph, the communication payloads for groups of $p_c = 0.2$ are exactly the one-fifth of those of $p_c = 1.0$ with the same other parameters. The distinction between simulation results of Random Graph and those of Unit-Disk Graph results from different average path hops l for a pair of random nodes in those two network scenarios. Indeed, the average Chord-hop per message in the simulations, depicted in Figure 4.4b, is independent of the network scenarios and matches the Chord system properties—it is proportional to $\log n$.

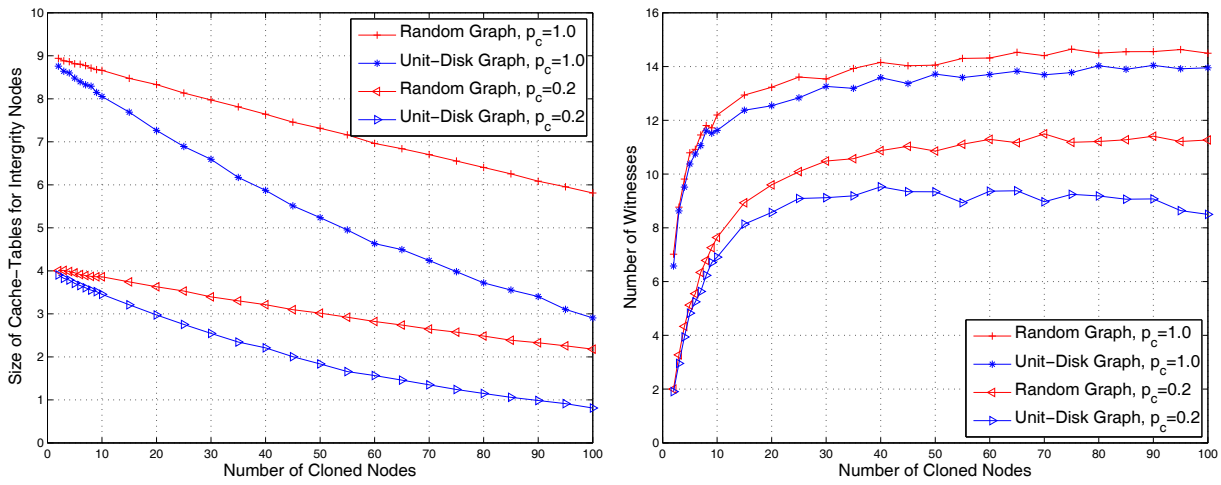
The average size of cache-tables for integrity nodes and the average witness number are illuminated in Figure 4.4c and Figure 4.4d respectively, which clearly indicate that those two performance metrics are not notably affected by network scenarios and network size n if n is sufficiently large, and thus the detection protocol does scale to network size. Even though the network topology for Unit-Disk model used in the simulations is square, the protocol can effectively detect clone attack in irregular topologies, like "Thin H", "Thin Cross", "S", "Large H", "L", and "Large H", which are used in the simulations of [121]. The communication payload of our DHT-based protocol for those network topologies is proportional with the average path hop l, while the security level and storage payload stay constant.

4.4.2 Results on Different Numbers of Cloned Node

We develop the second experiment to evaluate the protocol's performance on the different numbers of cloned nodes. Still using the four groups, we run simulations with one network size $n = 1000$ and the cloned node number δ increases from 2 to 100. We test each case with ten runs, and for each run we repeat 200 rounds of node detection, in each of which a seed is randomly generated and δ nodes are randomly chosen as clones.

Figure 4.5 depicts the experimental results about the average size of cache-tables for integrity nodes and the average number of witnesses, which support our security arguments in Section 4.2.3. In particular, we can see that the protocol shows strong resilience against message-discarding of cloned nodes. Even if there are 10% nodes that maliciously discard messages, the number of witnesses is pretty high. In fact, the more cloned nodes, the less size of cache-tables for integrity nodes as storage consumption and the more witnesses as security level. Therefore, we really only need to consider the boundary case of $\delta = 2$ for performance measurements.

In Figure 4.5, when there are more than 1% cloned nodes, the simulation results for Random Graph and Unit-Disk Graph are evidently distinct. This is because the message-dropping by malicious nodes affects the performance to a different extent. As implied in Figure 4.4a, the average transmission hop of claiming messages in Unit-Disk Graph



(a) Storage Consumption (b) Security Level

Figure 4.5: Simulation results of DHT-based detection on different number of cloned nodes, where $n = 1000$

is greater than that in Random-Graph, then messages are more likely to be dropped by cloned nodes in Unit-Disk Graph.

4.4.3 Verification of Performance Analysis

To evaluate its applicability of the theoretical analysis on cache-table size and witness number in Section 4.3.2, we carry out the third experiment, in which $n = 2000$, $d = 40$, there are two cloned nodes, and the claiming probability p_c increases from 10% to 100% at an interval of 5%. Since network scenarios do not affect the results on the two performance metrics, we only run the simulation on Random Graph. For the purpose of comparison, we test the performance for both cases that cloned nodes drop messages and comply with protocol (no-dropping).

We argued before that to apply Equations (4.1) and (4.2), we should adjust the successors-table size g to balance the effect of message transmission correlation. Figure 4.6 depicts the experimental results along with theoretical outputs by adopting 80% of original g in those two equations. From this figure, first we can see that message-dropping strategy by malicious nodes indeed does not affect the performance notably. Second, the theoretical outputs do reflect the practical results after we choose a proper adjustment on g. Admittedly, the specific adjustment value is related to Chord system parameters. Interestingly, for several other testing cases, such as $(n = 3000, g = 16)$ and $(n = 2000, g = 8)$, the 80%

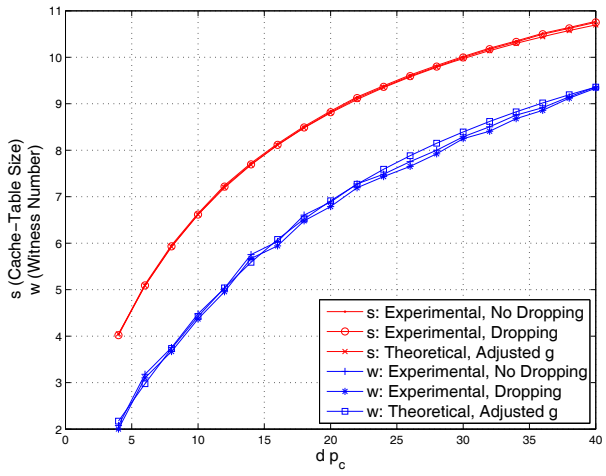


Figure 4.6: Simulation results for verifying performance analysis of the DHT-based detection, where g is adjusted by 80% in the theoretical calculation

appears good enough for evaluation purpose.[1]

4.4.4 Discussions

From the simulation results, we can see that the proposed DHT-based protocol can effectively detect clone for general sensor networks with high security level and efficient storage consumption, while its communication cost is in the same order of magnitude with previous detection schemes. One way to improve the communication performance is replacing the Chord overlay network with some specific DHT implementations on sensor networks. A number of such protocols, such as Scalable Source Routing [68], Virtual Ring Routing [38], and Virtual Cord Protocol [16], have been proposed. Combinations of our protocol with those DHT schemes might be an interesting research topic. On the other hand, if it is very likely for the adversary to deploy many clones out of one ID, we can use small claiming probability p_c for saving communication payload, without degrading the security level (i.e., the average witness number) dramatically.

The protocol security level, by Equation (4.2), is directly determined by message number m, which is upper bounded by node degree d. [2] If the DHT-based protocol operates in sparse sensor networks, to achieve a desirable witness number, several independent Chord systems by different round seeds may be used. Thus messages for one examinee would be

[1] In our simulations, the best selections of adjustment for $(n = 3000, g = 16)$ and $(n = 2000, g = 8)$ are 79% and 81% respectively, which produce similar match of experimental results and theoretical outputs.

[2] There is no point for $p_c \geq 1$. Because even if a node repeats sending claiming messages corresponding to one examinee, all those messages will follow a same transmission path.

indexed by a few deterministic destinations. In this case, the protocol's communication payload, storage consumption, and the average witness number are all multiplied by a same order. Note that this is also a common practice in P2P indexing systems to enhance robustness.

4.5 Randomly Directed Exploration Protocol

The DHT-based detection protocol can be applied to every kind of sensor network, and its security level is remarkable, as cloned nodes will be caught by one deterministic witness plus several probabilistic witnesses. However, the message transmission over a Chord overlap network incurs considerable communication cost, which may not be desired for some sensor networks that are extremely sensitive to energy consumption. To fulfill this challenge, we propose the second distributed detection protocol—randomly directed exploration, which tremendously reduces communication cost and presents optimal storage expense with adequate detection probability.

The randomly directed exploration protocol is actually inspired by several key observations to the node-to-network broadcasting scheme. We notice that this approach, despite its worst communication performance among all distributed approaches, is quite attractive from many other aspects. It does not require any additional assumption—every node just simply broadcasts its neighbor-list to other nodes. In addition, each node need only buffer its own neighbor-list, so the memory requirement is pretty low, except for the additional memory cost for preventing from receiving same broadcasting messages. For a dense sensor network, suppose that there are two cloned nodes in the network, each of which has d integrity neighbors, node-to-network broadcasting will drive all those neighbors to find clone, but in fact one witness that successfully catches the clone and then notifies all the network would suffice for the detection purpose. In this regard, instead of broadcast, anycast [153] seems a good alternative for routing claiming messages. However, we cannot afford an infrastructure for existing anycast protocols in sensor networks, and indeed it is unnecessary. In our proposed randomly directed exploration protocol, which is outlined in Figure 4.7, a claiming message initially is sent towards a random direction, and then its subsequent transmission will roughly maintain a line, while all intermediate nodes check the message to find clone.

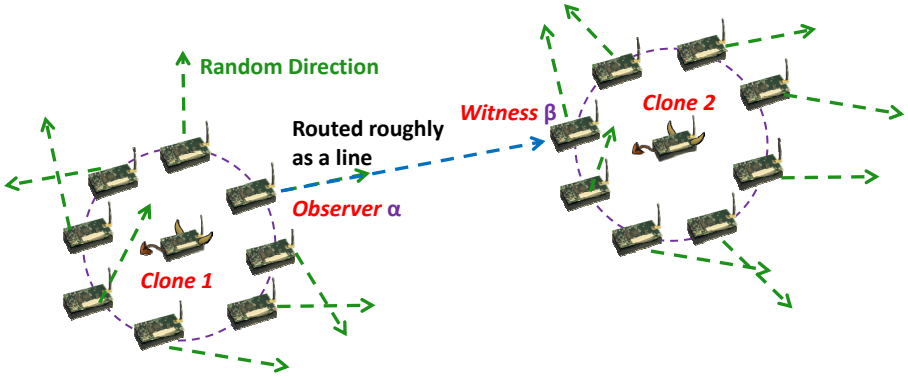


Figure 4.7: Outline of the randomly directed exploration protocol

4.5.1 Protocol Description

Preparation

To prepare for node clone detection, every node, upon deployed, informs all its neighbors about its ID and location. Then each node creates its own neighbor-list including the neighbors IDs and locations. Note that this kind of neighbor notification phase is necessary for many wireless sensor networks. In the randomly directed exploration protocol, this neighbor-list is used for assisting message routing and for clone detection, and indeed it constitutes the sole storage consumption.

Detection Procedure

One round of clone detection is still initialized by the initiator, which simply broadcasts an authenticated action command. During the detection procedure, each node, as an observer for all its neighbors, starts to generate a claiming message containing its own ID, location, and its neighbor-list. Formally, the claiming message by node α is defined as follows:

$$M_\alpha \quad = \quad ttl, \mathsf{id}_\alpha, L_\alpha, NeighborList_\alpha, \{\mathsf{id}_\alpha, L_\alpha, NeighborList_\alpha\}_{K_\alpha^{-1}} \quad ,$$

where *ttl* represents *time to live*, defined as the maximum number of hops for a claiming message. Since *ttl* will be altered by intermediate nodes during transmission, it should not be authenticated. The observer tries to deliver the claiming message r times. In each time, the node transmits it to a randomly selected neighbor. Note that r can be a real number and accordingly an observer transmits its claiming message at least $\lfloor r \rfloor$, up to$\lceil r \rceil$, and on average r times.

When an intermediate node β receives a claiming message M_α, it invokes Algorithm 4.3

Algorithm 4.3 *rde_processmessage*(M_α): An intermediate node processes a message in the randomly directed exploration protocol

```
1: verify the signature of M_α
2: compare its own neighbor-list with the neighbor-list in M_α
3: if found clone then
4:    broadcast the evidence;
5: ttl ⇐ ttl − 1
6: if ttl ≤ 0 then
7:    discard M_α
8: else
9:    nextnode ⇐ getnextnode(M_α) {See Algorithm 4.4}
10:   if nextnode = NIL then
11:      discard M_α
12:   else
13:      forward M_α to nextnode
```

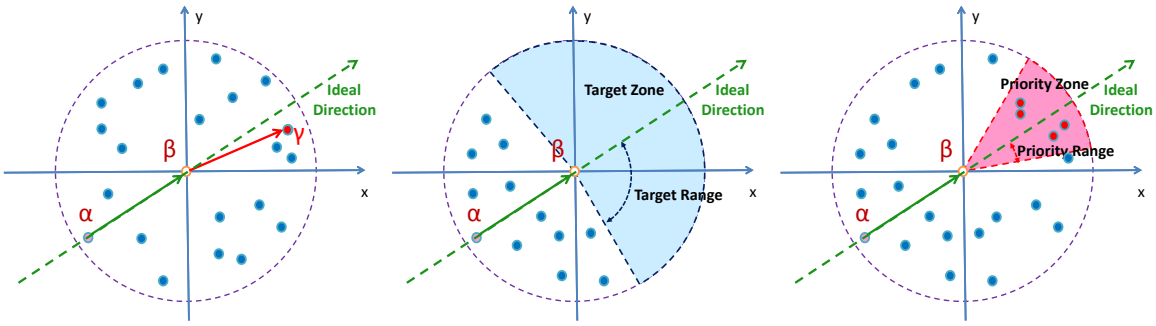
Algorithm 4.4 *getnextnode*(M_α): Determine the next node that receives the message

```
1: determine ideal angle, target zone, and priority zone
2: if no neighbors within the target zone then
3:    return NIL
4: if no neighbors within the priority zone then
5:    nextnode ⇐ the node closest to ideal angle
6: else
7:    nextnode ⇐ a probabilistic node in the priority zone, with respect to its probability
      proportional to angle distance from priority zone border
8: return nextnode
```

of *rde_processmessage*(M_α) to process the message. During the processing, node β, as an inspector, compares its own neighbor-list with the neighbor-list in the message, checking if there is a clone. Similarly, if detecting a clone, the witness node β would broadcast an evidence message $M_{evidence} = M_\alpha, M_\beta$ to notify the whole network such that the cloned nodes are revoked by all integrity nodes. To deal with routing, node β decreases the message's *ttl* by 1, and discards the message if *ttl* reaches zero; elsewhere, to determine the next node receiving the message, it will query Algorithm 4.4, which, as a technical core of the protocol, will be discussed at length in next section.

Determination of Next Node

Essentially, Algorithm4.4 contains the following three mechanisms:



(a) Deterministic Directed Transmission (b) Border Determination (c) Probabilistic Directed Transmission

Figure 4.8: Routing mechanisms in the randomly directed exploration protocol

- *Deterministic Directed Transmission* (Figure 4.8a): Since every node has information about its own location and all neighbors locations, it can readily calculate all neighbors angels. When node β receives a claiming message from previous node α, the ideal direction can be calculated. In order to achieve the best effect of line transmission, the next destination node should be node γ, which is closest to the ideal direction.

- *Network Border Determination* (Figure 4.8b): This takes network shape into consideration to reduce the communication cost. In many sensor network applications, nodes are randomly deployed, and there exist outside borders of network. When reaching some border in the network, the claiming message can be directly discarded. In our proposal for border local determination, another parameter *target range* is used along with ideal direction to determine a *target zone*. When no neighbor is found in this zone, the current node would conclude that the message has reached a border, and thus throw it away.

- *Probabilistic Directed Transmission* (Figure 4.8c): In the probabilistic directed transmission, parameter *priority range* along with the ideal direction is used to specify a priority zone, in which the next node will be selected. When no nodes are located in that zone, the deterministic directed candidate within the target zone will be selected as the next node. If there are several nodes in the priority zone, their selection probabilities are proportional to their angle distances to priority zone border. In this way, the desired line transmission property is reserved, while a certain extent of important randomness is introduced.

If purely using deterministic directed mechanism for routing, except for a limited number of initial random directions, a claiming message will be transmitted in a deterministic

manner. As a result, the adversary may remove some nodes in strategic locations to reduce detection probability dramatically. Moreover, transmissions of claiming messages from a cloned node's neighbors are highly correlated, which affects the protocol communication and security performance. By the elegant probabilistic directed mechanism, those drawbacks are overcome, and the protocol performance is improved significantly, which is supported by the simulations in the next section.

4.5.2 Analysis

The randomly directed exploration protocol is exceedingly memory-efficient. It does not rely on broadcasting. Consequently, no additional memory is required to suppress broadcasting flood. The protocol does not demand intermediate nodes to buffer claiming messages, overcoming main disadvantage of line-selected multicast scheme [121]. All memory requirement lies on the neighbor-list. In fact, the neighbor-list is a necessary component for all distributed detection approaches. Therefore, the protocol consumes almost minimum memory.

The communication cost of the randomly directed exploration depends on the routing parameter settings. On average, there are r claiming messages sent by per observer, and each message transmits at most ttl hops. For a dense sensor network, $r = 1.0$ might be a proper choice, and $ttl = \sqrt{n}$ would be sufficient for messages to go across the network. The choice of $\sqrt{n}$ is also used in the [121]. For some regular network topologies and by selecting a proper target range, the average message hop is actually much smaller than $\sqrt{n}$, resulting from the border determination. Therefore, the upper-bound of communication cost in the randomly directed exploration protocol is $O(\sqrt{n})$.

Because of sharing similar detection guidelines based on observer-inspector-witness model and using same cryptographic authentication primitives, the security arguments for the DHT-based protocol in Section 4.2.3, other than witness number, also apply to the randomly directed exploration protocol. Differently from the previous protocol, only neighbors of cloned node might become witnesses, thus the witness number is not a proper security measurement in this probabilistic protocol. Instead, we directly evaluate the detection probability in the following simulations to verify if the protocol satisfies the security object.

4.6 Experimental Results for Randomly Directed Exploration

We implement the randomly directed exploration protocol on the same simulation framework as the previous protocol. Since the randomly directed exploration protocol relies on a local network topology, the random graph model cannot be used for the simulations. Instead, we take the unit-disk graph as the sole network scenario. As much, we choose a constant node degree $d = 20$ and the node communication range is dynamically adjusted such that the average node degree keeps the approximate d. We select $\frac{\pi}{8}$ as the priority range of the protocol. As a result, there are average 2.5 neighbors in the priority zone of a node.

Unlike the DHT-based protocol, this protocol is sensitive to network topology; thus in order to obtain good simulation results, more than fifty network instances are randomly generated for every testing case, and each run performs one hundred rounds of detection; finally all results are averaged.

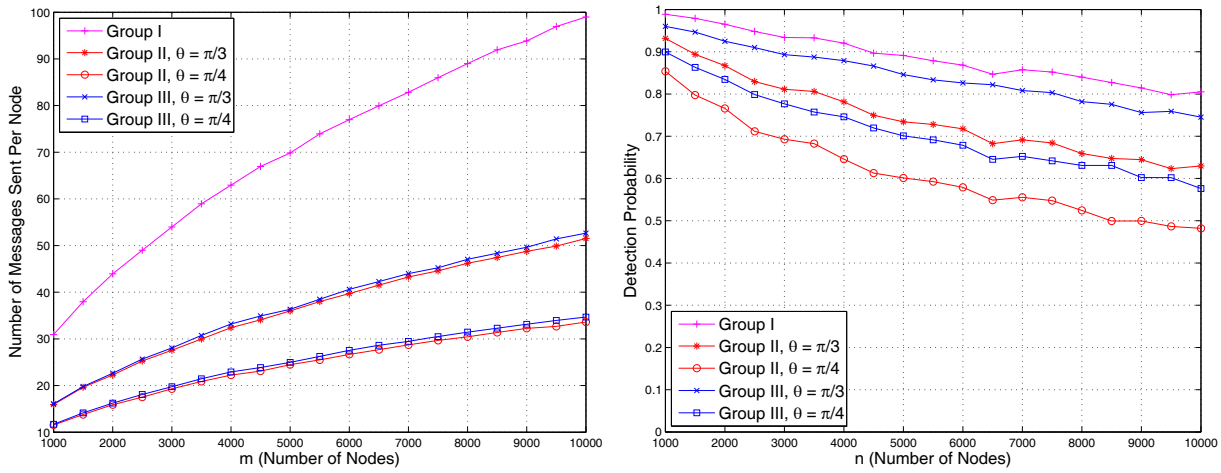
4.6.1 Performance on Different Network Sizes

We develop the first experiment which not only measures the randomly directed exploration protocol's performance on varying network sizes, but also verifies if our protocol design really fulfills the intention. In Section 4.5.1, we present three routing mechanisms and fuse them together to determine next node. In order to clearly demonstrate their actual impacts on protocol performance, we mix them as the following three groups, run simulations respectively, and compare the results for analysis.

In *group I*, only the mechanism of deterministic directed transmission is used. Consequently, all claiming messages, if not dropped by maliciously nodes, go exactly ttl hops. Next, *group II* adds network border determination into group I. As the network shape is a regular square, two different target ranges are chosen for wider and narrower border determination: $\theta = \frac{\pi}{3}$ and $\theta = \frac{\pi}{4}$. Finally, *group III* combines all the three mechanism, and indeed is our proposal. Overall, there are five testing settings: group I, group II with $\theta = \frac{\pi}{3}$, group II with $\theta = \frac{\pi}{4}$, group III with $\theta = \frac{\pi}{3}$, and group III with $\theta = \frac{\pi}{4}$.

For other parameter selections, n ranges from 1000 to 10000, message maximal hop $ttl = \sqrt{n}$, average number of claiming messages per observer $r = 1.0$, and there are two cloned nodes that are selected randomly among all nodes.

Figure 4.9 depicts the main experimental results: Figure 4.9a for the average number of messages sent per node, and Figure 4.9b for the detection probability. First of all, we can see that the protocol's communication overhead is quite impressive while its security level is



(a) Communication Cost (b) Detection Probability

Figure 4.9: Protocol performance on different network sizes, where $ttl = \sqrt{n}, r = 1.0$

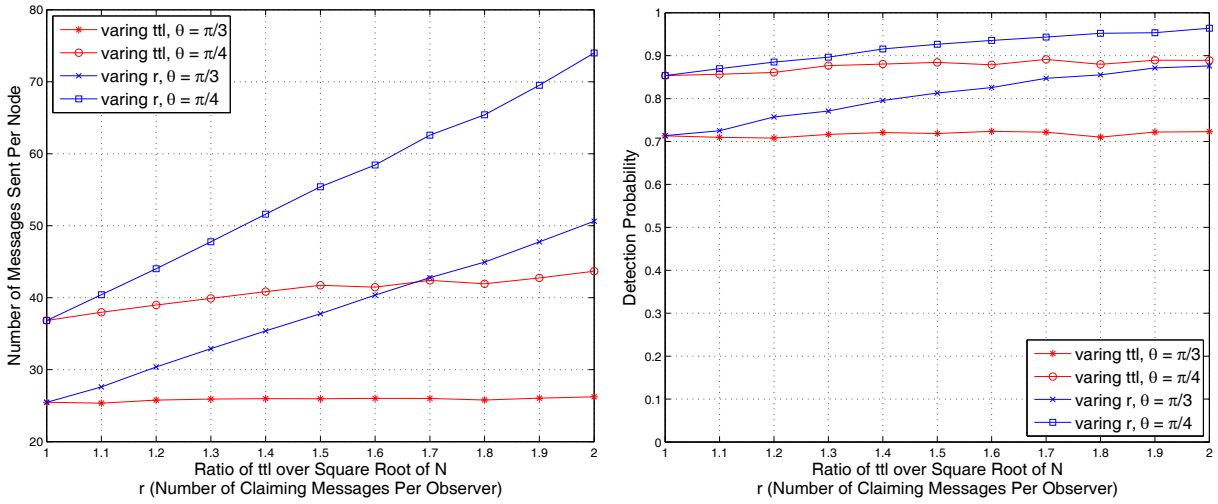
satisfactory for all the network sizes. Second, in accordance with the intuition, the comparison between group I and group II indicates that the boundary determination mechanism remarkably reduces the communication cost, while the detection probability is decreased in a moderate rate. Third, compared to group II, group III only slightly increases the average number of sent messages, but considerably improves the detected probability. This proves that the probabilistic directly mechanism fulfills the design objective, and our proposed priority zone candidates probability distribution both preserves the line characteristic of directed transmission and introduces splendid randomness on message transmission. In a word, the experimental results directly support our protocol design arguments and the protocol can properly detect node clone with outstanding communication performance.

4.6.2 Detection Probability for Multiple Cloned Nodes

If there are more than two cloned nodes, similar to the DHT-based protocol, the randomly directed exploration protocol shall detect the attack more easily. In the simulations for network size $n = 1000$ and the cloned node number δ increasing from 2 to 100, the protocol achieves 100% detection probability when $\delta \geq 3$ for $\theta = \frac{\pi}{3}$ and when $\delta \geq 4$ for $\theta = \frac{\pi}{4}$.

4.6.3 Impacts of Adjusting Parameters

We may select different values for parameters θ, ttl, and r to achieve tradeoffs between communication cost and detection probability. As the first experiment indicates, the bigger



(a) Communication Cost (b) Detection Probability

Figure 4.10: Protocol performance by adjusting ttl and r for $n = 5000$

θ, the more hops a claiming message travels, and the more likely cloned nodes are being caught. In this experiment to evaluate the effects of modifying ttl and r, , we still use the two values of θ, select $n = 5000$, and assume two randomly selected cloned nodes. First we fix $r = 1.0$, and gradually increase ttl form $\sqrt{n}$ to $2\sqrt{n}$, carrying out the simulation. Then we keep $ttl = \sqrt{n}$ and let r grow from 1.0 to 2.0. The experimental results of Figure 4.10a and Figure 4.10b show that because the border determination mechanism under a certain θ restrains the average message hop, the increasing of ttl after some threshold will has insignificant impact on protocol performance. In contrast, while the transmission overhead per node is in direct proportion with r, the detection probability is substantially improved. Therefore, r is always suitable for performance adjustment to meet different requirements on communication cost and security level.

4.6.4 Discussions

From the analysis and simulation results, randomly directed exploration protocol outperforms all other distributed detection protocols in terms of communication cost, storage requirements, while its detection probability is satisfactory, higher than that of line-selected multicast scheme in [121]. In addition, all nodes only need to know their direct neighbors information, and then inherent routing technique delivers messages in an efficient way to cover a great range of the network.

Because the three mechanisms of determining next node all rely on transmission di-

rection, the protocol performance in specific sensor networks will be related to network topologies. In general, if we can obtain a rough picture about node deployment pattern [3], we may adjust parameters θ and ttl to achieve acceptable detection probability at a cost of increasing communication overhead. For example, we may use bigger θ to tolerate irregular network outside shape, while ttl is set to the estimated value of maximal hop distance between two nodes. Of course, if a network topology is so distorted that there is no way to achieve line transmission solely based to nodes local knowledge, the randomly directed exploration detection becomes unsuitable and we may use the DHT-based protocol. In addition, if security is extremely important and it is required to certainly catch cloned nodes, the DHT-based protocol should be adopted.

4.7 Conclusions

Sensor nodes lack tamper-resistant hardware and are subject to the node clone attack. In this chapter, we have presented two distributed detection protocols: one is based on distributed hash table which forms a Chord overlay network and provides the key-based routing, caching, and checking facilities for clone detection, and the other uses probabilistic directed technique to achieve efficient communication overhead for satisfactory detection probability. While the DHT-based protocol provides high security level for all kinds of sensor networks by one deterministic witness and additional memory-efficient, probabilistic witnesses, the randomly directed exploration presents outstanding communication performance, minimal storage consumption for dense sensor networks.

[3]We stress that nodes do not require to know the network topology.

Chapter 5

Data Aggregation Integrity Based on Homomorphic Primitives

Many previous secure aggregation schemes in wireless sensor networks were devoted to data confidentiality, while few practical, efficient message authentication schemes accommodating valid data aggregation have been proposed. In fact, resembling general security cases in other fields, message integrity might be one of the most important objectives in sensor networks, because invalid, undetected data alteration will cause very dangerous outcomes. For example, in a sensor network monitoring battlefield and providing vital decision-supported information, it would be devastating that enemies are able to manipulate the sensor network outputs and trick users into accepting misleading data. In fact, data aggregation reduces the degree of difficulty for adversaries faking false reports, as the adversaries do not need to manipulate the readings of majority of sensor nodes, which is difficult and impractical; instead, by compromising several nodes close to the base station, they can easily alter the aggregated results to whatever match their interests. Therefore, the verifiable integrity of aggregated messages is an imperative security objective in sensor network data aggregation, and should be fulfilled by corresponding protocols. As one of principles for the cryptography primitives, those schemes that merely provide data confidentiality cannot serve as message authentication mechanisms. Message integrity objective should be addressed by specific detection or authentication schemes.

There are three kinds of message verification approaches for data aggregation: retroactive detection, abnormality-based detection, and cryptographic integrity primitives. Generally speaking, retroactive detection approaches, which involve considerable communication/interaction among the base station and sensor nodes to verify messages integrity, are not satisfactory because their costly performance penalty directly violates the intent of data aggregation. One may argue that the predictable data distribution can be used as a gauge to analyze and detect the abnormality of aggregated results; but the false rates are

generally too high to be practical, and thus it is not a dependable solution. Consequently, schemes based on solid cryptographic primitives are usually desired. Unfortunately, conventional cryptographic integrity primitives, such as message authentication code (MAC) and signature, are not compatible with data aggregation scenarios.

Based on new cryptographic homomorphic primitives [8, 98, 70], we propose three secure aggregation schemes that provide provably secure message integrity. The first one is a homomorphic MAC scheme for data aggregation, which is a revised version of the homomorphic MAC proposal on secure network coding application in [8]. This homomorphic MAC scheme, other than revisions to fit data aggregation scenarios, achieves a little bit performance improvement, as we observe and then remove an unnecessary step in the original scheme. The homomorphic MAC scheme is computation- and communication-efficient, but with one inherent restriction: all data-collecting nodes share one global key with the base station. The assumption that all those nodes are tamper-proof might be too strong to be realistic in many sensor network applications. In order to overcome this drawback, we further propose two secure aggregation schemes based on homomorphic hash [98, 70], at the expense of increasing communication and computation costs. One is to combine homomorphic hash with aggregate MAC [92], in which every node shares a different key with the base station, while the other is associated with identity-based aggregate signature [72], which enables intermediate nodes to verify the authenticity of messages. The proposed three protocols present different tradeoffs between computation, communication, security and can fit a wide variety of application areas.

The chapter is organized as follows. First, we introduce the data aggregation network settings, state security objective and implications, discuss homomorphic primitives, and define homomorphic MAC as well as homomorphic hash in Section 5.1. Then we present a concrete homomorphic MAC construction for secure aggregation integrity in Section 5.2. Afterwards, two secure data aggregation schemes based on homomorphic hash are proposed in Section 5.3. Section 5.4 concludes the work of this chapter. An earlier version of the three secure integrity scheme has been published in [104].

5.1 Background

5.1.1 Network Settings

We consider a sensor network that consists of n sensor nodes which are highly sensitive of energy consumption, and a base station that is only concerned about the statistical results, mainly mean and variance. Thus a data aggregation mechanism is implemented in the sensor network.

Loose Time Synchronization

Since loose time synchronization among sensor nodes is indispensable for efficient message aggregation and the sensor network is under attacks, it is assumed that there is a secure time synchronization scheme [127] available in the network. We do not explore a specific secure time synchronization selection because it is independent and relatively irrelevant.

Report and Its Identifier

At a designated time, the sensor network outputs a *report*, which is an overall aggregated result for a task and is *uniquely* identified by a report's identifier rid. The report's identifier may be the task description combined with the reporting time. It is clear that all sensor nodes should have an agreement on the report identifier specification and know how to correctly generate rid. Otherwise, nodes cannot distinguish messages of different kinds and data aggregation cannot be properly performed.

Roles Definitions

There are three kinds of roles in the sensor network:

- A *contributor* that collects environmental readings and generate a *raw message*.
- An *aggregator* that aggregates all messages that it received plus possibly its own raw message and then produces an *aggregated message*.
- A *verifier* that verifies the authentication of messages it received.

A node may play some of or all the three roles, while the base station is definitely a verifier.

Aggregation Mode

The data are aggregated though the network, and the base station eventually retrieves an aggregated result (i.e., the report). In order to produce the mean of a measurement, it suffices for the base station to retrieve the sum of the samples and the number of contributors. If the variance is also desired, the contributors should provide the squares of their readings and the aggregators accordingly merge the squares. By the average value, the number of contributors and the sum of the squares, one can readily calculate the variance as a basic statistical equation. In other words, we only need to consider an

additive aggregation. For the sake of simplicity, we assume that the sensor network is organized as a tree structure rooted on the base station, though our proposed schemes fit into any kind of data additive aggregation architecture.

In addition, to support advanced aggregation requirements, the concept of *weight* is introduced. Specifically, we allow that the measurements of different nodes have different weights for their contributions to the final report. In most cases, node weights are uniform; when distinct weights are required, we assume that aggregators and the base station are aware of the weights of messages contributors, either via an established agreement, or from explicit indications attached to messages.

Application Scenario

As a typical application scenario of the network settings, a sensor network is employed to routinely detect environmental information, e.g., temperature, humidity, radiation. Every node senses data in a hourly interval, and submits the results on a daily basis. For example, at two o'clock every day, starting from all leaf nodes, messages are transmitted and aggregated over a spanning tree.

5.1.2 Security Objective and Implications

The primary objective of our proposals is to provide the message integrity for data aggregation in a cryptographically secure manner, thus an authentication segment that facilitates verification shall be appended to a message. Generally speaking, it is impossible for a verifier to validate the integrity of an aggregated message without the knowledge of its contributors. This is because if contributors use different keys, the verifier certainly needs knows who those contributors are before using those keys in the verification stage; if a global key is employed and a verifier cannot retrieve contributors of messages, an adversary may easily construct a malicious message to pass the integrity verification by aggregating a single message from one contributor many times, say b times, which is indistinguishable with an aggregated message resulting from b legitimate contributors. In other words, data origin authentication is an inherent requirement for data aggregation integrity.

The simplest way of indicating data origin is to attach the list of contributors to a message. To avoid the communication cost in this approach, we may utilize a mechanism that allows a verifier to implicitly obtain the contributor list, such as derivation from the network topology. This is pretty realistic for the base station as the ultimate verifier. In a case that a verifier is capable of identifying all potential contributors, of which only a small fraction do not really participate in a message contribution, a list of exclusive nodes rather than the contributors may be appended to the message. Anyway, we henceforth assume

that an aggregator knows the appropriate weights to aggregate messages, and a verifier of a message can obtain its contributors and corresponding weights. When we discuss a scheme's communication cost, we do not consider the payload from contributor lists and weights, because, as we argued, there might be mechanisms to avoid it, or it is inevitable for message authentication.

5.1.3 Homomorphic Primitives

Homomorphic property in cryptographic operations may are very useful in a variety of applications, and thus stimulates research on homomorphic primitives, namely homomorphic encryption, homomorphic MAC, homomorphic hash, and homomorphic signature. Homomorphic encryption [63], in which a user without a decryption key can perform algebraic operations on ciphertext to achieve designated transformation results on the corresponding plaintext, has been studied for decades, and recently, an outstanding result, fully homomorphic encryption [71], was proposed, which allows arbitrary operations on ciphertext (and equivalently on plaintext). Even though the only two fully homomorphic encryption schemes [71, 145] by now have not provided competitive performance for most applications, they do reveal a perspective on a powerful, widely demanded technique and we expect that practical schemes will eventually emerge. Those homomorphic encryption schemes shall provide a solid foundation for data confidentiality of aggregated messages. As for homomorphic signature, current schemes [90, 32] are mainly aimed at one-sender many-recipients secure multi-cast scenarios, with costly computation overhead (compare to symmetric primitives), thus they may not be suitable for secure data aggregation integrity of WSNs. In contrast, homomorphic MAC and homomorphic hash can be effectively used to construct message integrity schemes of supporting additive aggregation with weights. Formally, homomorphic MAC and homomorphic hash are defined as follows.

Homomorphic MAC

Definition 5.1 (Homomorphic MAC [8])**.** *A homomorphic MAC should satisfy the following properties:*

1. Homomorphism. *Given two (message, tag) pairs* $(\boldsymbol{m}_1, t_1)$ *and* $(\boldsymbol{m}_2, t_2)$*, anyone can create a valid tag* t_a *for an aggregated message* $\boldsymbol{m}_a = w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2$ *for any scales* w_1, w_2 *as weights. Typically,* $t_a = w_1t_1 + w_2t_2$*.*

2. Security against Chosen Message Attack. *Even under a chosen message attack, in which an adversary is allowed to query tags of polynomial number of messages, it is still infeasible for the adversary to create a valid tag for a message other than a linear combination of some previously queried messages.*

*A homomorphic MAC consists of three probabilistic, polynomial-time algorithms (*Sign*,* Aggregate*,* Verify*)*

- $t_u = \mathsf{Sign}(k, \mathsf{rid}, \boldsymbol{m}_u, \mathsf{id}_u)$*: node* u *with identity* id_u*, as a contributor of a raw message* $\boldsymbol{m}_u$ *regarding report* rid*, computes a tag* t_u *for* $\boldsymbol{m}_u$ *using* k *as the key.*
- $t = \mathsf{Aggregate}((\boldsymbol{m}_1, t_1, w_1), \ldots, (\boldsymbol{m}_j, t_j, w_j))$*: an aggregator implements the homomorphic property for message-tag pairs in the absence of key* k*, that is, generates a tag* t *for the aggregated message* $\boldsymbol{m} = \sum_{i=1}^{j} w_i \boldsymbol{m}_i$
- $\mathsf{Verify}(k, \mathsf{rid}, \boldsymbol{m}, t)$*: a verifier verifies the integrity of message* $\boldsymbol{m}$ *regarding report* rid *by key* k *and tag* t*.*

The homomorphic MAC scheme is first defined and proposed in [8], intended to provide secure network coding. The definition above is equivalent to that in [8], with emphasis on the data aggregation.

Homomorphic Hash

Definition 5.2 (Homomorphic Hash [98, 70])**.** *A homomorphic hash function* H *is a hash function satisfying:*

1. Homomorphism. *For any two messages* $\boldsymbol{m}_1$*,* $\boldsymbol{m}_2$ *and scalars* w_1*,* w_2*, it holds that* $H(w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2) = H(\boldsymbol{m}_1)^{w_1} H(\boldsymbol{m}_2)^{w_2}$*.* [1]
2. Collision Resistance. *There is no probabilistic polynomial-time adversary capable of forging* $(\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3, w_1, w_2)$ *satisfying both* $\boldsymbol{m}_3 \neq w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2$ *and* $H(\boldsymbol{m}_3) = H(\boldsymbol{m}_1)^{w_1} H(\boldsymbol{m}_2)^{w_2}$*.*

The homomorphic hash can be used in many applications, such as secure network coding [70], secure peer-to-peer content distribution using erasure codes [98].

5.2 Secure Aggregation with Homomorphic MAC

Our first proposal is a specific homomorphic MAC scheme that fully complies with Definition 5.1. Basically, the scheme is a revised version of the homomorphic MAC for network coding proposed by Agrawal and Boneh (AB Scheme) [8].

[1] Intuitively, the homomorphic equation should be $H(w_1\boldsymbol{m}_1 + w_2\boldsymbol{m}_2) = w_1 H(\boldsymbol{m}_1) + w_2 H(\boldsymbol{m}_2)$. In fact, that just uses a different notation on group operation and essentially they are equivalent.

5.2.1 Scheme Description

To formally present our schemes, message $\boldsymbol{m}$ is formed as d segments of l bits. Let $q = 2^l$, then the message space is $\mathbb{F}_q^d$. In other words, message $\boldsymbol{m}$ can be represented as a vector of d segments: $(m_1, m_2, \cdots, m_d)$, where $m_i \in \mathbb{F}_q, i = 1, 2, \cdots, d$. As the additive operation is over finite field $\mathbb{F}_q$, q should be greater than the bound of the desired data sum. We stress that this is also an inherent requirement in the data aggregation.

To generate and verify tags, all contributors and verifiers share one global MAC key that consists of (k_1, k_2). Naturally, those nodes should be tamper-proof to protect the protocol security. Let $\mathcal{K}_1$ and $\mathcal{K}_2$ denote the key spaces of k_1 and k_2 respectively, $\mathcal{I}$ denote the space of node identities, and $\mathcal{R}$ denote the space of report identifiers. Two pseudo random functions are required: $R_1 : \mathcal{K}_1 \to \mathbb{F}_q^d$ and $R_2 : (\mathcal{K}_2 \times \mathcal{R} \times \mathcal{I}) \to \mathbb{F}_q$.

The three algorithms (Sign, Aggregate, Verify) are given as follows.

- $\mathsf{Sign}(k, \mathsf{rid}, \boldsymbol{m}_u, \mathsf{id}_u)$, by node u as a contributor
  1. $\boldsymbol{a} = R_1(k_1) \in \mathbb{F}_q^d$.
  2. $b_u = R_2(k_2, \mathsf{rid}, \mathsf{id}_u) \in \mathbb{F}_q$.
  3. $t_u = \boldsymbol{a} \cdot \boldsymbol{m}_u + b_u \in \mathbb{F}_q$, where $\cdot$ stands for the inner product of two vectors $\boldsymbol{a}$ and $\boldsymbol{m}_i$ over finite field $\mathbb{F}_q$, that is, $\boldsymbol{a} \cdot \boldsymbol{m}_u$ is equal to $a_1 m_{u,1} + a_2 m_{u,2} + \cdots + a_d m_{u,d}$ mod q.
- $\mathsf{Aggregate}((\boldsymbol{m}_1, t_1, w_1), \ldots, (\boldsymbol{m}_j, t_j, w_j))$, by an aggregator
  1. $\boldsymbol{m} = \sum_{i=1}^{j} w_i \boldsymbol{m}_i \in \mathbb{F}_q^d$, in which the additive operation is over $\mathbb{F}_q$.
  2. $t = \sum_{i=1}^{j} w_i t_i \in \mathbb{F}_q$.
- $\mathsf{Verify}(k, \mathsf{rid}, \boldsymbol{m}, t)$, by a verifier with the knowledge of contributors identities and weights
  1. $\boldsymbol{a} = R_1(k_1) \in \mathbb{F}_q^d$.
  2. $b = \sum_{i=1}^{j} [w_i R_2(k_2, \mathsf{rid}, \mathsf{id}_i)] \in \mathbb{F}_q$.
  3. if $\boldsymbol{a} \cdot \boldsymbol{m} + b = t$ outputs "ACCEPT"; otherwise outputs "REJECT".

5.2.2 Discussions and Comparisons

By the same reduction proof of Theorem 2 in [8], this scheme is probably secure against chosen message attack based on the pseudorandomness of R_1 and R_2. Since the tag size is l-bit, in order to achieve 80-bit security level, l should not be less than 80.

To support secure network coding, the space $\mathcal{I}$ in the AB scheme [8] is $\mathbb{F}_q^c$, albeit id_i is a vector base identifier, rather than a node id, and c is the number of vector base. Since every message in the network coding should include a vector in $\mathbb{F}_q^c$ to indicate the combination coefficients of c vector bases, which are analogue to weights in the data aggregation, usually $q = 2^8$ is recommended (as in the AB Scheme) to save communication cost while maintaining high success decoding probability for random network coding. Such a small q, however, undermines the security level, as the tag size would be 8-bit and an adversary can fake a message's tag at least with probability 1/256. Fortunately, the data aggregation does not suffer that limitation—the weights are not randomly chosen by aggregators. Therefore, we can safely use $q \geq 2^{80}$.

In addition, the AB homomorphic MAC scheme specifies $R_1 : \mathcal{K}_1 \rightarrow \mathbb{F}_q^{d+c}$, $\boldsymbol{a} = R_1(k_1) \in \mathbb{F}_q^{d+c}$, and then $t_u = \boldsymbol{a} \cdot (\boldsymbol{m}_u || \mathsf{id}_u) + b$. We observe that the occurrence of id_u in $(\boldsymbol{a} \cdot (\boldsymbol{m}_i || \mathsf{id}_u))$ is unnecessary and then it is removed in our revision because id_u has been used in the computation of $b = R_2(k_2, \mathsf{rid}, \mathsf{id}_u)$. This modification slightly improves the computation performance and can apply to both network coding and data aggregation scenarios.

We notice that our proposed scheme has a similar structure to the data aggregation MAC scheme proposed by Castelluccia *et al.* (CCMT scheme) [39]. In their scheme, the space of message $\boldsymbol{m}$ is limited to $\mathbb{F}_q$, which means that the tag is as long as the maximal length of messages. This approach violates a principle on MAC that a MAC scheme should support arbitrary length of message and output short, fixed length of tags. Admittedly, the length of messages in our scheme has to be determined beforehand, but it is a basic requirement for data aggregation. In addition, the CCMT scheme does not supply a reduction security proof; Theory 2 in [39] pertaining to the scheme security is more like an argument than a proof. Nonetheless, the CCMT scheme provides a necessary integrity scheme for data aggregation, and our homomorphic MAC scheme can be thought as the combination of the CCMT scheme and the AB scheme.

The security of the proposed MAC scheme relies on the pseudorandomness of R_1 and R_2. In principle, all provably secure pseudorandom generators are public-key based[2], involving heavy computation. As a widely employed method, we may use AES [52] to implement R_1 and R_2. In this way, the proposed scheme is very computationally efficient, and the key lengths of k_1 and k_2 are 128-bit. On the other hand, an 80-bit tag will suffice to allow a verifier to check the authenticity of an aggregated message, which presents the optimal communication overhead. One inherent drawback in homomorphic MACs is that one single MAC key is shared by all contributors and verifiers. If sensor nodes are not tamper-proof and one of them is compromised by an adversary, the whole system security is breached.

[2]A public-key based approach does not necessarily indicate that it involves public/private keys; instead, it implies that the approach employs typical public-key cryptosystem operations, e.g., exponentiation over a big group.

5.3 Integrity Schemes Based on Homomorphic Hash

In order to overcome the drawback of one global MAC key in the previous scheme, we propose two schemes based on homomorphic hash.

5.3.1 Constructions of Homomorphic Hash Function

The first step is to find a homomorphic hash function suitable for sensor networks. At present, there are only two homomorphic hash functions: one is based on the hardness of discrete logarithm [98], and the other is based on the intractability of integer factorization [70].

Discrete Logarithm [98]

Let $\mathbb{G}$ be a cyclic group of prime order p in which the discrete logarithm problem is hard, and the public parameters contain a description of $\mathbb{G}$ and d random generators $g_1, g_2. \cdots, g_d \in \mathbb{G}$. Then a homomorphic hash on message $\boldsymbol{m} = (m_1, m_2, \cdots, m_d) \in \mathbb{Z}_p^d$ can be constructed by

$$H(\boldsymbol{m}) \stackrel{\text{def}}{=} \prod_{i=1}^{d} g_i^{m_i} \ . \tag{5.1}$$

It is easy to verify that the homomorphic property is satisfied in this construction, and the collision resistance is guaranteed by the hardness of the discrete logarithm problem in $\mathbb{G}$.

Integer Factorization [70]

Let N be the product of two safe primes[3] so that the group $\mathbb{Q}_N$ of quadratic residues modulo n is cyclic, and let $g_1, g_2, \cdots, g_d$ be generators of $\mathbb{Q}_N$. Then a homomorphic hash on message $\boldsymbol{m} = (m_1, m_2, \cdots, m_d) \in \mathbb{Z}^d$ can be constructed by

$$H_N(\boldsymbol{m}) \stackrel{\text{def}}{=} \prod_{i=1}^{d} g_i^{m_i} \mod N \ . \tag{5.2}$$

Finding a collision is computationally equivalent to factoring N, which is generally believed to be intractable.

[3] A prime number p is a safe prime if $(p-1)/2$ is also a prime.

Comparison

The homomorphic hash function (5.2) can use the form of $H_N(\boldsymbol{m}) = 2^{\boldsymbol{m}} \mod N$ by choosing a proper N such that 2 is a generator of $\mathbb{Q}_N$ and converting message $\boldsymbol{m}$ to one single integer. Subsequently, it eliminates the requirements of multi-generators and presents some computational advantage over hash function (5.1) by fast exponentiation. However, its hash value size is the same as the size of N, which is at least 1024-bit to provide 80-bit security, while by using elliptic curve cryptography (ECC), the hash value size of function (5.1) can be approximately as low as 160-bit. Moreover, the practicability of implementing ECC in low-cost sensor nodes has been successfully demonstrated in [109, 141]. Therefore, we select function (5.1) as the homomorphic hash for secure data aggregation integrity in WSNs.

5.3.2 Aggregation Integrity by Homomorphic Hash

Since we choose the homomorphic hash function (5.1), the message space is $\mathbb{F}_p^d$, where p is a prime number and $p \geq 2^{160}$ for 80-bit security. For a raw message $\boldsymbol{m}_i$, node i computes a raw hash value $h_i = H(\boldsymbol{m}_i)$, and uses a mechanism to sign h_i, which will be specified later, in a way that allows verifiers to check the authenticity of h_i. When a verifier receives an aggregated message $\boldsymbol{m} = \sum_{i=1}^{j}(w_j \boldsymbol{m}_i)$ along with j pairs of (raw hash value, weight) (h_i, w_i), it first determines whether the hash values are legitimate, and then verifies the message's integrity by checking whether

$$\prod_{i=1}^{j} h_i^{w_i} \stackrel{?}{=} H(\boldsymbol{m}) \ .$$

This scheme is proven secure in the standard model via reductionist from the discrete logarithm problem [98, 32], when raw hash values are authenticated by a secure mechanism. In the following two subsections, we describe two communication-efficient mechanisms to authenticate h_i.

5.3.3 Authentication by Aggregate MAC

Aggregate MAC [92] presents the property that multiple MAC tags, computed by different contributors on multiple raw hash values, can be aggregated into a single tag that is verified by a verifier who shares a distinct key with each contributor. The construction of aggregate MAC has been long known. In fact, an aggregate MAC which is provably secure [92] can be constructed from essentially any standard message authentication code as follows.

For simplicity, we assume that the base station is the sole verifier. Let k_i be the symmetric key shared by node i and the base station, Mac be a standard deterministic MAC, for example: CBC-MAC [21], HMAC [20]. To authenticate a raw hash value h_i, node i generates a tag: $t_i = \mathsf{Mac}_{k_i}(\mathsf{rid}, h_i)$. Any aggregator can aggregate j tags by simply computing the XOR of all the tag values: $t = \bigoplus_{i=1}^{j} t_i$. Then the base station uses the aggregate tag t to verify the authenticity of all raw hash values by checking whether

$$t \stackrel{?}{=} \bigoplus_{i=1}^{j} \mathsf{Mac}_{k_i}(\mathsf{rid}, h_i) \ .$$

5.3.4 Authentication by Identity-Based Aggregate Signature

Aggregate MACs, like all other symmetric-key MACs, demand verifiers to comprehend contributors keys. In many circumstances, it would be much appreciated that all intermediate nodes can verify the authenticity of raw hash values (and then aggregated messages). In terms of communication cost, the best scheme providing such a property is an identity-based aggregate signature (IBAS), in which different raw hash values produced by many different contributors, whose public keys are their identities, can be authenticated by one single aggregate signature.

As far as we know, there are three IBAS schemes which are provably secure: GR scheme [72], BN scheme [22], and BGOY scheme [30]. The BN scheme [22] requires interactions of all signers, and the BGOY scheme demands a sequential signature aggregation procedure; thus both are not suitable for secure aggregation in WSNs. One presumably too strong assumption in the GR scheme [22] is that all signers must use a same unique string during the period of signing, which, fortunately, is not a problem at all in the sensor network aggregation application, because a unique rid for every report is known to all nodes.

GR Paring-Based IBAS Scheme [72]

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of some large prime order q which support a bilinear mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. That is, $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$ for all $Q, R \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$. The GR IBAS scheme works as follows.

- *Setup*: To set up the scheme, a private key generator (PKG)
    1. generates groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q and an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

2. chooses an arbitrary generator $P \in \mathbb{G}_1$.
3. picks a random $s \in \mathbb{Z}/q\mathbb{Z}$ as the master key of PKG and sets $Q = sP$.
4. chooses three cryptographic hash functions $H_1, H_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}/q\mathbb{Z}$.

- *Private key generation*: Node i receives from the PKG the values of $sP_{i,\alpha}$ as its private key for $\alpha \in \{0,1\}$, where $P_{i,\alpha} = H_1(\mathsf{id}_i, \alpha) \in \mathbb{G}_1$.
- *Signing*: To sign $\boldsymbol{h}_i$, node i
  1. computes $P_{\mathsf{rid}} = H_2(\mathsf{rid}) \in \mathbb{G}_1$.
  2. calculates $c_i = H_3(\boldsymbol{h}_i, \mathsf{id}_i, \mathsf{rid}) \in \mathbb{Z}/q\mathbb{Z}$.
  3. generates random $r_i \in \mathbb{Z}/q\mathbb{Z}$.
  4. computes signature (S_i, T_i), where $S_i = r_i P_{\mathsf{rid}} + sP_{i,0} + c_i sP_{i,1}$ and $T_i = r_i P$.
- *Signature Aggregation*: Signatures (S_i, T_i) for $1 \leq i \leq j$ can be aggregated into (S, T), where $S = \sum_{i=1}^{j} S_i$, and $T = \sum_{i=1}^{j} T_i$.
- *Verification*: Any node can verify the signature by checking whether

$$\hat{e}(S, P) \stackrel{?}{=} \hat{e}(T, P_{\mathsf{rid}})\ \hat{e}(Q, \sum_{i=0}^{j} P_{i,0} + \sum_{i=0}^{j} c_i P_{i,1}).$$

This scheme is proven secure in the random oracle model, on the assumption of hardness of the computational Diffie-Hellman problem.

Generally speaking, paring is a highly computation-intense operation and more costly than ordinary public key based operations. Considering the fact that identity-based schemes eliminate the cost of transmitting nodes public keys and most of practical identity-based encryptions are paring-based, the use of the GR paring-based IBAS scheme in the secure WSN data aggregation is justifiable. In addition, the work on TinyPBC [118] which implements and measures paring operations in typical sensor nodes hardware gives an affirmative answer to the question of whether paring is feasible in the sensor networks, albeit their paring implementation is understandably slow.

5.3.5 Discussions

To verify the integrity of an aggregated message, a verifier should retrieve the raw hash values of the contributors, which constitutes considerable communication payload and is an

instinctive downside for homomorphic-hash-based approaches. When the message size in an application does not exceed the homomorphic hash result size (160-bit typically), then the homomorphic hash is redundant, and directly applying aggregate MAC/signature to raw messages is preferred. If the message size is substantially greater than the hash value size, which is quite common for WSNs, then using homomorphic hash would significantly reduce the communication cost, as in the application scenario described in Section 5.1.1.

To provide 80-bit security, the signature of the GR scheme is roughly 320-bit, while a typical aggregate MAC tag is 80-bit. The third scheme, which combines homomorphic hash function (5.1) with the GR identity-based aggregate signature, provides the most promising security for data aggregation integrity. Since all intermediate nodes are capable of verifying the integrity of (raw or aggregated) messages in that scheme, a malicious node that tries to inject invalid messages into the sensor network can be easily caught. It is worth noticing that this is achieved at the computational cost of paring operation by intermediate nodes. By contrast, the second scheme (with aggregate MAC) does not require paring operation and is useful in practice. If detecting an invalid (message, tag) pair in the second scheme, the base station can require the corresponding child to submit its aggregation record, and then interacts with grandchildren until reaching leaf nodes. In this fashion, the base station can determine which nodes should be responsible for faking messages and then expels them from the network.

5.4 Conclusions

In this chapter, we have presented three secure aggregation schemes that provide provably secure message integrity with different tradeoffs between computation cost, communication payload, and security assumptions. The first proposal is a concrete homomorphic MAC scheme for sensor network data aggregation integrity, and the other two are combining homomorphic hash with aggregate MAC and identity-based aggregate signature respectively. We have detailed on the selections and constructions of those three cryptographic primitives and discussed their practicability in wireless sensor networks.

Chapter 6

Data Aggregation with Secure Bloom Filter

In this chapter, we propose a secure data aggregation protocol which fits a specific but popular class of aggregations in wireless sensor networks. Unlike most previous approaches, which are aimed to provide security mechanisms for ordinary aggregation operations, our proposal, in the first place, is an effective and succinct aggregation protocol, which is equipped with built-in security mechanisms, fulfilling the fundamental security purpose of preventing compromised nodes as well as outside adversaries from harming network aggregated results. The security technique backing up our protocol is to combine HMAC (Hash-based Message Authentication Code), a conventional and provably secure cryptographic primitive, with the popular, space-efficient data structure Bloom filter [27]. The resulting data structure is defined as secure Bloom filter, and its security implications are addressed at length. Then we use it to construct a secure aggregation protocol for sensor networks. The theoretical analysis and extensive simulation results demonstrate that the proposed protocol presents remarkable performance on security, communication cost, and energy consumption balance degree among sensor nodes.

This chapter is structured as follows. First, we describe the application scenarios, and define network and adversary models in Section 6.1. Then, we present secure Bloom filter as the technical core at length in Section 6.2. Afterwards, we provide the proposed secure aggregation protocol and corresponding analysis in Section 6.3. The simulation design and results are elaborated in Section 6.4. Lastly, we conclude this work in Section 6.5. An earlier version of the proposed secure data aggregation scheme appeared in [105].

6.1 Preliminaries

6.1.1 Network and Adversary Models

Homogeneous Sensor Network

We consider an ordinary, homogeneous sensor network that consists of n low-cost sensor nodes which are resource-constrained in terms of computation, communication, storage, and power supply. Those sensor nodes are especially sensitive to energy consumption and thus data aggregation is desired. As a rule, there exists a central, powerful base station that is responsible for collecting sensor network results.

Adversary Model

Sensor nodes are not equipped with tamper-proof hardware, and the adversary is capable of compromising and fully controlling arbitrary number of sensor nodes, but we assume that the base station is immune to all physical attacks and it is trustworthy. Moreover, the adversary can eavesdrop and alter any messages from integrity nodes. In a word, the adversary is granted the full-scale attack capacity against the sensor network except the base station.

Multi-hop Data Aggregation

In our network model, sensor nodes may be densely deployed, or nodes may only have a few neighbors. Intuitively, the network size n should not be small and most of communications between nodes and the base station are multi-hop; otherwise, there is no point for introduction of data aggregation. Our protocol does not rely on specific data aggregation modes; any effective aggregation organization, no matter it is cluster-, chain-, tree-, or grid-based [130], suffices for the proposed protocol. We do not address the problem how to construct a specific aggregation organization, because it is quite irrelevant to our proposal.

Similar to other data aggregation schemes, we define two kinds of roles for sensor nodes:

- A *contributor* that detects events and generates *raw messages*.
- An *aggregator* that aggregates all messages that it received plus possibly its own raw messages [1] and then forwards *aggregated messages*.

Messages are aggregated though the network; and the base station eventually retrieves a number of aggregated messages, which are defined as *reports*.

[1] In such a case, the aggregator is also a contributor.

Base Station Authenticated Broadcast

It is helpful for final data analysis if the base station is aware of all nodes' positions; but it is not mandatory for the proposed protocol. We assume that there is a secure broadcast message authentication scheme available for the base station, which is pretty natural for securing sensor networks. Otherwise, it is infeasible for the base station to issue reliable, integrity commands to sensor nodes and manage the sensor network. This simple requirement can be easily satisfied for wireless sensor networks. We may use the popular, lightweight, symmetric-key hash-chain-based μTESLA authentication protocol [124] for this purpose. Alternatively, all nodes are preloaded with the base station's public key, and then messages from the base station are signed and verified through classical signature schemes. For simplicity, we ignore specific choices on broadcast message authentication.

Simplistic Key Predistribution Requirement

Every node shares a distinct long-term key with the base station. Those keys are predistributed into sensor nodes prior to deployment. No pairwise keys between sensor nodes or any other kinds of keys are demanded. Sensor nodes are equipped with a cryptographic hash function. Those constitute all security primitive requirements of the proposed protocol. As a general method to reduce its key storage, the base station may employ a pseudorandom generator (PRG) along with one main key, which generates a node's key using the node's ID and the main key as the input of PRG.

6.1.2 Application Scenarios

As a typical data aggregation application scenario of our proposal, a wireless sensor network is employed in a hostile environment to detect a number of *predefined, basic* events with two states: ON and OFF. For most of time, these events stay OFF: they did not occur. When sensor nodes detect an event taking place, they notify the base station about their discovery. Since there are many nodes simultaneously detecting same events, data aggregation is desired to reduce communication costs and to balance nodes energy consumption. The base station may be tolerant of a little inaccuracy on final aggregated results, as long as the majority of nodes event reports are recovered such that it is able to figure out that specific events happened in a certain region.

Even though this scenario looks like relatively simplistic, it is fairly popular for wireless sensor network applications. For instance, an intrusion-detection sensor network [112] successfully triggers an alarm for a perimeter intrusion event. Moreover, sophisticated tasks can be decomposed into many basic events. In such cases, sensor nodes are responsible for simple event detection, which significantly reduces their production expenditure, while the

base station performs advanced data analysis. Other application scenarios are voting and counting mechanisms for sensor networks. For example, in many secure sensor network protocols, malicious nodes are determined by majority results from nodes voting; and under many circumstances, the base station may need to frequently count the number of nodes that match some standards.

6.1.3 Security Objectives

Data confidentiality is not a concern for the application scenarios at which our proposal is targeted. For simple event detection networks, if an adversary is physically close to sensor nodes, he can easily sense those simple events on his own. Certainly, some extent of message privacy is appreciated. Barely given a message and without any other knowledge, should an adversary not tell which nodes have contributed to the message and what events are reported.

Generally speaking, the main security goal of secure aggregation is to strictly restrain adversary's influence only on those compromised nodes. First, even with collusion of all other sensor nodes, an adversary cannot fake a specific node's event report with more than a low, theoretically-limited probability. Second, other than compromised nodes' input on an event report, the adversary is only able to affect accumulative results in a trivial way. Otherwise, the scheme should detect the abnormality and discard related results. If intermediate nodes manipulate or drop integrity nodes' event-report messages, the secure aggregation scheme should be able to effectively detect the attacks.

6.1.4 Native Solution without Data Aggregation

There exists a straightforward, native solution to fulfilling the aforementioned security objectives, which does not engage data aggregation and then incurs heavy communication cost. When node u detects an event e, it generates a raw message composed of $\mathsf{Mac}(K_u, e)$ and sends to the base station, where K_u is the symmetric key shared by node u and the base station, and Mac denotes a tag by a standard deterministic MAC. When the base station receives a message, it may try all combinations to figure out which node claims which event. In order to prevent message-dropping by malicious nodes, the base station broadcasts all messages it received through the broadcast authentication mechanism. If a node does not obtain the authenticated acknowledgment of its raw message in a reasonable time slot, it will notify the base station about a potential of the attack.

It may seem tempting to use aggregate MAC [92], which aggregates multiple MAC tags by different contributors into a single tag that can be verified by the base station, to reduce the communication payload of the native scheme. Unfortunately, in such an

application, an aggregate tag has to list all contributors and associated events such that the base station can verify the tag. Consequently, the size of an aggregated message is still proportional to the number of contributors and events. Moreover, it violates the message privacy objective.

6.2 Secure Bloom Filter

The technical core of our proposal is a variant of Bloom filter, for which we coin the term *secure Bloom filter*.

6.2.1 Foundation: Bloom Filter

Bloom filter, conceived by Bloom [27], is a space-efficient probabilistic data structure that succinctly represents a set in order to support membership queries. Due to its distinguished space advantages and excellent distributed properties, Bloom filter has been widely used in numerous areas, such as web cache sharing [61] and distributed storage system [43].

Typical Implementation of Bloom Filter

Typically, a Bloom filter is implemented as a bit-array of m bits associated with h different hash functions, each of which maps an element to one of the m array positions in a uniformly random manner. All bits in an initial Bloom filter are set to 0, standing for an empty set. To insert an element e into a set represented by a Bloom filter BF, h array positions are calculated by hash functions on e and the bits at those positions in BF are set to 1. Correspondingly, when it is required to check the membership of an element v within the Bloom filter BF, supplying v to hash functions outputs h array positions; if any of the bits at the h positions is 0, then element v does not belong to the set; otherwise, the element is claimed to be a member of the set.

Properties of Bloom Filter

It is easy to see that there is no *false negative* in the Bloom filter membership verification—an element which tests negative within a Bloom filter definitely is not a legitimate member of the set. On the other hand, Bloom filter may yield *false positive*: a member outside the set passing the membership verification on the Bloom filter. The probability of a false positive for an element not in the set, or the *false positive probability*, can be calculated in a straightforward manner.

Let ζ be the probability of a bit being 0 in a Bloom filter, the false positive probability is then

$$P_{\text{FP}} = (1-\zeta)^h \quad . \tag{6.1}$$

After inserting t elements in a Bloom filter, assume that the hash functions outputs are independently uniformly distributed, we have

$$\zeta = \left(1 - \frac{1}{m}\right)^{ht} \approx e^{-ht/m} \quad ;$$

the false positive probability is therefore

$$P_{\text{FP}} = \left(1 - \left(1 - \frac{1}{m}\right)^{ht}\right)^h \approx \left(1 - e^{-ht/m}\right)^h \quad . \tag{6.2}$$

The right hand side is minimized when

$$h = \frac{m}{t} \ln 2 \approx 0.6931 \frac{m}{t} \quad .$$

In such a case, $\zeta = 0.5$, and the false positive probability is $2^{-h} \approx (0.6185)^{m/t}$. Hence, the optimal results are achieved when each bit of the Bloom filter is 0 with probability 1/2.

In a typical parameters determination procedure of Bloom filter, users specify desired false positive probability P_{FP} and predicted inserted element number t, then h and m can be calculated via

$$h = \lceil -\log_2 P_{\text{FP}} \rceil$$

and

$$m = \lceil ht / \ln 2 \rceil \quad .$$

6.2.2 Specification of Secure Bloom Filter

Suppose that the Bloom filter is used in a distributed environment and each element is associated with a specific user claiming an event. It is easy to see that the original Bloom filter data structure does not support a necessary security property—A malicious user or an adversary which is allowed to manipulate the data can forge other users' inputs. We introduce the *secure Bloom filter* to prevent this attack. Formally, it is defined as follows.

Definition 6.1 (Secure Bloom Filter)**.** *Suppose that every user is allocated with a random, secret key. A secure Bloom filter is a data structure that not only maintains all properties of the original Bloom filter, but also guarantees the infeasibility of an adversary constructing*

a valid input associated with a user claiming an event, without the knowledge of the user's key. The probability of a successful forging is defined as the advantage of an adversary against the secure Bloom filter, and is denoted by Adv.

Secure Bloom filter can been constructed by deploying a cryptographic hash function, through HMAC, to substitute a family of hash functions for the Bloom filter.

Definition of a Standard HMAC

We use the cryptographic hash function by means of HMAC because it is provably secure [20] and has been standardized [97]. Let $\mathsf{H}(*)$ denote a cryptographic hash function. According to RFC 2104 [97], to authenticate a message e with a secret symmetric key K, $\mathsf{HMac}(K, e)$ is defined by

$$\mathsf{HMac}(K, e) = \mathsf{H}\left((K \oplus \text{opad}) \parallel \mathsf{H}((K \oplus \text{ipad}) \parallel e)\right),$$

where $\parallel$ denotes concatenation, $\oplus$ denotes exclusive-OR (XOR), opad is the one-block-size outer padding (0x5c5c...5c), and ipad is the one-block-size inner padding (0x3636...36).

Construction of Secure Bloom Filter

Now we make use of HMAC to construct secure Bloom filter that efficiently supports event reporting and message aggregation in wireless sensor networks. Suppose there are c predefined events $(e_1, e_2, \ldots, e_c)$, and each event is associated with an event identifier, which is a nonce and may be refreshed by the base station on a regular basis. When node u with identity id_u detects event e_i, whose current event identifier is N_{e_i}, the node creates a secure Bloom filter BF_{u,e_i} as follows.

For simplicity and computational efficiency, we assume that the size of a secure Bloom filter is a power of two, say $m = 2^d$, where d is an integer. We stress that in principal m can be any positive integer. Recall that node u shares a distinct symmetric key K_u with the base station. The node calculates $\mathsf{HMac}(K_u, id_u \| N_{e_i})$ and the output is divided into h equal size pieces of d bits each [2] such that each piece indicates a position on the Bloom filter array. We denote this processing by $\mathsf{BF}_{u,e_i} \leftarrow \mathsf{HMac}(K_u, id_u \| N_{e_i})$. Based on the pseudorandomness assumption of the cryptographic hash function $\mathsf{H}(*)$, it effectively performs as k distinct hash functions with uniformly-distributed output, and all theoretical analysis on the classical Bloom filter still holds for the secure Bloom filter.

[2] If the output block size of HMAC is greater than dh, it can be truncated. When the size is less than dh, we may conduct several rounds of HMAC operations with different salts to obtain sufficient output.

Selection of Cryptographic Hash Function

For the specific choice of the cryptographic hash function, at present we select SHA-1. Even though collisions have been found in SHA-1 [147], which raises serious concerns on its security, virtually it does not damage the secure Bloom filter, since its security, same as HMAC [20], relies on second preimage resistance of underlying hash function, rather than the more strong requirement of collision resistance. By now, there is no published result of second preimage attacks against SHA-1. Of course, when SHA-3 is finalized in the future, SHA-1 may be replaced by SHA-3 in the protocol for the purpose of higher security level.

6.2.3 Security Property of Secure Bloom Filter

We now show how to select appropriate parameters to achieve a desired security level of a secure Bloom filter. The following theorem states the analytic formula of the adversary's advantage against a secure Bloom filter.

Theorem 6.1. *The advantage of an adversary against secure Bloom filter with parameters* (m, h) *is*

$$\mathsf{Adv}(m,h) = m^{-h} \max_{1 \le i \le h} \tau(h,i) \ , \tag{6.3}$$

where

$$\tau(h,i) = \begin{cases} i^h - \sum_{j=1}^{i-1} \binom{i}{j} \tau(h,j) & \text{if } i > 1 \\ 1 & \text{if } i = 1 \end{cases} \ . \tag{6.4}$$

Proof. Based on the security of HMAC and the pseudorandomness of cryptographic hash function, we derive the advantage of an adversary against secure Bloom filter with parameters (m, h). Without loss of generality, we suppose that adversary $\mathcal{A}$ is required to output a secure Bloom filter $\mathsf{BF}^{\mathcal{A}}$ which should be identical to $\mathsf{BF}_{u,e}$ for node u claiming event N_e.

Let $p_1(i)$ be the probability of the number of bit ones in $\mathsf{BF}_{u,e}$ being i. Obviously $p_1(i)$ is none-zero only if $1 \le i \le h$. Based on the pseudorandomness of cryptographic hash function, we have

$$p_1(i) = \frac{\binom{m}{i} \tau(h,i)}{m^h} \ ,$$

where $\tau(h, i)$ is equal to the output of the following problem: How many distinct words we can get if we use i different letters to construct words of length h while it is required that all i letters are used.

If the underlining HMAC algorithm is secure, then the adversary only can output random $\mathsf{BF}^{\mathcal{A}}$ with the number of bit ones that he selects. When the bit one number of

Table 6.1: Lookup table for secure Bloom filter's parameters selections

h	i for maximal $\tau(h,i)$	maximal $\tau(h,i)$	minimal m	
			for $\mathsf{Adv} \leq 2^{-80}$	for $\mathsf{Adv} \leq 2^{-100}$
6	5	1,800	35,999	362,838
7	5	16,800	11,066	80,176
8	6	191,520	4,684	26,495
9	7	2,328,480	2,417	11,278
10	8	30,240,000	1,434	5,733
11	8	4.79E+08	952	3,356
12	9	8.08E+09	680	2,159
13	10	1.43E+11	514	1,492
14	10	2.73E+12	406	1,093
15	11	5.91E+13	334	842
16	12	1.32E+15	282	671

$\mathsf{BF}_{u,e}$ is i, the probability that a random $\mathsf{BF}^{\mathcal{A}}$ with i bits of one equals $\mathsf{BF}_{u,e}$ is

$$p_2(i) = \frac{1}{\binom{m}{i}} \ .$$

Subsequently, the overall advantage that adversary can achieve is

$$\mathsf{Adv}(m,h) = \max_{1 \leq i \leq h} p_1(i)p_2(i) = m^{-h} \max_{1 \leq i \leq h} \tau(h,i) \ .$$

Consider $\tau(h,i)$ in the context of the word-counting problem. When $i = 1$, $\tau(h,i) = 1$. If $i > 1$, by recursion, we can get

$$\tau(h,i) = i^h - \sum_{j=1}^{i-1} \binom{i}{j} \tau(h,j) \ .$$

This completes the proof. □

In practice, h is relatively small, thus we just explore all values of $\tau(h,i)$ for $1 \leq i \leq h$ by Equation (6.4), and then retrieve $\max_{1 \leq i \leq h} \tau(h,i)$. Subsequently, $\mathsf{Adv}(m,h)$ can be computed by Equation (6.3), or we can determine the minimal m satisfying a designated security level. Intuitively, for 80-bit security, the selections of (m,h) should satisfy $\mathsf{Adv}(m,h) \leq 2^{-80}$. To facilitate parameter selections, we calculate and form Table 6.1 as secure Bloom filter parameters lookup table for $h = 6, 7, \ldots, 16$.

It is worth noting that, by applying Stirling's approximation on

$$p_2(h) = \frac{1}{\binom{m}{h}} = \frac{(m-h)!\ h!}{m!} \ ,$$

we can get

$$p_2(h) \approx h! \left(\frac{e}{m}\right)^h \left(1 - \frac{h}{m}\right)^{m-h+0.5} \ ,$$

which may be further approximated as

$$p_2(h) \approx h!\ e^{(h^2-0.5h)/m} m^{-h} \approx h!\ m^{-h} \ ,$$

when $m \gg h$. In other words, even if an adversary is aware that a particular secure Bloom filter has h bits of ones, the probability that he successfully forges it is roughly $h!\ m^{-h}$.

6.3 Proposed Protocol

6.3.1 Protocol Description and Analysis

An interesting property of Bloom filter is inherently supporting aggregation by bitwise-ORing Bloom filters of a same kind together to generate one Bloom filter such that the set represented by the output Bloom filter is the union of the sets of input Bloom filters. Let $\vee$ denote the bitwise-OR operation on Bloom filters. This desired aggregation property perfectly applies to secure Bloom filters generated by sensor nodes with different keys, and thus we take advantage of it in the proposed protocol, which consists of the following four stages.

Stage 1: Initialization

To activate sensor nodes detection on selected events $(e_1, e_2, \ldots, e_c)$, the base station broadcasts to all nodes an authenticated task message, including fresh nonces $(N_{e_1}, N_{e_2}, \ldots, Ne_c)$ as event identifies, along with events specifications if necessary. For dynamic parameter settings, the task message contains the selected values for the parameters.

Stage 2: Event Claiming and Aggregation

When node u detects that event N_{e_i} occurs, it generates a new Bloom filter

$$\mathsf{BF}_{u,e_i} \leftarrow \mathsf{HMac}(K_u, id_u \| N_{e_i}) \ ,$$

Algorithm 6.1 *ProcessReport*(BF_j): A verifier processes a report BF_j

```
1: if the bit one percentage of BF_j exceeds ρ then
2:    ALERT: BF_j is corrupted, discard it
3:    return false
4: create an empty BF_test for testing
5: for each BF_{u,e_i} do
6:    if BF_{u,e_i} ∧ BF_j = BF_{u,e_i} then
7:       assert that node u has claimed event e_i
8:       BF_test ⇐ BF_test ∨ BF_{u,e_i}
9: if BF_test ≠ BF_j then
10:   ALERT: BF_j is corrupted, discard it
11:   cancel off all previous assertions
12:   return false
13: return true
```

which constitutes the raw message by node u as a contributor to claim event N_{e_i}.

The aggregation on messages is straightforward and efficient. Suppose an aggregator node v receives j messages of BF_i^v, where $i = 1, 2, \cdots, j$, some of which might be contributed by node v itself, it calculates an aggregated message by

$$\mathsf{BF}' = \mathsf{BF}_1^v \vee \mathsf{BF}_2^v \vee \cdots \vee \mathsf{BF}_j^v \ ,$$

To maintain a reasonably slow false positive probability of Bloom filter, aggregators cannot aggregate messages without a limitation. As we see now, the way that we use Bloom filter is quite different from most of other applications. As the message carrier to traverse networks, Bloom filter should not use large m for the sake of communication cost. In fact, $m \leq nc$ for most cases. As result, in addition to m and h for Bloom filter settings, our protocol requires a third parameter: bit one percentage upper-bound ρ. Accordingly, aggregator v tries its best to output one or several aggregated messages with respect to the threshold ρ. In that way, there would be no legitimate Bloom filter whose bit one percentage exceeds ρ. Recall that the false positive probability is optimized when the probability of a bit being 0 in a Bloom filter is 1/2 (see Section 6.2.1), thus threshold ρ is set to be 50% for most circumstances.

After aggregating messages with respect to ρ, aggregator v sends the aggregated message(s) to next node according to the underlying data aggregation organization.

Stage 3: Processing Reports

Eventually, the base station receives l reports BF_j, where $j = 1, 2, \cdots, l$. Using its dis-

tinct keys shared with nodes, the base station calculates all nc Bloom filters corresponding to node-event pairs beforehand. For each report, the base station conducts Algorithm 6.1 to proceed it, intended to discover which nodes claimed which events and to detect attacks.

Let $\wedge$ denote bitwise-AND operation, during the processing procedure, if

$$\mathsf{BF}_{u,e_i} \wedge \mathsf{BF}_j = \mathsf{BF}_{u,e_i} \ ,$$

that is, all bit one occurrences in BF_{u,e_i} appear in BF_j, the base station determines that node u has claimed event e_i. Certainly, false positives may occur for this verification as an inherent downside of Bloom filter. We define the *false positive rate* R_{FP} as the ratio of the count of false positives over the node-event pair number (i.e., nc).

In the ideal case where all l reports are independent and have ρ proportion of bit 1, the false positive probability within a report is ρ^h by Equation (6.1). Suppose nc is much greater than the raw message number, then the number of false positives roughly follows a binomial distribution of parameter $(nc, 1-(1-\rho^h)^l)$, and then its expected value approximates $nc(1-(1-\rho^h)^l)$. Therefore,

$$R_{\text{FP}}^{(\text{ideal})} \approx 1-(1-\rho^h)^l \approx 1-e^{-l\rho^h} \ . \tag{6.5}$$

In reality, most of reports have less than ρ fraction of bit one, and the expected value of false positive number is related to l as bigger l may indicate more number of raw messages. Therefore, $R_{\text{FP}}^{(\text{ideal})}$ is an upper bound of R_{FP}.

If compromised nodes misleadingly claim events with their own keys, clearly there are no effective methods to prevent this attack except for designing and conducting complicated and expensive detection protocols. We are not concerned about that issue; instead the proposed protocol is supported to detect message manipulation attack. The steps related to $\mathsf{BF}_{\text{test}}$ in Algorithm 6.1 carry out that mission.

Recall that by selecting proper (m, h), it is infeasible for the adversary to construct a legitimate secure Bloom filter associated with integrity nodes. During transmission, messages may be maliciously altered by adding or deleting some bits. To detect this kind of attack, actually we only need to consider the case that some bits in reports are *randomly* flipped by the adversary. We stress that the base station is solely responsible and capable of processing messages and those effective manipulations shall eventually reflect on reports; if the adversary replaces a message by another message, in fact it is an interception rather than manipulation, and the countermeasure will be addressed later. If the adversary adds a valid message to an unrelated report, that is not a manipulation attack either, and merely it slightly increase the false positive rate, which is still bounded by Equation (6.5).

We evaluate the probability P_{det} of the base station detecting that a random bit zero in a report is flipped to bit one, which might be the manipulation case that is detected

with least probability. Using the previous ideal case of studying R_{FP}, we can get

$$P_{\mathrm{det}}^{(\mathrm{ideal})} \approx \left(1-\rho^h\left(1-\left(1-\frac{1}{\rho m}\right)^h\right)\right)^{nc} \approx \left(1-\rho^h+\rho^h e^{-h/\rho m}\right)^{nc} \quad . \tag{6.6}$$

Stage 4: Verification by All Nodes

To detect whether malicious nodes intercepted messages related to intact nodes, the base station broadcasts those l reports throughout the whole network. In addition, in order to prevent the message replay attack, the base station should notify all sensor nodes to update event identifiers with new nonces. Surely this stage can be twisted with the initialization stage to add/update/delete events. All those messages are still authenticated by the base station's broadcast authentication scheme.

As long as an intact node is not isolated by compromised nodes from the base station, the node can receive those reports and verity the authenticity of messages. Therefore, the node that claimed the event can determine whether its report is included in the final results. If not, the node notifies this abnormality to the base station in any available secure, uni-cast means by its node key. In addition, if nodes find that there are some false positives related to them, they may report to the base station. If the base station determines that the number of missing raw messages exceeds a reasonable threshold, it may conclude that there exists a message interception attack in the network and further security mechanisms can be conducted.

6.3.2 Discussions

Compression on Bloom Filter

The entropies of raw messages are pretty low—most of bits are zero because there is only one element in a set that the a raw message Bloom filter represents. As a natural way to reduce the transmission sizes of messages, this kind of Bloom filters should be compressed in our protocol. The problem of compressed Bloom filter has been researched by Mitzenmacher [116], who demonstrates that a better compression rate can be achieved by utilizing smaller h and bigger m while maintaining the same level of false positive probability.

However, that rationale does not fit the case of our protocol. First, the selection of h matters to the security level of secure Bloom filter. Second, there are many Bloom filters with varying entropies during the protocol procedure, one pre-determined parameter set

hardly can provide overall desired transmission savings. Therefore, we does not make use of the results in [116]. Instead, a succinct compression mechanism is employed in our protocol: one additional bit b attached to a Bloom filter indicates that the filter is normally stored as before when $b = 0$, and if $b = 1$, then the filter is expressed as a list of all bit one positions. Since the number of bit ones is dynamical, such a list should contain a number-indication segment of bit length $w = \lceil \log_2(m/d) \rceil$, where $m = 2^d$. Before transmitting a Bloom filter, a node chooses the coding which leads to smaller transmission size. Specifically, the list expression is used only if the number of bit ones is less than $(m - w)/d$.

Parameters, Computation and Storage Costs

For sensor networks of size n varying from hundreds to thousands, and to provide 80-bit security level, we recommend $m = 1024, h = 11$ according to Table 6.1. Because we use SHA-1, whose output size is 160 bit, performing one HMAC function that consists of two SHA-1 calculations would suffice for nodes to construct one raw message. This is the major computation burden on nodes; other calculations such as bit one percentage checking and aggregation are simple bitwise operations and are relatively trivial. We assume that the key length is 160-bit, so the primary storage cost of our protocol is merely 20 bytes. Therefore, our protocol is computation- and storage-efficient.

6.4 Simulations

To verify the theoretical analysis above and evaluate the protocol performance, we implement the proposed data aggregation protocol and run simulations on our extended framework over the OMNeT++ simulator (see Section 1.4). By design, we implement secure Bloom filter as a generic class, which is platform-independent and can be used for general purposes.

6.4.1 Simulation Design

Network Scenarios

We run simulations in two network scenarios. The first one is a hierarchical network represented by a Tree rooted on the base station, which fairly emulates many wireless sensor networks. We generate a tree topology in a way that the number of a non-leaf node's children is uniformly selected from four, five, and six. In our experiments, the network size for the Tree Topology increases from 200 to 2000 at a step of 200.

The other one is the Unit-Disk Graph, under which we run simulations for network sizes ranging from 500 to 5000 with a space of 500 between. Likewise, we adjust the node communication range such that the average node degree keeps the approximate 20.

Mechanisms of Claiming Events

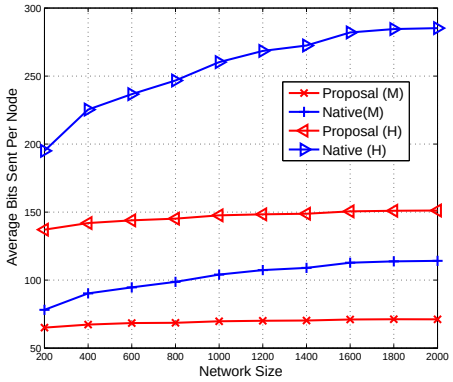
On each network scenario, we specify two different levels of workload to activate nodes claiming events, one for mild work burden, and the other for relatively heavy turnover. For the Tree Topology, in each round of protocol procedure, 20% and 50% of sensor nodes, respectively for two levels of workload, are randomly chosen as contributors to claim an event. For the Unit-Disk Graph, we designate that nodes are capable of detecting events taking place no farther than a distance of 100, and at the beginning of each round, an event occurs at two and six positions (respectively for two workload levels and analogue to 20% and 50% contributors settings in the Tree topology) that are uniformly selected in the square.

Simulation Arrangement

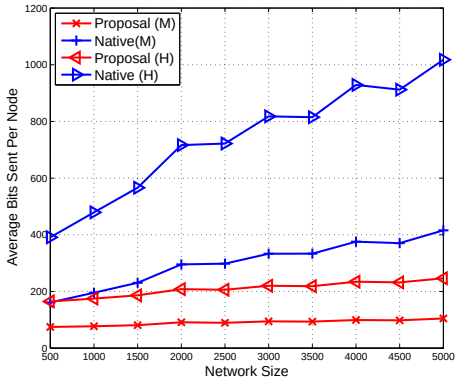
Overall, there are four test cases: Tree Topology with mild workload, Tree Topology with heavy workload, Unit-Disk Graph with mild workload, and Unit-Disk Graph with heavy workload. Each case is associated with ten different network sizes, requiring 40 experiments. Similarly, in order to obtain relatively fair and comparable results, ten random networks in accordance with the parameter settings are constructed for each experiment, each of whose simulation executions is quoted as a run. In each run, we perform 100 rounds of protocol processing with new events and randomly selected contributors/event-locations. For the purpose of communication cost comparison, in addition to our proposed protocol, the native scheme described in Section 6.1.4 is also conducted under the identical conditions. We assume that the message length is 128-bit in the native scheme. Since the last stage is needed for both our protocol and the native scheme, the experiment results on communication cost do not consider that stage. To evaluate the protocol's detection probability P_{det}, we imitate the manipulation attack by randomly flipping a bit zero of a report and then measure whether the base station subsequently detects that the report is illegitimate. For each report, this procedure is repeated by twenty times. The final runs outputs are averaged to create the corresponding experiment results.

6.4.2 Experimental Results

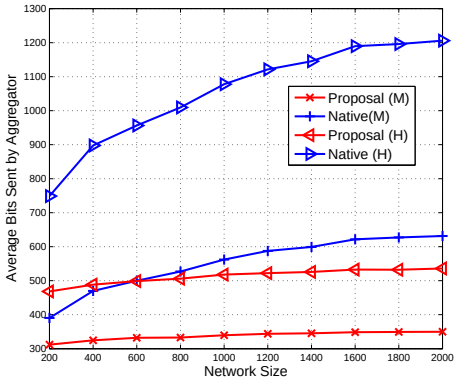
In figures of this chapter, "Proposal", "Native", "(M)", and "(H)" respectively stand for our proposed data aggregation protocol, the native scheme, mild workload, and heavy



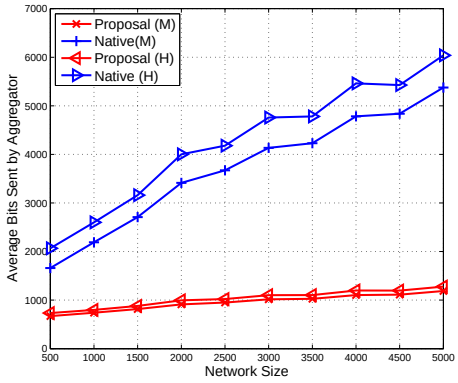
(a) Average Communication Cost (Tree)



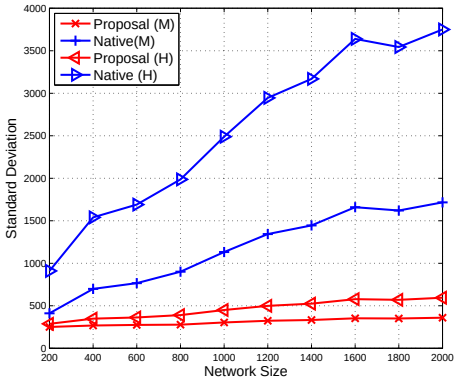
(b) Average Communication Cost (Unit-Disk)



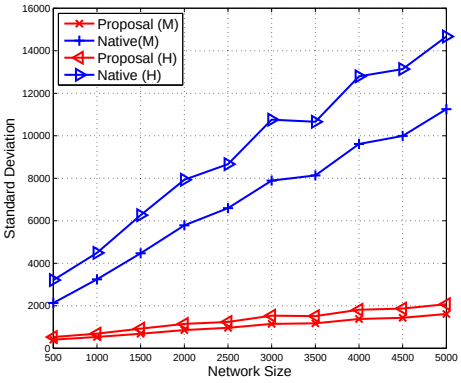
(c) Aggregators Communication Cost, Mean (Tree)



(d) Aggregators Communication Cost, Mean (Unit-Disk)

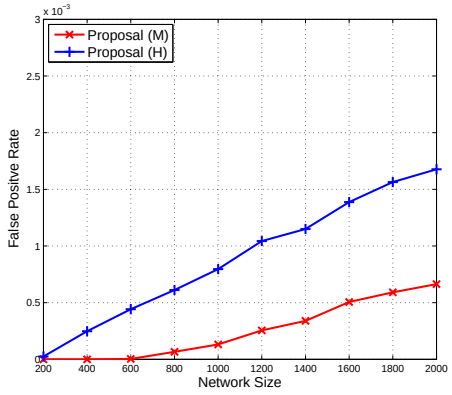


(e) Standard Deviation of (c)

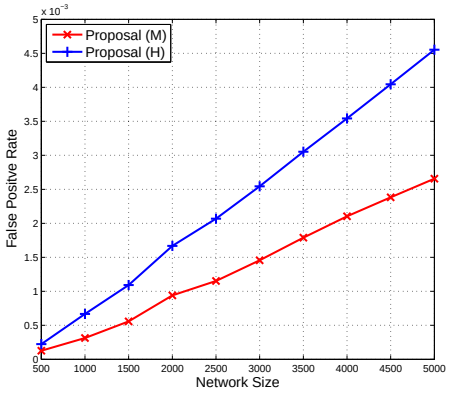


(f) Standard Deviation of (d)

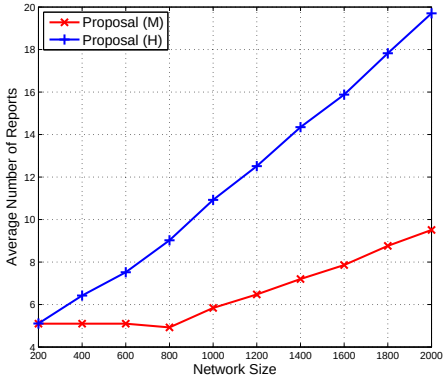
Figure 6.1: Simulation Results on Protocols Communication Performance



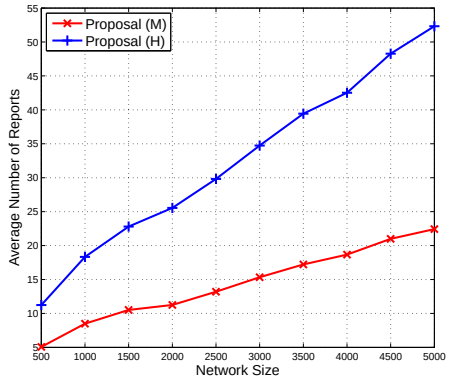
(a) False Positive Rate (Tree)



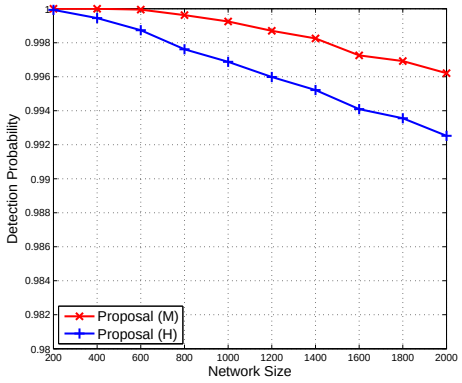
(b) False Positive Rate (Unit-Disk)



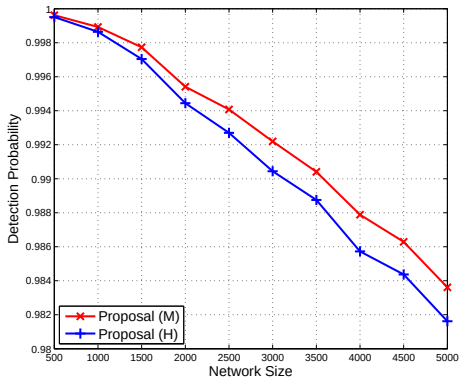
(c) Report Number (Tree)



(d) Report Number (Unit-Disk)



(e) Detection Probability (Tree)



(f) Detection Probability (Unit-Disk)

Figure 6.2: Simulation Results Related to Protocol Security

workload. The protocols communication performance along with their comparison are depicted in Figure 6.1. Similar to many sensor network protocols, the communication cost of the proposed protocol is evaluated by average transmission bits per node, and the experimental results on communication cost are depicted in Figure 6.1a and Figure 6.1b. For a data aggregation scheme, the energy consumption of aggregators and the balance among them are critical performance metrics; so we also measure the means (Figure 6.1c and Figure 6.1d) and the standard deviations (Figure 6.1e and Figure 6.1f) of bits sent by aggregators (intermediate nodes for the native scheme). The simulation results clearly demonstrate that for all four test cases, our protocol remarkably outweighs the native scheme on all three metrics; not only the protocol reduces the overall energy consumption and reserves previous energy of aggregators, but also it achieves splendid balance among aggregators, which is highly desired for data aggregation scheme. Moreover, as those measurements for our protocol gently grow with n, it proves that the proposed protocol scales to network size well, which is quite appreciated for large-size sensor networks.

The simulation results regarding the protocol's security are illuminated in Figure 6.2. Specifically, Figure 6.2a and Fig 6.2b depict the protocol false positive rates; Figure 6.2c and Fig 6.2d show the average report numbers; Figure 6.2e and Figure 6.2f exhibit the detection probabilities. The false positive rates of our protocol are fairly small, compared to typical Bloom filter applications. By applying the experimental results of report number l to Equation (6.5), we verify that the previous theoretical analysis on R_{FP} holds as all false positive rates are indeed bounded on Equation (6.5). Lastly, the high detection probabilities, all of which are greater than 98% in the simulations, exemplify the strong security of our protocol.

6.5 Conclusions

In this chapter, we have proposed a practical secure data aggregation protocol for wireless sensor network based on Bloom filter, and conducted thorough theoretical analysis on the related topics. The extensive simulation results sufficiently demonstrate that the proposal presents remarkable performance on communication cost, energy consumption balance, and security.

Chapter 7

Concluding Remarks

This chapter summarizes the research contributions of the thesis and provides recommendations for future work.

7.1 Summary of Contributions

Efficient Entity Authentication for Low-Cost Devices

Our contributions on entity authentication for low-cost devices are three-fold.

1. *Circulant-P2 Matrix.* As the foundation for our entity authentication proposals, we have introduced a special type of circulant matrix with the name of circulant-P2 matrix, derived from the matrix dimension m that is a prime number satisfying integer two is a primitive element of $GF(m)$. We have proved the linear independence of matrix vectors, present efficient matrix operation algorithms, and explored other important properties.

2. *One-Way Authentication Protocols.* Based on the learning parity with noise problem and circulant-P2 matrix, we have developed two unilateral authentication protocols.

    - The first one is the innovative LCMQ protocol, which is provably secure against all PPT attacks and provides excellent performance on almost all metrics except for the requirement that the verifier should have the computational capacity to perform extended Euclidean algorithm.

- The other one is the $HB^{\mathcal{C}}$ protocol, which utilizes the conventional HB-like authentication structure to preserve the bit-operation only computation requirement for both authentication participants. Compared to previous HB-like protocols, the $HB^{\mathcal{C}}$ protocol substantially reduces the key storage consumption, while other performance measurements as well as the vulnerability to the OOV attack remain. Two enhancement mechanisms are provided to protect the HB-like protocols from the OOV attack and to improve performance.

3. *Mutual Authentication.* We have proposed a framework to incorporate enhanced HB-like protocols, including $HB^{\mathcal{C}}$, into mutual authentication, which is communication-efficient and especially suitable for extremely computation-constrained sensor nodes. The application scenarios in sensor networks have been discussed at length.

Sensor Node Clone Distributed Detection Protocols

We have proposed two distributed protocols to detect node clone with different tradeoffs on network environment, communication cost, and security level. One is based on the popular distributed technique: distributed hash table, while the other utilizes our proposed probabilistic directed exploration mechanism along with initial random direction to facilitate efficient detection.

- The DHT-based protocol can deterministically detect node clone in general sensor networks, and it is accompanied with robust, theoretically analytic formulas to evaluate protocol performance on storage consumption and security level.

- In contrast, the randomly directed exploration protocol presents extraordinary communication performance and minimal storage consumption with high detection probability in dense sensor networks.

For both protocols, the extensive simulations provide thorough performance measurements, and verify the system design and analysis.

Secure Data Aggregation Schemes in Sensor Networks

To fulfill different challenges for securing data aggregation in sensor networks, we have proposed two kinds of schemes. The first is to provide cryptographic integrity mechanisms for general data aggregation, whereas the other is a substantial data aggregation scheme that is suitable for a specific and popular class of aggregation applications, armed with built-in security techniques that effectively defeat outside and inside attacks.

- Data integrity mechanisms that are compatible with general message aggregation are important in sensor networks. Based on recent developments of homomorphic primitives, we have presented three integrity schemes purposely for WSNs data aggregation: a concrete homomorphic MAC construction, homomorphic hash plus aggregate MAC, and homomorphic hash with identity-based aggregate signature. They provide different tradeoffs on security assumption, communication payload, and computation cost.

- By applying HMAC to Bloom filter, we have developed a new data structure: secure Bloom filter, which has inherent aggregation compatibility as well as reliable security properties, and builds the foundation of our proposed secure data aggregation scheme for designated applications. The theoretical analysis and comprehensive simulations indicate that the protocol has competitive performance on security, communication cost, and energy consumption balance among homogeneous nodes.

7.2 Future Work

Providing reliable security mechanisms for large-scale sensor networks of low-cost nodes is an interesting and challenging research area. While many issues have been addressed successfully, other problems still remain open and need further study. On the other hand, many techniques behind our work in this thesis may be extended to other new rising application areas with added efforts.

Extended Applications of Circulant-P2 Matrix

With the attractive properties of circulant-P2 matrix, we believe that it has broader applications on cryptographic algorithms, especially combined with the presence of noise. In the recent years, the learning parity with noise problem has been generalized into the learning with error problem [134], and inspired many lattice-based public-key cryptosystems [115, 122] with promising features, such as provable security reduced to the worse case of a hard problem [134] and fully homomorphic encryption [71]. Unfortunately, those algorithms performance is not satisfactory for practical applications. By means of circulant-P2 matrix, we may achieve substantial improvements in this field. On the other hand, there exist possibilities that some efficient algorithms purely relying on circulant-P2 matrix can be constructed. We will study both of the directions.

Computationally Efficient Cryptographic Primitives

Computationally efficient entity authentication is the starting point of our research on a family of cryptographic primitives for low-cost devices. In the future, we intend to develop message authentication code, symmetric encryption, and hash functions that only require bit-operations. The learning parity with noise problem is surely a main tool that we can count on, but we also need to explore more other techniques.

Homomorphic Algorithms

Homomorphic cryptographic algorithms are the most reliable mechanisms to ensure security for general data aggregation in wireless sensor networks. Even though many homomorphic approaches have been proposed, they suffer from certain restriction that limits their applications in sensor network. In particular, there is no practical homomorphic mechanism that enables any user to verify aggregate messages' integrity in a communication-efficient fashion. This formidable task will be in our study domain.

Secure Cloud Computing

Security is one of the most important and urgent concerns in the cloud computing. Even though the application environment of secure cloud computing is quite different from wireless sensor networks, some techniques may still be useful in this new area. Specifically, cryptographic primitives based on circulant-P2 matrix, homomorphic algorithms, and secure Bloom filter will facilitate our research on secure cloud storage [14, 58] and anonymous search in cloud databases [133].

References

[1] Castalia. http://castalia.npc.nicta.com.au/. 12

[2] OMNeT++ Network Simulator Framework. http://www.omnetpp.org/. 12, 13

[3] The Network Simulator — ns-2. http://www.isi.edu/nsnam/ns/. 12

[4] TinyOS. http://www.tinyos.net/. 12

[5] TOSSIM - TinyOS Documentation Wiki. http://docs.tinyos.net/index.php/TOSSIM. 12

[6] IEEE 802.15.4-2006: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). IEEE Computer Society, 2006. 6

[7] ZigBee 2007 Specification. ZigBee Alliance, 2007. 6

[8] Shweta Agrawal and Dan Boneh. Homomorphic MACs: MAC-Based Integrity for Network Coding. In *Applied Cryptography and Network Security*, pages 292–305. LNCS 5536, 2009. 102, 105, 106, 107, 108

[9] Mehdi-Laurent Akkar, Nicolas Courtois, Romain Duteuil, and Louis Goubin. A Fast and Secure Implementation of Sflash. In *PKC 2003*, pages 267–278. LNCS 2567, 2002. 50

[10] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002. 14, 19, 74

[11] H. M. Ammari and S. K. Das. Integrated Coverage and Connectivity in Wireless Sensor Networks: A Two-Dimensional Percolation Problem. *IEEE Transactions on Computers*, 57(10):1423–1434, 2008. 11

[12] Ross Anderson, Haowen Chan, and Adrian Perrig. Key Infection: Smart Trust for Smart Dust. In *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP'04)*, pages 206–215, 2004. 17, 69

[13] Th Arampatzis, J. Lygeros, and S. Manesis. A Survey of Applications of Wireless Sensors and Wireless Sensor Networks. In *Proceedings of the 13th Mediterranean Conference on Control and Automation*, pages 719–724, 2005. 5

[14] Giuseppe Ateniese, Seny Kamara, and Jonathan Katz. Proofs of Storage from Homomorphic Identification Protocols. In *Advances in Cryptology - ASIACRYPT 2009*, pages 319–333. LNCS 5912, 2009. 136

[15] Chen Avin. *Random geometric graphs: an algorithmic perspective*. PhD thesis, University of California at Los Angeles, 2006. 10

[16] Abdalkarim Awad, Christoph Sommer, Reinhard German, and Falko Dressler. Virtual Cord Protocol (VCP): A flexible DHT-like routing service for sensor networks. In *5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008)*, pages 133–142, 2008. 76, 91

[17] Hari Balakrishnan, M. Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica. Looking up data in P2P systems. *Communications of the ACM*, 46(2):43–48, 2003. 71

[18] B. Balasundaram and S. Butenko. Optimization problems in unit-disk graphs. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization, 2nd Edition*. Springer, 2008. 10

[19] P. Baronti, P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, and Y. F. Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. *Computer Communications*, 30(7):1655–1695, 2007. 19

[20] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying Hash Functions for Message Authentication. In *Advances in Cryptology - CRYPTO'96*, pages 1–15. LNCS 1109, 1996. 111, 121, 122

[21] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362 – 399, 2000. 111

[22] Mihir Bellare and Gregory Neven. Identity-Based Multi-signatures from RSA. In *Topics in Cryptology - CT-RSA 2007*, pages 145–162. 2007. 111

[23] Zinaida Benenson, Nils Gedicke, and Ossi Raivio. Realizing robust user authentication in sensor networks. In *Real-World Wireless Sensor Networks (REALWSN)*, 2005. 14

[24] Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A Practical Stream Cipher with Provable Security. In *Advances in Cryptology - EUROCRYPT 2006*, pages 109–128. LNCS 4004, 2006. 50

[25] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C.A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978. 26

[26] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In *Advances in Cryptology - CRYPTO'98*, pages 1–12. LNCS 1462, 1998. 48

[27] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970. 115, 119

[28] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003. 27

[29] Carlo Blundo, Alfredo De Santis, Amir Herzberg, Shay Kutten, Ugo Vaccaro, and Moti Yung. Perfectly-Secure Key Distribution for Dynamic Conferences. *Information and Computation*, 164(1):1–23, 1998. 10

[30] Alexandra Boldyreva, Craig Gentry, Adam O'Neill, and Dae Hyun Yum. Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 276–285, Alexandria, Virginia, USA, 2007. ACM. 111

[31] Béla Bollobás and Oliver Riordan. *Percolation*. Cambridge University Press, 2006. 11

[32] Dan Boneh, David Freeman, Jonathan Katz, and Brent Waters. Signing a Linear Subspace: Signature Schemes for Network Coding. In *Public Key Cryptography - PKC 2009*, pages 68–87. LNCS 5443, 2009. 105, 110

[33] Heinz Breu and David G. Kirkpatrick. Unit disk graph recognition is NP-hard. *Computational Geometry*, 9(1-2):3–24, 1998. 10

[34] Julien Bringer and Hervé Chabanne. Trusted-HB: A Low-Cost Version of HB+ Secure Against Man-in-the-Middle Attacks. *IEEE Transactions on Information Theory*, 54(9):4339–4342, 2008. 14, 29

[35] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks. In H. Chabanne, editor, *Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2006)*, pages 28–33, 2006. 29

[36] S.R. Broadbent and J.M. Hammersley. Percolation processes I. Crystals and mazes. *Proceedings Of The Cambridge Philosophical Society*, 53(03):629–641, 1957. 10

[37] Richard Brooks, P. Y. Govindaraju, Matthew Pirretti, N. Vijaykrishnan, and Mahmut T. Kandemir. On the Detection of Clones in Sensor Networks Using Random Key Predistribution. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1246–1258, 2007. 17

[38] Matthew Caesar, Miguel Castro, Edmund B. Nightingale, Greg O'Shea, and Antony Rowstron. Virtual ring routing: network routing inspired by DHTs. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '06)*, Pisa, Italy, 2006. ACM. 76, 91

[39] Claude Castelluccia, Aldar C. F. Chan, Einar Mykletun, and Gene Tsudik. Efficient and provably secure aggregation of encrypted data in wireless sensor networks. *ACM Trans. Sen. Netw.*, 5(3):1–36, 2009. 20, 108

[40] Haowen Chan and Adrian Perrig. Efficient security primitives derived from a secure aggregation algorithm. In *Proceedings of the 15th ACM conference on Computer and communications security (CCS'08)*, Alexandria, Virginia, USA, 2008. ACM. 21

[41] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pages 197–213, Berkeley, CA, United States, 2003. 10, 69

[42] Haowen Chan, Adrian Perrig, and Dawn Song. Secure hierarchical in-network aggregation in sensor networks. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 278–287, Alexandria, Virginia, USA, 2006. ACM. 21

[43] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: a distributed storage system for structured data. In *Proceedings of the 7th USENIX*

Symposium on Operating Systems Design and Implementation, volume 7, pages 15–15, Seattle, WA, 2006. USENIX Association. 119

[44] William Y. Chang. Wireless Sensor Networks and Applications. In *Network-Centric Service-Oriented Enterprise*, pages 157–209. 2008. 5

[45] Gilbert Chen, Joel W. Branch, and Boleslaw K. SzymanskiChen. Local leader election, signal strength aware flooding, and routeless routing. In *Proceedings. 19th IEEE International of Parallel and Distributed Processing Symposium (IPDPS'05)*, 2005. 73

[46] Heesook Choi, Sencun Zhu, and Thomas F. La Porta. SET: Detecting node clones in sensor networks. In *Third International Conference on Security and Privacy in Communications Networks and the Workshops (SecureComm 2007)*, pages 341–350, 2007. 17

[47] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990. 10

[48] Mauro Conti, Roberto Di Pietro, Luigi Vincenzo Mancini, and Alessandro Mei. A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'07)*, Montreal, Quebec, Canada, 2007. ACM. 18, 19

[49] Vedat Coskun, Erdal Cayirci, Albert Levi, and Serdar Sancak. Quarantine Region Scheme to Mitigate Spam Attacks in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 5(8):1074–1086, 2006. 16

[50] Nicolas Courtois, Louis Goubin, Willi Meier, and Jean-Daniel Tacier. Solving Underdefined Systems of Multivariate Quadratic Equations. pages 211–227. 2002. 49

[51] James M. Crawford and Michael J. Kearns. The Minimal Disagreement Parity Problem as a Hard Satisfiability Problem. Computational Intelligence Research Laboratory and AT&T Bell Labs, Available at http://www.cs.cornell.edu/selman/docs/crawford-parity.pdf, 1995. 26

[52] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag, 2002. 7, 108

[53] Pengfei Di, M. Yaser Houri, Qing Wei, Jorg Widmer, and Thomas Fuhrmann. Application of DHT-Inspired Routing for Object Tracking. In *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS 2007)*, pages 1–9, 2007. 76

[54] Wenliang Du, Jing Deng, Yunghsiang S. Han, Pramod Varshney, Jonathan Katz, and Aram Khalili. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. *ACM Transactions on Information and System Security (TISSEC)*, 2005. 10, 69

[55] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A key predistribution scheme for sensor networks using deployment knowledge. *IEEE Transactions on Dependable and Secure Computing*, 3(1):62–77, 2006. 12

[56] Dang Nguyen Duc and Kwangjo Kim. Securing HB^+ Against GRS Man-in-the-Middle Attack. In *Proceedings of Symposium on Cryptography and Information Security (SCIS 2007)*, Sasebo, Japan, 2007. 29

[57] Paul Erdös and Alfréd Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960. 8

[58] Chris Erway, Alptekin Küpçü, Charalampos Papamanthou, and Roberto Tamassia. Dynamic provable data possession. In *Proceedings of the 16th ACM conference on Computer and communications security (CCS'09), Full version available at: Cryptology ePrint Archive: Report 2008/432*, pages 213–222, Chicago, Illinois, USA, 2009. ACM. 136

[59] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and Communications Security*, pages 41–47, Washington, DC, USA, 2002. 9, 17, 69, 70

[60] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking*, pages 263–270, Seattle, 1999. IEEE Computer Society. 19

[61] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000. 119

[62] Elena Fasolo, Michele Rossi, Jorg Widmer, and Michele Zorzi. In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 14(2):70–87, 2007. 19

[63] Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, 2007(1):1–10, 2007. 20, 105

[64] Marc Fossorier, Miodrag Mihaljevi, Hideki Imai, Yang Cui, and Kanta Matsuura. An Algorithm for Solving the LPN Problem and Its Application to Security Evaluation of the HB Protocols for RFID Authentication. In *Progress in Cryptology - INDOCRYPT 2006*, pages 48–62. LNCS 4329, 2006. 27

[65] Aviezri S. Fraenkel and Yaacov Yesha. Complexity of solving algebraic equations. *Inf. Process. Lett.*, 10(4/5):178–179, 1980. 50

[66] Keith B. Frikken and Joseph A. Dougherty I. V. An efficient integrity-preserving scheme for hierarchical sensor aggregation. In *Proceedings of the first ACM conference on Wireless network security (WiSec'08)*, pages 68–76, Alexandria, VA, USA, 2008. ACM. 21

[67] Dmitry Frumkin and Adi Shamir. Un-Trusted-HB: Security Vulnerabilities of Trusted-HB. In *The 5th Workshop on RFID Security (RFIDSec 09)*, 2009. 29

[68] Thomas Fuhrmann. Scalable routing for networked sensors and actuators. In *Proceedings of IEEE SECON 2005*, pages 240–251, 2005. 76, 91

[69] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, volume Chapter 7.2 The Polynomial Hierarchy. W. H. Freeman, 1979. 49, 50

[70] Rosario Gennaro, Jonathan Katz, Hugo Krawczyk, and Tal Rabin. Secure Network Coding Over the Integers. In *Public Key Cryptography - PKC 2010*, pages 142–160. LNCS 6056, 2010. 102, 106, 109

[71] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178, Bethesda, MD, USA, 2009. ACM. 20, 105, 135

[72] Craig Gentry and Zulfikar Ramzan. Identity-Based Aggregate Signatures. In *Public Key Cryptography - PKC 2006*, pages 257–273. LNCS 3958, 2006. 102, 111

[73] E. N. Gilbert. Random Plane Networks. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):533–543, 1961. 11

[74] Henri Gilbert, Matt Robshaw, and Herve Sibert. An Active Attack Against HB^+ - A Provably Secure Lightweight Authentication Protocol. Technical report, Cryptology ePrint Archive: Report 2005/237, 2005. 28

[75] Henri Gilbert, Matthew J.B. Robshaw, and Yannick Seurin. Good Variants of HB^+ are Hard to Find. In *Financial Crypt 2008*, 2008. 29

[76] Henri Gilbert, Matthew J.B. Robshaw, and Yannick Seurin. HB$^{\#}$: Increasing the Security and Efficiency of HB^{+}. In *Advances in Cryptology EUROCRYPT 2008, Full version available at: Cryptology ePrint Archive: Report 2008/028*. 2008. 14, 15, 29, 30, 31, 40, 41, 42, 54, 55, 56, 57, 59, 61

[77] Solomon W. Golomb and Guang Gong. *Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar*. Cambridge University Press, 2004. 33

[78] Ghaith Hammouri and Berk Sunar. PUF-HB: A Tamper-Resilient HB Based Authentication Protocol. In *Applied Cryptography and Network Security*, pages 346–365. LNCS 5037, 2008. 29

[79] Johan Håstad. Some optimal inapproximability results. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, El Paso, Texas, United States, 1997. 27

[80] D. R. Heath-Brown. Artin's conjecture for primitive roots. *Quarterly Journal of Mathematics*, 37(1):27–38, 1986. 46

[81] Jun-Won Ho, Matthew Wright, and Sajal K. Das. Fast Detection of Replica Node Attacks in Mobile Sensor Networks Using Sequential Analysis. In *INFOCOM 2009*, pages 1773–1781, 2009. 17

[82] Nicholas Hopper and Manuel Blum. Secure Human Identification Protocols. In *Advances in Cryptology - ASIACRYPT 2001*, pages 52–66. LNCS 2248, 2001. 27

[83] Lingxuan Hu and David Evans. Secure aggregation for wireless networks. In *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, pages 384 – 391, 2003. 20

[84] YihChun Hu, Adrian Perrig, and David B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. In *Proceedings of IEEE Infocom 2003*, San Francisco, 2003. 16

[85] Yueh-Min Huang, Meng-Yen Hsieh, and Frode Eika Sandnes. Wireless Sensor Networks and Applications. In *Sensors, Advancements in Modeling, Design Issues, Fabrication and Practical Applications*, pages 199–219. 2008. 5

[86] Mark L. Huson and Arunabha Sen. Broadcast scheduling algorithms for radio networks. In *IEEE Military Communications Conference (MILCOM '95)*, volume 2, pages 647–651, 1995. 10

[87] Teerawat Issariyakul and Ekram Hossain. *Introduction to Network Simulator NS2*. Springer, 2008. 12

[88] Pawan Jadia and Anish Mathuria. Efficient Secure Aggregation in Sensor Networks. In *High Performance Computing (HiPC 2004)*, pages 40–49. LNCS 3296, 2004. 21

[89] Canming Jiang, Bao Li, and Haixia Xu. An Efficient Scheme for User Authentication in Wireless Sensor Networks. In *21st International Conference on Advanced Information Networking and Applications Workshops*, pages 438–442, 2007. 14

[90] Robert Johnson, David Molnar, Dawn Song, and David Wagner. Homomorphic Signature Schemes. In *Topics in Cryptology - CT-RSA 2002*, pages 204–245. LNCS 2271, 2002. 105

[91] Ari Juels and Stephen A. Weis. Authenticating Pervasive Devices with Human Protocols. In *Advances in Cryptology CRYPTO 2005*, pages 293–308. LNCS 3621, 2005. 14, 15, 27, 28, 40, 41, 59

[92] Jonathan Katz and Andrew Lindell. Aggregate Message Authentication Codes. In *Topics in Cryptology - CT-RSA 2008*, pages 155–169. LNCS 4964, 2008. 102, 110, 118

[93] Jonathan Katz and Ji Shin. Parallel and Concurrent Security of the HB and HB+ Protocols. In *Advances in Cryptology - EUROCRYPT 2006*, pages 73–87. LNCS 4004, 2006. 15, 27, 28, 31, 41, 43, 46, 58, 59, 61, 66

[94] Jonathan Katz and Adam Smith. Analyzing the HB and HB$^+$ Protocols in the "Large Error" Case. Technical report, Cryptology ePrint Archive, Report 2006/326, 2006. 28, 41

[95] Harry Kesten. *Percolation theory for mathematicians*. Birkhauser, 1982. 11

[96] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. In *Advances in Cryptology - EUROCRYPT'99*, pages 206–222. LNCS 1592, 1999. 50

[97] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. RFC 2104 - HMAC: Keyed-Hashing for Message Authentication. IETF, 1997. 121

[98] Maxwell N. Krohn, Michael J. Freedman, and David Mazières. On-the-fly verification of rateless erasure codes for efficient content distribution. In *IEEE Symposium on Security and Privacy 2004*, pages 226–240, 2004. 102, 106, 109, 110

[99] Xuefei Leng, Keith Mayes, and Konstantinos Markantonakis. HB-MP+ Protocol: An Improvement on the HB-MP Protocol. In *IEEE International Conference on RFID*, pages 118–124, 2008. 29

[100] Éric Levieil and Pierre-Alain Fouque. An Improved LPN Algorithm. In *Security and Cryptography for Networks*, pages 348–359. LNCS 4116, 2006. 27, 30, 50, 64

[101] Zhijun Li and Guang Gong. Computationally Efficient Mutual Entity Authentication in Wireless Sensor Networks. In Jun Zheng, Shiwen Mao, Scott F. Midkiff, and Hua Zhu, editors, *Proceedings of First International Conference on Ad Hoc Networks (ADHOCNETS 2009)*, volume 28 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, pages 192–207, Niagara Falls, Ontario, Canada, 2009. Springer. 53

[102] Zhijun Li and Guang Gong. DHT-Based Detection of Node Clone in Wireless Sensor Networks. In Jun Zheng, Shiwen Mao, Scott F. Midkiff, and Hua Zhu, editors, *Proceedings of First International Conference on Ad Hoc Networks (ADHOCNETS 2009)*, volume 28 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST)*, pages 240–255, Niagara Falls, Ontario, Canada, 2009. Springer. 72

[103] Zhijun Li and Guang Gong. Randomly Directed Exploration: An Efficient Node Clone Detection Protocol in Wireless Sensor Networks. In *Proceedings of IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, pages 1030–1035, Macau SAR, P.R.C, 2009. 72

[104] Zhijun Li and Guang Gong. Data Aggregation Integrity Based on Homomorphic Primitives in Sensor Networks. In *Proceedings of 9th International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW 2010)*, pages 149–162. LNCS 6288, 2010. 102

[105] Zhijun Li and Guang Gong. On Data Aggregation with Secure Bloom Filter in Wireless Sensor Networks. Technical report, Technical Report, CACR 2010-22, University of Waterloo, 2010. 115

[106] Zhijun Li and Guang Gong. Computationally efficient mutual entity authentication in wireless sensor networks. *Ad Hoc Networks*, 9(2):204–215, 2011. 53

[107] Zhijun Li, Guang Gong, and Zhiguang Qin. Secure and Efficient LCMQ Entity Authentication Protocol. Technical report, Technical Report, CACR 2010-21, University of Waterloo, submitted to IEEE Transactions on Information Theory, 2010. 25

[108] Rudolf Lidl and Harald Niederreiter. *Finite Fields (Encyclopedia of Mathematics and its Applications)*, volume 20. Addison-Wesley, (Revised version, Cambridge University Press, 1997.), 1983. 33

[109] An Liu and Peng Ning. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In *International Conference on Information Processing in Sensor Networks (IPSN '08)*, pages 245–256, 2008. 72, 110

[110] Donggang Liu and Peng Ning. LocationBased Pairwise Key Establishments for Static Sensor Networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, Conference on Computer and Communications Security, pages 72 – 82, Fairfax, Virginia, 2003. 12

[111] Donggang Liu, Peng Ning, and Rongfang Li. Establishing Pairwise Keys in Distributed Sensor Networks. *ACM Transactions on Information and System Security (TISSEC)*, 8(1):41 – 77, 2005. 10, 69

[112] Yuheng Liu, Chao Li, Yang He, Jing Wu, and Zhang Xiong. A Perimeter Intrusion Detection System Using Dual-Mode Wireless Sensor Networks. In *Second International Conference on Communications and Networking in China (CHINACOM '07)*, pages 861–865, 2007. 117

[113] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland Amsterdam, 1977. 26

[114] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. 7

[115] Daniele Micciancio and Oded Regev. Lattice-based Cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer Berlin Heidelberg, 2009. 135

[116] Michael Mitzenmacher. Compressed bloom filters. *IEEE/ACM Transactions on Networking*, 10(5):604–612, 2002. 127, 128

[117] J. Munilla and A. Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. *Computer Networks*, 51(9):2262–2267, 2007. 29

[118] Leonardo B. Oliveira, Michael Scott, Julio Lopez, and Ricardo Dahab. TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. In *Fifth International Conference on Networked Sensing Systems (INSS 2008)*, pages 173–180, 2008. 72, 112

[119] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the Security of HB# against a Man-in-the-Middle Attack. In *Advances in Cryptology - ASIACRYPT 2008*, pages 108–124. LNCS 5350, 2008. 15, 30, 31, 60, 62

[120] Khaled Ouafi and Serge Vaudenay. Smashing SQUASH-0. In *Advances in Cryptology - EUROCRYPT 2009*, pages 300–312. 2009. 15

[121] Bryan Parno, Adrian Perrig, and Virgil Gligor. Distributed Detection of Node Replication Attacks in Sensor Networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 49–63, 2005. 10, 18, 89, 96, 99

[122] Chris Peikert. Some Recent Progress in Lattice-Based Cryptography. In Omer Reingold, editor, *Theory of Cryptography*, volume 5444 of *Lecture Notes in Computer Science*, pages 72–72. Springer Berlin / Heidelberg, 2009. 135

[123] Pedro Peris-Lopez, Julio Hernandez-Castro, Juan Estevez-Tapiador, and Arturo Ribagorda. M^{2}AP: A Minimalist Mutual-Authentication Protocol for Low-Cost RFID Tags. In *Ubiquitous Intelligence and Computing*, pages 912–923. LNCS 4159, 2006. 15

[124] Adrian Perrig, Robert Szewczyk, Victor Wen David Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 189–199, Rome Italy, 2001. IEEE. 20, 117

[125] Roberto Di Pietro, Luigi V. Mancini, Alessandro Mei, Alessandro Panconesi, and Jaikumar Radhakrishnan. Connectivity properties of secure wireless sensor networks. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, Washington DC, USA, 2004. ACM. 10

[126] Selwyn Piramuthu. HB and Related Lightweight Authentication Protocols for Secure RFID Tag/Reader Authentication. In *CollECTeR Europe Conference*, 2006. 29

[127] Radha Poovendran, Cliff Wang, and Sumit Roy. *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*. Springer Verlag, 2007. 73, 103

[128] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000. 10

[129] Bartosz Przydatek, Dawn Song, and Adrian Perrig. SIA: Secure Information Aggregation in Sensor Networks. In *Proceedings of the first international conference on Embedded networked sensor systems (SenSys'03)*, pages 255 – 265, Los Angeles, California, USA, 2003. 21

[130] Ramesh. Rajagopalan and Pramod. K. Varshney. Data-aggregation techniques in sensor networks: a survey. *IEEE Communications Surveys & Tutorials*, 8(4):48–63, 2006. 19, 116

[131] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, San Diego, California, United States, 2001. ACM. 76

[132] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. GHT: a geographic hash table for data-centric storage. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNS)*, Atlanta, Georgia, USA, 2002. ACM. 19

[133] Mariana Raykova, Binh Vo, Steven M. Bellovin, and Tal Malkin. Secure anonymous database search. In *Proceedings of the 2009 ACM workshop on Cloud computing security (CCSW'09)*, pages 115–126, Chicago, Illinois, USA, 2009. ACM. 136

[134] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, Baltimore, MD, USA, 2005. ACM. 27, 135

[135] Ronald L. Rivest. On the invertibility of the XOR of rotations of a binary word. Unpublished draft of July 18, 2009. Revised November 10, 2009, 2009. 34, 35

[136] Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*. Springer-Verlag, 2001. 76

[137] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53. LNCS 196, 1984. 72

[138] Adi Shamir. SQUASH A New MAC with Provable Security Properties for Highly Constrained Devices Such as RFID Tags. In *Fast Software Encryption*, pages 144–157. 2008. 14, 15

[139] Peter Steffen, Westhoff Dirk, and Castelluccia Claude. A Survey on the Encryption of Convergecast Traffic with In-Network Processing. *IEEE Transactions on Dependable and Secure Computing*, 7:20–34, 2010. 20

[140] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: a scalable peer-to-peer

lookup protocol for internet applications. *IEEE/ACM Transactions on Networking (TON)*, 11(1):17–32, 2003. 76, 82

[141] Piotr Szczechowiak, Leonardo Oliveira, Michael Scott, Martin Collier, and Ricardo Dahab. NanoECC: Testing the Limits of Elliptic Curve Cryptography in Sensor Networks. In *European Conference on Wireless Sensor Networks (EWSN)*, pages 305–320. LNCS 4913, 2008. 72, 110

[142] Patrick Traynor, Raju Kumar, Heesook Choi, Guohong Cao, Sencun Zhu, and Thomas La Porta. Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. *IEEE Transactions on Mobile Computing*, 6(6):663–677, 2007. 10

[143] Somanath Tripathy and Sukumar Nandi. Defense against outside attacks in wireless sensor networks. *Computer Communications*, 31(4):818–826, 2008. 14

[144] Huei-Ru Tseng, Rong-Hong Jan, and Wuu Yang. An Improved Dynamic User Authentication Scheme for Wireless Sensor Networks. In *IEEE Global Telecommunications Conference (GLOBECOM '07)*, pages 986–990, 2007. 14

[145] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully Homomorphic Encryption over the Integers. In *Advances in Cryptology - EUROCRYPT 2010*. LNCS 6110, 2010. 20, 105

[146] András Varga and Rudolf Hornig. An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, pages 1–10, Marseille, France, 2008. ICST. 12

[147] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In *Advances in Cryptology - CRYPTO 2005*, pages 17–36. LNCS 3621, 2005. 81, 122

[148] Dirk Westhoff, Joao Girao, and Mithun Acharya. Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution, and Routing Adaptation. *IEEE Transactions on Mobile Computing*, 5(10):1417–1431, 2006. 20

[149] Kirk H.M. Wong, Yuan Zheng, Jiannong Cao, and Shengwei Wang. A Dynamic User Authentication Scheme for Wireless Sensor Networks. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, pages 244–251, 2006. 14

[150] Anthony D. Wood and John A. Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002. 16

[151] Jiang Wu and Douglas R. Stinson. Minimum node degree and k-connectivity for key predistribution schemes and distributed sensor networks. In *Proceedings of the First ACM Conference on Wireless Network Security (WiSec'08)*, Alexandria, Virginia, USA, 2008. 10

[152] Yang Xiao, Venkata Krishna Rayi, Bo Sun, Xiaojiang Du, Fei Hu, and Michael Galloway. A survey of key management schemes in wireless sensor networks. *Computer Communications*, 30(21-12):2314–2341, 2007. 9

[153] Dong Xuan, Weijia Jia, Wei Zhao, and Hongwen Zhu. A routing protocol for anycast messages. *IEEE Transactions on Parallel and Distributed Systems*, 11(6):571–588, 2000. 92

[154] Guang-Zhong Yang and M. Yacoub. *Body Sensor Networks*. Springer, 2006. 6

[155] Yi Yang, Xinran Wang, Sencun Zhu, and Guohong Cao. A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'06)*, pages 356 – 367, 2006. 21

[156] Yi Yang, Sencun Zhu, and Guohong Cao. Improving sensor network immunity under worm attacks: a software diversity approach. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'08)*, pages 149–158, Hong Kong, China, 2008. ACM. 10

[157] Yanchao Zhang, Wei Liu, Wenjing Lou, and Yuguang Fang. Location-based compromise-tolerant security mechanisms for wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 24(2):247–260, 2006. 16

[158] Bo Zhu, Venkata Gopala Krishna Addada, Sanjeev Setia, Sushil Jajodia, and Sankardas Roy. Efficient Distributed Detection of Node Replication Attacks in Sensor Networks. In *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, pages 257–267, 2007. 18, 19

[159] Sencun Zhu, Sanjeev Setia, and Sushil Jajodia. LEAP : efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the 10th ACM conference on Computer and Communication Security (CCS' 03)*, pages 62–72, Washington D.C., 2003. ACM. 17, 69