Design of access authentication schemes in 5G wireless networks

Chow, Man Chun

2021

Chow, M. C. (2021). Design of access authentication schemes in 5G wireless networks.
Master's thesis, Nanyang Technological University, Singapore.
<https://hdl.handle.net/10356/151481>

<https://hdl.handle.net/10356/151481>

<https://doi.org/10.32657/10356/151481>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0
International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 22:57:16 SGT

**Design of Access Authentication Schemes in 5G Wireless
Networks**

CHOW MAN CHUN

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarized materials, and has not been submitted for a higher degree to any other University or Institution.

03-06-2021

.....

Date

.....

CHOW MAN CHUN

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

3/6/2021

.....

Date

.....

Prof. Maode Ma

Authorship Attribution Statement

This thesis contains material from 3 papers published in the following peer-reviewed journal(s) / from papers accepted at conferences / book chapters in which I am listed as an author.

Chapter 2.2 is published as M. C. Chow and M. Ma, "Secure D2D in 5G Cellular Network: Architecture, Requirements and Solutions," in *Advances in Computing, Informatics, Networking and Cybersecurity*, Springer Nature, 2021.

Chapter 3 is published as M. C. Chow and M. Ma, "A Lightweight D2D Authentication Scheme against Free-riding Attacks in 5G Cellular Network," in *Proceedings of 2020 International Electronics Communication Conference, IECC 2020*, 2020. DOI: 10.1145/3409934.3409952.

Chapter 4 is published as M. C. Chow and M. Ma, "A Blockchain-Enabled 5G Authentication Scheme Against DoS Attacks," in *Proceedings of 2020 International Conference on Electronics, Communications and Information Technology, CECIT 2020*, 2020.

For both Chapter 2.2, 3 and 4, the contributions of the co-authors are as follows:

- I wrote the drafts of the manuscript.
- Manuscripts were revised together with Prof. Ma.
- I am the first author of all publications.
- Prof. Ma is the corresponding author of all publications.

03-06-2021

.....
Date

.....
CHOW MAN CHUN

Acknowledgement

I praise the almighty God as He is the beginning of all knowledge.

I would also like to express my gratitude to my supervisor, Prof. Maode Ma, for his expertise, guidance, and funding supports throughout my master's degree journey. He always provided professional and invaluable comments, and I believe the completion of this work would not be possible without his support.

Finally, I would like to thank my parents for their love, encouragement and supports.

Summary

The futuristic 5G network is designed to accommodate the increasing number of connected devices with higher speed, lower latency, and better security. To achieve these goals, two major strategies have been proposed recently: First, building the cellular network with many tiny base stations, 5G reduces the wireless transmission delays and improves spectral efficacy. Second, by allowing device-to-device (D2D) communication in the cellular network, it offloads the heavy network traffic among mobile devices in a distributed manner. However, due to the wireless nature of 5G communication, there will be many security challenges such as mutual authentication, device anonymity, end-to-end data secrecy, free-riding attack prevention, and so on. Although the Third Generation Partnership Project (3GPP) has recently standardized the 5G-AKA protocol for mutually authenticating user equipment (UE), base stations, and the core network, we found that there are still some security issues yet to resolve in the protocol. Furthermore, 3GPP has not standardized the authentication protocol for 5G-D2D communication, making D2D communication remain vulnerable in the 5G networks. In view of these issues, there is a need to develop authentication protocols that address all these security requirements in the 5G and 5G-D2D networks. The designed protocols should be computationally lightweight, such that all mobile devices including low-end smartphones and 5G-IoT connected devices can also enjoy 5G and 5G-D2D safely.

To fill these research gaps, in this thesis, we first provide the background and a comprehensive literature survey to the security in 3GPP 5G and 5G-D2D network. We list out the security architecture of the 5G Core network and 5G D2D network, define their security requirements based on the current architecture, and then discuss the existing solutions by evaluating both their advantages and shortcomings. Next, we propose two security schemes, namely the Secure Blockchain-based Authentication and Key Agreement Scheme (5GSBA), and the Lightweight Traceable D2D authentication and Key Agreement Scheme (LT-AKA). 5GSBA aims to improve security in 5G, and LT-AKA aims to improve 5G-D2D. After that, for each proposal, we evaluate their security and logical correctness using formal verification tools. Performance evaluations for each scheme also show that the proposed schemes are computationally and spatially efficient while maintaining a high level of security.

List of Figures

Figure 1 - Timeline of Cellular Network Evolution.....	2
Figure 2 - Technology used in 5G Network.....	3
Figure 3 - 3GPP 5G Core System and Security Architecture	3
Figure 4 – 5G D2D Coverage Scenarios	5
Figure 5 - 3GPP ProSe System and Security Architecture	6
Figure 6 - Sequence Diagram of 5G-AKA Authentication.....	13
Figure 7 - System and Security Model for 5GSBA	53
Figure 8 - Sequence Diagram of 5GSBA.....	58
Figure 9 - Structure of A Block.....	59
Figure 10 - Structure of A Transaction	60
Figure 11 - Scyther Tool	69
Figure 12 - SPDL Model of 5GSBA.....	71
Figure 13 - Formal Verification of 5GSBA with Scyther Tool	72
Figure 14 – Experimental Total Computational Delays.....	77
Figure 15 - Average Delays under Unknown Attacks	78
Figure 16 – Total Bandwidth Consumption.....	79
Figure 17 - Total Communication Delays.....	80
Figure 18 - Storage Overhead for UE	81
Figure 19 - Size of Blockchain within A Decade.....	82
Figure 20 - Energy Consumption for UE	83
Figure 21 - Average Delays under Many Requests.....	85
Figure 22 - Successful Authentications under DoS Attacks	85
Figure 23 - System Model of LT-AKA.....	89
Figure 24 - Our Proposed LT-AKA Protocol.....	91
Figure 25 - SPDL Model of LT-AKA.....	99
Figure 26 - Formal Verification of LT-AKA with Scyther Tool	100
Figure 27 - Total Computational and Transmission Overheads	105
Figure 28 - Storage Overhead for UE	106
Figure 29 - Average Delays Incurred due to Unknown Attacks	107

Figure 30 - Energy Consumption of UE	108
Figure 31 - Total Energy Consumption of a UE Pair due to Unknown Attacks.....	109

List of Tables

Table 1 - Security Issues from 1G to 4G.....	7
Table 2 – Notations in 5GSBA	57
Table 3 - Experimental Time for Different Functions	76
Table 4 - Theoretical Computational Overheads	77
Table 5 - Parameters in LT-AKA.....	91
Table 6 - Experimental Computational Time for Algorithms.....	103
Table 7 - Theoretical and Experimental Computational Time.....	103
Table 8 – Total Signaling Message Length in bits.....	104

List of Acronyms

The acronyms and abbreviations used in this report are presented as follows:

3GPP	Third Generation Partnership Project
4G	The Fourth Generation Technology Standard for Cellular Networks
5G	The Fifth Generation Technology Standard for Cellular Networks
5GC	5G Core Network
5GSBA	Secure Blockchain-based AKA Scheme for the 5G Networks
AEAD	Authentication Encryption with Associated Data
AKA	Authentication and Key Agreement
AMF	Access and Mobility Function
AMPS	Advanced Mobile Phone Service
ARPF	Authentication credential Repository and Processing Function
AUSF	Authentication Server Function
AVISPA	Automated Validation of Internet Security Protocols and Applications
BAN	Burrows-Abadi-Needham Logic
CA	Certificate Authority
CUE	Cellular User Equipment
D2D	Device to Device
DHKE	Diffie-Hellman Key Exchange
DUE	D2D-Compatible User Equipment
DoS	Denial of Service
EAP	Extensible Authentication Protocol
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
EPC	Evolved Packet Core
EPS	Evolved Packet System

GSM	Global System for Mobile Communication
GUTI	Global Unique Temporary Identifier
HMAC	Keyed-hash Message Authentication Code
LT-AKA	Lightweight Traceable Authentication and Key Agreement Scheme
LTE	Long Term Evolution
ME	Mobile Equipment
NSSF	Network Slice Selection Function
PKI	Public Key Infrastructure
PLMN	Public Land Mobile Network
PLS	Physical Layer Security
PoW	Proof-of-Work
PoS	Proof-of-Stake
ProSe	Proximity-based Service
SDN	Software-Defined Network
SEAF	Security Anchor Function
SMF	Session Management Function
SUCI	Subscription Concealed Identifier
SUPI	Subscription Permanent Identifier
UDM	Unified Data Management
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UPF	User Plane Function
URLLC	Ultra-reliable Low Latency Network
USIM	Universal Subscriber Identity Module
eMBB	Enhanced Mobile Broadband
eNB	Evolved Node B
gNB	Next Generation Node B
mMTC	Massive Machine Type Communication

MEC	Mobile Edge Computing
MIMO	Multiple-Input Multiple-Output
MITM	Man-in-the-middle Attack
NFV	Network Function Virtualization
NSSF	Network Slice Selection Function

Table of Contents

<i>Statement of Originality</i>	<i>I</i>
<i>Supervisor Declaration Statement</i>	<i>II</i>
<i>Authorship Attribution Statement</i>	<i>III</i>
<i>Acknowledgement</i>	<i>IV</i>
<i>Summary</i>	<i>V</i>
<i>List of Figures</i>	<i>VI</i>
<i>List of Tables</i>	<i>VIII</i>
<i>List of Acronyms</i>	<i>IX</i>
Chapter 1. Introduction	1
1.1. Background	1
1.1.1. 3GPP 5G Networks	1
1.1.2. 3GPP 5G-D2D Communication	4
1.1.3. 5G Security Requirements.....	6
1.2. Motivation	9
1.3. Objectives	10
1.4. Contribution	10
1.5. Organization of the Thesis	11
Chapter 2. Literature Review	12
2.1. 5G Networks	12
2.1.1. 5G-AKA Scheme.....	12
2.1.2. Existing Solutions for 5G Authentication	14
2.2. 5G-D2D Networks	24
2.2.1. Physical Layer Security (PLS)	24
2.2.2. Cryptographic Techniques.....	28
2.3. Conclusion of the Literature Review	50
Chapter 3. A Secure Blockchain-based Authentication and Key Agreement Scheme for 3GPP 5G Networks (5GSBA)	51
3.1. Introduction	51
3.2. System and Security Model	52
3.2.1. System Model	52
3.2.2. Security Model	54
3.3. Details of the 5GSBA Protocol	54
3.3.1. Preliminary	54
3.3.2. Details of the Protocol	57
3.4. Security Evaluation	61
3.4.1. Logical Proof with Burrows-Abadi-Needham (BAN) Logic	61
3.4.2. Formal Verification with Scyther Tool	68

3.4.3. Security Analysis	72
3.5. Performance Evaluation	74
3.5.1. Computational Overheads	75
3.5.2. Communication Overheads	78
3.5.3. Storage Overheads	80
3.5.4. Energy Consumption for UE	82
3.5.5. Resistivity to DoS and DDoS Attacks	83
3.7. Conclusion of the 5GSBA	86
<i>Chapter 4. A Lightweight Traceable D2D Authentication and Key Agreement Scheme in 5G Cellular Networks (LT-AKA).....</i>	<i>87</i>
4.1. Introduction	87
4.2. System and Security Model	88
4.2.1. System Model	88
4.2.2. Security Model	89
4.3. Details of the LT-AKA Protocol	90
4.3.1. Preliminary	90
4.3.2. Details of the Protocol	91
4.4. Security Evaluation	93
4.4.1. Logical Proof with Burrows-Abadi-Needham (BAN) Logic	93
4.4.2. Formal Verification with Scyther Tool	97
4.4.3. Security Analysis	100
4.5. Performance Evaluation	102
4.5.1. Computational Overhead	102
4.5.2. Transmission Overhead	104
4.5.3. Storage Overhead for UE	105
4.5.4. Average Delays	106
4.5.5. Energy Consumption for UE	107
4.6. Conclusion of the LT-AKA.....	109
<i>Chapter 5. Conclusion</i>	<i>110</i>
5.1. Contributions	110
5.2. Future Research Work	110
<i>List of Publication.....</i>	<i>112</i>
<i>Bibliography.....</i>	<i>113</i>

Chapter 1. Introduction

1.1. Background

The ever-increasing number of mobile devices and network traffic has fostered the development of the next-generation cellular network. In the future, it is expected that the cellular network should be fast, responsive, and capacious to many network users. However, due to the wireless nature of the cellular network communication, there could be many new security challenges. In this chapter, we investigate the core specifications in the latest 3GPP 5G networks. Then, we further discuss one of the promising new features in 5G: the 5G D2D communications. Through the extensive discussion of specifications, we identify some important security threats, and derive security requirements for the 5G networks.

1.1.1. 3GPP 5G Networks

1.1.1.1. Overview

The Fifth Generation Technology Standard for Cellular Networks (5G) is the next-generation mobile network expected to launch by the end of 2020 [1]. In 5G, many different application scenarios are supported under the same network and the same infrastructure. For example, enhanced mobile broadband (eMBB) in 5G provides high-speed internet connection to mobile devices and laptop computers. The ultra-reliable low latency network (URLLC) in 5G supports vehicular applications that require fast and real-time responses. Also, the massive Machine Type Communication (mMTC) in 5G allows many power-constrained IoT devices to upload their data with very low energy consumption. All these application scenarios will be connected to the same network, using the same set of base stations and functional entities. To simply put, the 5G network is expected to be capable to provide both high speed, high capacity, high coverage, and low latency connections at the same time with efficient resource allocation [2].

1.1.1.2. Evolution from 1G to 5G

Every upgrade of the cellular network generation represents a major paradigm shift, and each shift emerges around every 10 years [1], [3]. The development of the cellular network is illustrated in Figure 1. To be specific, in the 1980s, the first-generation networks (1G) provided basic analog voice communication services. Then, in the 1990s, 2G networks digitalized the voice network and introduced Short Message Services (SMS) for text messaging. In the 2000s, 3G networks enabled basic internet access, mobile televisions, and video calls. In the 2010s, 4G enabled deprecated the circuit-switched network to use the packet-switched network. It also provided a much higher internet connection speed. In the next decade (the 2020s), 5G networks will launch with the following highlights [1]:

- **One network for everything, everywhere:** ranging from mobile devices, laptop computers, internet-of-vehicles (IoV), internet-of-things (IoT) gadgets to tiny smart meters, all devices with different speed and latency requirements will connect to the same 5G core network. Also, the coverage of 5G networks will become ubiquitous, such that users can enjoy a seamless network experience.
- **Low latency network:** unlike 4G LTE networks that the quality of service (QoS) could not be easily controlled, 5G networks aim to minimize the network latency for critical applications such as autonomous driving or video conferencing.
- **Higher network throughput:** it is foreseen that bandwidth-intensive application, such as virtual reality (VR) and augmented reality (AR) will be common in the future. Hence, 5G will also provide high-speed wireless broadband to mobile users.

Figure 1 - Timeline of Cellular Network Evolution

1.1.1.3. 5G Technology

In response to the stringent requirements, researchers have been proposing many new strategies, such as improving the physical layer with beamforming and multiple-input and multiple-output (MIMO) technology, deploying many tiny base stations with millimeter waves (mmWave), splitting the network control planes with network function virtualization (NFV) and software-defined network (SDN), enabling device-to-device communication (D2D), improving machine-to-machine communication (M2M) and so on [4]. With all these techniques, the limited spectrum resources can reuse effectively to provide better services to more devices simultaneously. Figure 2 shows a summary of technology used in the 5G network.

Figure 2 - Technology used in 5G Network

1.1.1.4. 3GPP 5G Core Network System Architecture

According to the latest 3GPP 5G Security Architecture Release 15.7 in 2019 [5], the overall system architecture involves many functional entities including the Next Generation Node B (gNB), Access and Mobility Function (AMF), Security Anchor Function (SEAF), Authentication Server Function (AUSF), Authentication credential Repository and Processing Function (ARPF), Network Slice Selection Function (NSSF), Session Management Function (SMF), Unified Data Management (UDM), User Plane Function (UPF) and so on. Figure 3 shows the simplified architecture of the latest 3GPP 5G Core network. However, it is noteworthy that the latest 3GPP 5G Core does not include the ProSe Function, or the ProSe Application Server mentioned in ProSe system architecture specification TS 23.303 [6]. Since 3GPP did not finalize how D2D is supported in the 5G network, many researchers take different approaches: some proposed to put ProSe related features into the existing functional entities, while others proposed to add back the ProSe related entities into the core network. As the importance of D2D communication is rising, this discrepancy is expected to be resolved in the near future.

Figure 3 - 3GPP 5G Core System and Security Architecture

1.1.2. 3GPP 5G-D2D Communication

1.1.2.1. Overview

As one of the promising features in 5G, D2D communications is the technology allowing adjacent mobile devices to communicate directly without relaying user data over the base stations. In recent years, increasing amount of papers are carried out to use D2D for providing efficient multi-hop cellular network, content distribution service, P2P gaming and so on [7]. Apart from academia, the Third Generation Partnership Project (3GPP) has put D2D communication under the umbrella of Proximity Service (ProSe) [6]. 3GPP started feasibility study and standardization of D2D communication since 3GPP Rel. 12 as ProSe (Proximity Service). It covered some basic requirements such as D2D communication via optimized path, device discovery and public safety applications [8]. Since then, more new services and requirements such as multi-hop relay network, push-to-talk, emergency services and location services are also included in the latest specification [9].

1.1.2.2. 5G-D2D Advantages

Having ever-increasing amount of network traffic and many new applications in future mobile systems, D2D communication in 5G provides these benefits:

Network Efficiency: When nearby users can communicate directly without relying on base stations, the burden of base stations can be relieved such that it allows more concurrent users to join the network [9]. Also, with proper radio resource allocation scheme and power control scheme, it is believed that the hybrid of D2D network and regular cellular network would further improve the spectral efficiency thus the overall throughput of the cellular network [10], [11]. Moreover, using D2D network as a relay to conventional base station signals, cellular coverage can also be extended such that the network performance at the edge of cellular can be improved [12].

Smart Application: D2D communication is also an enabler to smart-city applications such as wireless sensors network which all sensors are connected and transmitted with relays [13]. It is also envisioned that doctors can use tiny health monitoring sensors to monitor their patients real-time with D2D relay network [11].

Public Safety Application: In life-threatening situations such as terrorist attacks or natural disasters, D2D could also ensure public safety by constructing a relay network for urgent communication, even when some or all base stations are unavailable [13].

Social Application: Similar to other existing D2D protocols, 5G D2D also allows adjacent users to perform distributed content sharing, file sharing, cell group chatting and P2P gaming without cellular network [12].

1.1.2.3. 5G-D2D Network Topology

Depending to the presence of cellular network, D2D communication can be categorized into the following three different modes [14], [15], and it is also illustrated in Figure 4:

In-coverage mode: All D2D UEs are under the coverage of base station signal, and all network resources are managed by cellular network. Therefore, D2D connection tasks such as device identification, discovery, connection establishment and key management can be delegated to network operators.

Relay-coverage / Partial-coverage mode: Some D2D UEs are out of coverage of base station signals. In this scenario, UEs under coverage will become network relays to allow all other UEs to access the core network. Similar to in-coverage mode, D2D connection tasks can also be delegated to network operators because all UEs can still access core network.

Out-of-coverage mode: When all devices are out of base station signals, D2D connection tasks should be handled by UEs independently. This connection mode is particularly useful for public safety applications because base stations may be heavily congested or even damaged at emergency.

Figure 4 – 5G D2D Coverage Scenarios

Besides, unlike the conventional cellular network which all communication are based on licensed bands, D2D communication can be implemented on both licensed bands or unlicensed bands [7]. For instance, proposed and implemented by Qualcomm, FlashLinQ is a D2D protocol which provides peer discovery, link management and power management over a special licensed band (2.586 GHz) [16]. Wi-Fi Direct, another predominant D2D protocol introduced by Wi-Fi Alliance, is using the unlicensed Wi-Fi and NFC spectrum to provide D2D device discovery and D2D communication [17].

1.1.2.4. 5G-D2D System Architecture

3GPP proposed Proximity-based Service (ProSe) architecture models [6] based on 4G LTE Evolved

Packet Core (EPC) framework, and it is expected that the overall architecture would remain similar in 5G Core Network (5GC). The proposal included various connection scenarios. For example, both two UE are in the same carrier (i.e., non-roaming scenario), both two UEs are connected to different carriers (i.e., inter-Public Land Mobile Network (PLMN) scenario) and more than one UE are roaming (i.e., roaming scenario). To illustrate all these connection modes graphically, Figure 5 shows the simplified system architecture of ProSe in core network which focuses only on interaction between UEs, ProSe Functions and ProSe App Server. In the figure, ProSe Function and ProSe App Server are located within the core network. The ProSe Function entity is responsible for UE provision, and the ProSe App Server is responsible for ProSe Direct Discovery and all other D2D related services. ProSe Function can communicate with other ProSe Function entities with path PC6 and PC7. It also communicates with UEs through path PC3 and cooperate with ProSe App Server through PC2. PC1 is the application-level connection which allows ProSe related applications in UEs to access ProSe App Server directly. Finally, the D2D communication path between UEs is labelled as PC5 [2]: this path can be either a one-to-one direct connection or a one-to-many group communication. Also, under the public safety circumstances, this path should also support relay traffic such that all UEs can still access the core network even if some UEs are outside of coverage.

Figure 5 - 3GPP ProSe System and Security Architecture

1.1.3. 5G Security Requirements

1.1.3.1. Lessons from 1G to 4G

To understand what the security requirements in the 5G networks can be, and how are the industry planning to resolve them, we should briefly review how security measures in the previous generations of cellular networks evolved. Table 1 shows some security measures employed for each generation, and some new security threats found while using the network [18]. It shows that every generation has improved the security incrementally, but there are also some threats remained unresolved. Also, there are more new security threats surfaced due to the advancing and more sophisticated hacking techniques.

Table 1 - Security Issues from 1G to 4G

Network	Security Measures	Some Security Threats
1G (AMPS)	No security measures	<ul style="list-style-type: none"> • Wire-tapping: all calls can be intercepted.
2G (GSM)	Unilateral Authentication Symmetric Encryptions	<ul style="list-style-type: none"> • Location tracking with IMSI. • Fake base stations.
3G (UMTS)	Mutual Authentication	<ul style="list-style-type: none"> • Location tracking with IMSI. • Violation of data security during roaming.
4G (LTE)	Mutual Authentication with Improved EPS-AKA Non-3GPP Access with EAP-AKA	<ul style="list-style-type: none"> • Location tracking with IMSI. • Easier Denial of Service (DoS) attacks due to more base stations and connected devices. • Lack of backward secrecy [19] • Desynchronization attacks during handover [19]

1.1.3.2. Security Domains

According to the system architecture and network topology modes aforementioned, we assume that all functional entities within and among different operators should communicate with each other using the existing inter-PLMN connections. Also, there should be ProSe Function entities located within the core network which facilitate D2D related services. Consequently, all security solutions related to 5G and 5G-D2D can also be classified into the following four domains:

- **Security within the 5G core network:** it refers to the connections between UE and the D2D related functional entities inside the core network. It includes the communication between ProSe Function entity and core networks (PC4) and communication between ProSe Function entities and UE devices (PC1, PC3). They should be classified into the same group because these connections should be protected and trusted under the pre-existing 5G core network security architecture.
- **Security between 5G core networks:** it refers to the connections which allow D2D services to work even if two D2D UEs are using different PLMNs (i.e., carriers) or during roaming. An example would be the inter-PLMN communication between different ProSe Function entities (PC6-PC7). These connections could be transmitted over any medium, and they should be secured by network operators just like the pre-existing PSTN (Public Switched Telephone Network) connections.
- **Security specific to 5G wireless connections:** it refers to the wireless point-to-point connection between the gNB and the UE, and the tunnel between UE and 5GC. For example, it could be some broadcast messages sent from gNBs, authentication and key agreement messages sent from gNBs or UEs, and secured communication between UE and gNB. It should

be protected by 5G-AKA or EAP-AKA' protocols. Most of the security papers related to 5G authentication fall into this class, and we will discuss it in the literature review.

- **Security specific to 5G-D2D connections:** it refers to the wireless point-to-point connection between two UEs. For example, it could be the D2D device discovery, connection establishment, authentication, and communication between different UE devices (PC5). D2D UEs may work under unlicensed bands or without cellular coverage, and it is not protected by pre-existing 5G security architecture. Most security papers related to D2D security are classified into this class, and they will also be discussed in the literature review.

1.1.3.3. Security Threats

5G mobile devices nowadays contain much sensitive information such as personal identity, financial and health data, thus, network security has been unprecedentedly important to avoid any data leakage or misuse. However, as 5G-D2D has the unique network topology which connects both ad-hoc networks (PC5) and centralized network (PC3) wirelessly aforementioned, it is vulnerable to more security threats pre-existed in both ad-hoc network and cellular network including [11], [14], [15], [20], [21]. To conclude, the security threats in 5G and 5G-D2D are:

- **Data Fabrication:** an active attacker can create fake data in order to manipulate other UEs.
- **Data Modification:** an active attacker can change the data sent from a specific user to falsify other users.
- **Eavesdropping:** a passive attacker can intercept the communication channels to read user data between two UE devices, or the control messages between UE and the core network.
- **Impersonation:** an attacker can pretend to be another UE device to get access to the network.
- **Jamming and Denial of Service (DoS) Attack:** attackers can deliberately disrupt or block the connection of normal users in a network.
- **Privacy Violation:** D2D services such as device discovery may contain identity data, so an attacker can trace the device by continuously sniffing D2D broadcasts.

These are some security threats exclusively for 5G-D2D:

- **Freeriding:** some selfish users only receive content from others but refuse to help others relay data in order to save battery. The availability and service quality of the D2D network may be severely affected.
- **Location Spoofing:** an active attacker may send out malicious location information to confuse existing D2D users or even disrupt D2D group forming.

1.1.3.4. Security Requirements

To resolve the security concerns, research related to 5G and 5G-D2D security requirements have been carried out extensively. To the best of knowledge, academicians have summarized the following major security requirements for a secure 5G and 5G-D2D communication system [14], [15], [21], [22]:

- **Data confidentiality:** be able to prevent data leakage or eavesdropping.
- **Data Integrity:** be able to detect and prevent data modification.
- **Authentication:** be able to verify the identity of UEs and prevent from impersonation attack.
- **Fine-grained access control:** be able to create data access rules or policies in small granularity such that specific unauthorized users can be barred from accessing private data.
- **Revocability:** be able to deprive privileges of D2D UEs if malicious behavior is detected.
- **Availability and Dependability:** be able to service even under jamming, DoS or DDoS attacks.
- **Non-Repudiation:** be able to prevent UEs from denying the authorship or reception of message.
- **Privacy or Anonymity:** be able to hide identity or location information of UEs.
- **Traceability:** be able to identify and the source of false message.

1.2. Motivation

While the 5G network is providing many benefits, it also poses more security challenges. Acknowledged by the potential security issues, the Third Generation Partnership Project (3GPP) has standardized authentication and key agreement procedure 5G-AKA in TS 33.501 [23]. Although it has resolved some pre-existing security issues found in the 4G LTE network, some security issues such as device anonymity, session key perfect forward secrecy, and single-point-of-failure remain unresolved. There are also many researchers proposed different protocols to improve security in 5G authentication. However, some of them could not deter DoS and DDoS attacks effectively as stated [24]–[26], some of them could not provide session key perfect forward secrecy [27]–[30], and some of them are computationally expensive [31], [32].

On the other hand, as one of the promising features in 5G, D2D communication allows adjacent mobile devices to communicate directly without relaying user data over the base stations. However, 3GPP has not standardized the security mechanism for establishing one-to-one, one-to-many, or public safety relay D2D connections in 5G. Although there are many researchers proposed different D2D related protocols for cellular networks, some of them are inapplicable to the 5G scenario because of the different system model assumptions [33], [34]. There are also some other proposals remained impractical because of hardware limits or it is computationally inefficient to implement in resource-constrained mobile devices [35]–[37]. For some proposals that achieved rationally low computational

overhead, their security levels could also be improved to prevent specific kinds of attacks in the D2D network [38], [39]. Consequently, there is a need to introduce novel mechanisms that protect the security of UEs under 5G cellular networks and 5G D2D communication efficiently.

1.3. Objectives

To enjoy the advantages of 5G and the new applications brought by 5G-D2D, it is essential to overcome the security threats efficiently. Therefore, the objective of this research is to design secure mechanisms for 5G and 5G-D2D that address various kinds of requirements. Specifically, the proposed mechanisms should satisfy the following requirements:

- **Mutual Authentication and Key Agreement:** both two devices should authenticate each other correctly and exchange a session key securely without revealing to adversaries.
- **Device anonymity protection:** the permanent identity of any UE should be hidden all the time, such that adversaries cannot read the message to trace the device location easily.
- **Safe against malicious active and passive attacks:** active attacks such as impersonation, data fabrication, DoS/DDoS attacks should be detected and prevented. Passive attacks such as eavesdropping, and location tracking should be prevented.
- **Energy and Computationally Efficient:** save power by reducing transmission and computational costs.

For 5G-D2D authentication, these extra requirements should also be satisfied:

- **Safe against D2D specific insider attacks:** since there are many D2D devices in the network, some device may be malicious devices which perform free-riding attacks. It should be detected and prevented such that the usability of the 5G D2D network can be maintained.

1.4. Contribution

The contribution of this thesis is to review and improve network security in 5G networks with new authentication schemes. First, to review the security issues in 5G holistically, an extensive literature review related to the security architecture of 5G and 5G-D2D, and the major research work in these areas are carried out. Then, the main contributions of this research work are presented as follows:

- To improve the security in 5G authentication, a Secure Blockchain-based Authentication and Key Agreement Protocol for 3GPP 5G networks (5GSBA) is proposed. Using the blockchain technology and the one-time hash function, the 5GSBA ensures mutual authentication and prevents the Denial of Service (DoS) and the Distributed Denial of Service (DDoS) attacks.

- To further improve the security in 5G D2D networks, we propose a Lightweight Traceable Authentication and Key Agreement protocol (LT-AKA) for 5G D2D communication. LT-AKA ensures secure mutual authentication of one-to-one D2D connections in 5G networks. It also prevents free-riding attacks that hamper the service quality of the network.

To ensure the proposed protocols satisfy the security requirements listed in Chapter 1.1.3, security evaluations based on both logical proofing, formal analysis, and quantitative reasoning are carried out for each proposal. Finally, performance evaluations on both protocols showed that they improved the security with tiny performance overheads compared with other existing schemes.

1.5. Organization of the Thesis

The rest of the thesis is organized as follows: Chapter 2 reviews the major existing works related to 5G and 5G-D2D authentication. Chapter 3 introduces the 5GSBA scheme for 3GPP 5G networks in detail. Chapter 4 introduces the LT-AKA scheme for 5G-D2D. Finally, in Chapter 5, a conclusion and recommendations for future work are discussed.

Chapter 2. Literature Review

Many different solutions are proposed to mitigate different specific security issues in 5G and 5G D2D. In this literature review, the first part would focus on the security issues of the latest 3GPP 5G authentication and key agreement scheme (5G-AKA) and some existing solutions for 5G authentication. Then, in the second part, we would focus on the solutions designed for 5G D2D ad-hoc connections, which refers to the wireless point-to-point connection between two UEs. Finally, we would overview the existing formal verification techniques.

2.1. 5G Networks

5G authentication in 5G networks refers to the mutual authentication process between the UE and the 5GC. This process should run at least once during the connection establishment (e.g., booting the UE or leaving the airplane mode). Recently, 3GPP has standardized two authentication protocols in the TS 33.501 [40], namely the “5G Authentication and Key Agreement Scheme (5G-AKA)” and the “Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA’).” In this chapter, we would focus on 5G-AKA only because it is the default protocol for 3GPP Access (i.e., the UE is using 5G-NR to connect with the 5G core). For EAP-AKA’, it is omitted because it is designed for non-3GPP Access (e.g., Wi-Fi connections), which is out of the scope of our discussion.

2.1.1. 5G-AKA Scheme

2.1.1.1. Overview of 5G-AKA

Enhanced based on the architecture of EPS-AKA in the LTE security architecture, 5G-AKA aims to ensure the authenticity between UE, serving network, and the home network. Figure 6 shows a simplified sequence diagram of the 5G-AKA authentication procedures.

Figure 6 - Sequence Diagram of 5G-AKA Authentication

Compared with the authentication scheme in 4G (i.e., the EPS-AKA), 5G-AKA has the following highlights:

- Introduction of 5G-SUCI:** in the EPS-AKA, there is no encryption mechanism to protect the permanent identity (i.e., the IMSI) of UE. Consequently, UE reveals its IMSI during the first-time registration with the network. Even though temporary identity GUTI is introduced, it fails to protect the UE: if the core network lost the records of GUTI, or the UE has never been used before, UE has to use its IMSI to start authentication anyway. In 5G-AKA, 5G-SUCI is introduced to conceal the IMSI (or SUPI) using the ECIES encryption scheme. Thus, the device anonymity of UEs has greatly improved.
- Separation of functional entities:** in EPS-AKA, critical key information can be stored in any functionalities in the HSS. However, in 5G-AKA, it has explicitly stated that all critical key information can only be stored in the dedicated functional entity called UDM/ARPF. Also, other serving networks have no way to access it directly. By doing so, network administrators can simply focus on improving data security in one server, reducing the chance of key leakage in the future.

- **Traceability of Authentications in Serving Networks:** in EPS-AKA, serving networks can confirm authentication without acknowledging the home network. Thus, the home network loses the traceability of authentication success events. In 5G-AKA, the RES message from SEAF to AUSF is made compulsory, such that AUSF in the home network is always fully acknowledged about the authentication success events between the serving networks and the UEs.

2.1.1.2. Security Vulnerabilities of 5G-AKA

Although 5G-AKA has provided stronger security than EPS-AKA, researchers have still found some security vulnerabilities in the protocol, making it less secure as it has claimed. For example, [41] and [42] have found that 5G-AKA suffers from linkability attacks: adversaries can use the synchronization error messages (MAC_FAIL and SYNC_FAIL) to detect if the UE is currently located in a certain area. Also, 3GPP TR 33.846 [43] has found that 5G-AKA fails to prevent denial-of-service (DoS) attacks because the core network has no way to justify if the SUCI is a replayed message. 5G-AKA is also a centralized protocol that heavily relies on the authentication functional entities AUSF/ARPF, so it could be vulnerable to Distributed DoS (DDoS) attacks and the single-point-of-failure issue in the AUSF/ARPF. Moreover, [44] has found that 5G-AKA fails to provide perfect forward secrecy and post-compromise secrecy due to the use of the long term symmetric key and the sequence number. In fact, according to the 3GPP TS 33.501 [23], the device anonymity protection of UE in 5G-AKA is also vulnerable: network operators can voluntarily opt-out the encryption in the subscription concealed identifier (SUCI) that encrypts the Subscription Permanent Identifier (SUPI) of UEs. In this case, UEs sends the cleartext of the SUPI through the wireless channels (i.e., the “null-scheme”), which could be dangerous for IMSI-catching attacks. What is worse, in some emergent situation, UEs are also allowed to send its SUPI directly to initiate authentication procedures. To conclude, 5G-AKA is vulnerable to many network attacks including but not limited to linkability attacks, DoS attacks, and IMSI-catching attacks. The lack of perfect forward secrecy also makes 5G-AKA vulnerable to session data recovery if the long-term key is compromised in the future.

2.1.2. Existing Solutions for 5G Authentication

In this section, we divide the existing solutions into two groups, the conventional authentication and key agreement (AKA) schemes, and the blockchain-based authentication schemes.

2.1.2.1. One-to-one Authentication and Key Agreement (AKA) Schemes

Since 2012, J. Cao et. al. [45] have proposed a simple and robust handover authentication scheme for both heterogeneous eNB (HeNB) and eNB in the LTE networks. Since the system model in the proposal

is based on the 4G EPS network, it should be easily reusable in the 5GC network. This proposal not only proposed handover authentication, but also an authentication procedure for all UEs. To start mutual authentication, the IMSI of UE is sent to the MME and then the HSS of the network. Then, HSS issues a proxy signature specifically for the UE and sends back to MME. After the MME verified the UE using the authentication procedures similar to EPS-AKA, this proxy signature will be sent to the UE for the subsequent handover authentication. For the handover authentication, UE sends the proxy signature to the destination eNB to start handover requests, and the same procedure is used for both S1-based and X2-based handover. Security evaluation shows that the scheme is resistant to replay attacks. However, since the IMSI is directly sent over the air during authentication, the protocol suffers from IMSI catching attacks.

In 2017, El Idrissi et. al. [46] proposed an efficient authentication protocol for the 5G heterogeneous network. In the proposal, the existing EAP-AKA protocol is improved by adding Elliptic Curve Diffie-Hellman (ECDH). Hence, for most of the operations, it works similar to EAP-AKA. Then, for the handover authentication, the proposed protocol generates the handover key TK_W on top of the UE shared key TK_H . Thus, a new key can be derived without the intervention of MME (i.e. the SEAF in 5GC) and without using the original pre-shared secret of UE. The security evaluation with AVISPA shows that the protocol can achieve injective agreement successfully, and the ECDH ensures session key perfect forward secrecy. The performance evaluation shows that the improved EAP-AKA generates fewer transient keys than the unmodified version. Also, the bandwidth consumption has been reduced by 50% compared to the unmodified version. Since this protocol is built based on the EAP framework, it should work in a heterogeneous network with different kinds of connection medium. However, it could also generate excessive bandwidth and suffer from DoS attacks because of the generic EAP framework design.

Khan et. al. [32] have proposed a Privacy Enhanced Fast Mutual Authentication (PEFMA) scheme in 5G networks based on identity-based cryptosystems. Similar to 5G-AKA, the proposal suggests encrypting the SUPI of UE to conceal the device identity. However, unlike 5G-AKA that uses ECIES with a fixed public key, the proposal suggests using identity-based encryption (IBE) to encrypt the SUPI with on the serving network ID, then the serving network responds to the challenge using an identity-based signature with his own network ID. Finally, the home network computes the session identity-based encryption key of the serving network using his private key and the information provided by the serving network, such that the home network can decrypt the SUCI of UE. To reduce the computational overheads, the authors also suggested running PEFMA only if the TMSI (or GUTI in 5G) is unknown. Security analysis shows that it can protect device anonymity and reduce the management costs on the PKI infrastructure. However, since IBE involves bilinear pairing operations, this scheme could be computationally expensive for most low-end smartphones and IoT devices to run.

Recently, Gharsallah et. al. [28] have proposed a Secure Efficient and Lightweight Authentication protocol (SEL-AKA) for 5G networks. In the proposal, a symmetric key K stored in UDM/ARPF is shared to each device, then it can encrypt the SUPI using this symmetric key to generate SUCI. Also, the authentication request messages are signed using HMAC. The proposed scheme follows most of the procedures in the 5G-AKA, but all asymmetric key operations are replaced with symmetric keys. The security evaluation with AVISPA shows that it can achieve mutual authentication while protecting the session keys (i.e., the K_{SEAF} and K_{AUSF}). Also, security analysis shows that the protocol is resistant to DoS attacks and replay attacks. However, since the symmetric key is shared among many UEs, the system is vulnerable to insider attacks that some compromised UEs in the network can decrypt the SUCI or even impersonate a specific UE with the same symmetric key.

Alezabi et. al. [47] have proposed a set of efficient authentication protocols and a set of re-authentication protocols for 4G and 5G heterogeneous networks. In the proposal, standard protocols such as EAP-AKA', INEA, and EPS-AKA are enhanced using a Pseudo-ID (UE_{ID}) and the table in HSS to hide IMSI. Also, a new key hierarchy is created to support all these protocols to work together. Consequently, the heterogeneous networks running different protocols support vertical handover between LTE-Advanced, WLAN, and WiMAX connections. Security analysis with AVISPA shows that it protects message integrity, device anonymity, and forward and backward secrecy. However, since the session key is still derived from the sequence number and the permanent key, if the permanent key is stolen in the future, all session keys should be vulnerable. Performance analysis shows that both the handover delays, key size, and energy consumptions are minimal compared with the unmodified protocols.

Braeken et. al. [27] have proposed a novel 5G authentication protocol aiming to prevent active attacks and malicious serving networks. In the proposal, it reveals that 5G-AKA has these vulnerabilities: it suffers from linkability attacks, its sequence number (SQN) is easily leaked, and its security would be completely broken if the serving network is compromised (this one of the pre-assumptions in the 3GPP standard). To fix these issues, the proposal completely deprecates the sequence number (SQN) in 5G-AKA and employs a new random nonce ($R1$) as the challenge-response material. In the protocol, the serving network generates $R1$ and broadcasts to all UE. When a UE starts a new authentication request, it generates another random nonce $R2$, appends with the received $R1$, its SUPI, and the name of the serving network. Then, it encrypts the request using the public key of the home network. Hence, the serving network does not know about the SUPI of the incoming UEs, and the protocol can be greatly simplified because there is no need to synchronize the sequence number. The security evaluation with Rubin Logic shows that it achieves mutual authentication with injective agreements, and the security analysis shows that it can prevent linkability attacks in 5G-AKA, and parts of the session key forward secrecy can be protected. However, the practicality of this protocol remains uncertain, because the serving networks should have the SUPI of incoming UEs for consecutive transactions, such as the

assignment of GUTI, tolling, and so on.

Apart from the novel authentication protocol aforementioned, Braeken [48] have further proposed another symmetric key-based 5G-AKA authentication recently. Similar to the [27], this protocol also aims to protect the device anonymity and unlinkability with low computational costs. To mutually authenticate between UE, SN, and HN, UE initiates authentication requests with two different formats: if the SN or HN did not report desynchronization, and the sequence number in local UE is preserved, it sends the synchronized request with fewer parameters. In the case of desynchronization, UE sends the desynchronized request which contains the hashed secrets and the XOR-ed sequence number n . By doing so, UE does not send synchronization errors upon receiving responses from SN, so the linkability attacks in 5G-AKA no longer exist. Security analysis shows that it prevents replay attacks, linkability attacks, MITM attacks. However, the proposed protocol could only provide part of the session key forward secrecy, and it may not prevent DoS attacks because the desynchronized request could be replayed to flood the SN and HN. Performance analysis shows that the proposed protocol has much lower communication overheads than 5G-AKA because only 2 signaling messages are needed.

F. Leu et. al. [49] have recently proposed a fault-tolerant 5G authentication scheme (FT5AS) in the 5G networks. Unlike most proposals suggesting an authentication protocol, this proposal suggests a mechanism to use many AUSF servers to make an AUSF pool. Then, a mediator server is installed to monitor and manage that AUSF pool. This mediator controls all AUSF using an SDN controller, and it uses heartbeat signals to detect any failure in the AUSF. To make sure the failed AUSF will not interrupt the ongoing authentication sessions, a distributed topology (DT) storage is also facilitated for storing the AUSF status table. This table contains the current ongoing authentication status and information, such that other working AUSF can reuse the information to respond to the requests from UEs even if the responsible server is accidentally down. Simulation results show that FT5AS achieved fairly high throughput among all existing implementations, and it generates relatively low end-to-end delays. From the security aspect, this scheme can improve the availability of 5G authentication. However, since 5G-AKA is vulnerable to DoS attacks and the resource of the AUSF pool is finite, a stronger protocol should still be considered to efficiently improve the availability of 5G authentication at much lower costs.

Hu et. al. [29] have proposed a security-enhanced 5G authentication scheme for insecure channels. In the proposal, unlike 3GPP standards that pre-assumed all channels between SN and HN are protected using IPsec tunnels, it assumes that there is no protection between the channel between UE and SN and SN to HN. Then, it suggested that UE should generate one ephemeral public-private key pair, and the public key is included in the SUCI of the authentication requests. Also, all transactions between SN and HN should be protected by asymmetric encryption or symmetric key (K_{SEAF}) after authentication. To prevent linkability attacks in 5G-AKA due to synchronization failure, UE aborts the protocol execution if it found the MAC and SQN are not matched. Also, the authentication response from SN to UE is

encrypted with the ephemeral key generated by UE. Security analysis shows that this protocol should achieve mutual authentication, but no performance evaluation is presented in the paper.

2.1.2.2. Group Authentication and Key Agreement (AKA) Schemes

In 2013, C. Lai et. al. [24] have proposed a secure and efficient group authentication and key agreement protocol (SE-AKA) for LTE networks. It is compatible with the 5GC networks due to the similar system model, and it supports both one-to-one and group-based authentication in the cellular network. To mutually authenticate multiple devices, it should firstly choose one UE as the group leader. This group leader should go through a full authentication with both MME (i.e., the SEAF in 5GC) and HSS (i.e., the ARPF in 5GC). Then, the group leader UE will be assigned with one ECDH session key, and the MME will have one group temporary key (GTK). Subsequently, the remaining group member UEs can authenticate much faster because they only need to authenticate with the MME using the group temporary key. The security evaluation with ProVerif shows that the protocol can mutually authenticate UEs and HSS. Also, it is resistant to replay attacks, DoS attacks, MITM attacks, redirection attacks (i.e., impersonation by forwarding the messages). The use of ECDH also enables session key perfect forward and backward secrecy. The performance analysis also shows that SE-AKA consumes less computational delays when the number of devices in the group is large. However, due to the use of ECDH, its computational overhead is higher than the symmetric-key based EPS-AKA.

In 2019, J. Cao et. al. [25] have proposed a lightweight privacy-preservation access authentication scheme for massive devices in the 5G cellular networks (LPPA). It improves the security vulnerabilities found in the LGTH [50] scheme designed for 4G machine-type communication (MTC) networks. In particular, LGTH cannot protect the identity privacy of UEs, and it is found to be vulnerable to interior forging attacks and DoS attacks. To solve the security issues, LPPA encrypts the identity of all UEs during authentication. Then, to authenticate many devices at the same time, many devices choose a group leader to communicate with the MME (or SEAF in 5G). After that, MME will verify the aggregation MAC obtained by the group leader to mutual authenticate all devices. Finally, after several negotiation steps, a distinct session key between each terminal device (TD) and the MME can be derived locally. To prevent the DoS attack, an aggregation time window interval is introduced during the authentication, such that TD could not send access request messages too often. The security evaluation shows that LPPA prevents replay attacks, eavesdropping, man-in-the-middle (MITM) attacks, interior forgery attacks, and some DoS attacks. Performance evaluation shows that although it is not the fastest protocol among other schemes, it can withstand attacks much better than other existing schemes.

Enhanced based on the LPPA scheme [25] aforementioned, J. Cao et. al. [30] have proposed another lightweight and secure access authentication scheme (LSAA) for both UE and massive MTC (mMTC) devices in the 5G networks. LSAA provides a faster UE registration, and a faster mMTC device registration through group-based authentication compared with LPAA. Like LPAA, a group leader is

chosen to start massive device authentication. However, LSAA employs a more computationally lightweight Extended Chebyshev chaotic maps to do Diffie-Hellman key exchange. To simply put, a chaotic map is a technique that chooses a map and finds a starting point from it. Then, the adversaries know nothing about the content if he doesn't have the starting point. Besides of chaotic map, the Chinese remainder theorem is employed in the acknowledgment message to massively notify many devices about the session key establishment at one time. Finally, aggregation MAC is also used for message authentication. Security analysis with ProVerif and Scyther shows that the scheme is resistant to replay attacks, MITM attacks, eavesdropping attacks. It also provides session key perfect forward secrecy and device anonymity. However, the DoS and DDoS attacks prevention are not mentioned in the paper. Performance analysis shows that LSAA is faster than 12 existing schemes including 5G-AKA, EAP-AKA, SE-AKA [24], and so on. Nevertheless, since the implementation of chaotic map Diffie-Hellman may not be optimized in most mobile platforms, the actual performance overheads could be more than the stated values.

P. Yu et. al. [26] have proposed a quantum-resistance authentication and data transmission scheme for the narrow-band Internet of Things (NB-IoT) in 3GPP 5G networks. To provide quantum attack resistance, the proposal has employed lattice-based encryption in the aggregation signcryption (i.e. encryption and signature). Unlike most schemes that generate a symmetric key with the protocol, this proposal encrypts the message and sign it directly because NB-IoT should only transfer a small amount of data. Specifically, a group leader should be chosen among all NB-IoT devices, then, messages will be encrypted with AMF's public key and signed using the device's private key. After that, the group leader receiving all messages will be aggregate them into a single message and send it to the AMF. Finally, the AMF will respond with a downlink message, and the group leader will also forward it to all group members. All group members should be able to decrypt the message and verify the signature of it. Security analysis shows that it prevents replay attacks, quantum attacks, and eavesdropping. It also guarantees device anonymity by encrypting the ID in the ciphertext. Performance evaluation shows that it consumes much less computational overheads and signaling messages than other existing group-based authentication schemes including LGTH [50] and SE-AKA [24]. However, as this protocol assumes that the cleartext should be short, it could potentially take a much longer time when the messages are large.

2.1.2.3. Handover / Re-Authentication Schemes

J. Choi et. al. [51] have proposed the use of chameleon hashing to do authentication and handover re-authentication. Specifically, UE calculates one fresh message (M, X) and its chameleon hash and sends it to the AAA server during initial authentication. Then, the AAA server returns the signature of that hash to the UE. During handover, UE modifies the M , calculates the X using the chameleon hash trapdoor, and sends the new (M, X) with the signed chameleon hash to the new AP. After verifications in the new AP, the AAA server connected to the new AP will finally return the signed authentication

response for the UE. Consequently, mutual authentication can be achieved. Performance analysis shows that it consumes much less energy than the IBC-based schemes. However, this paper did not include security evaluation, so the reliability of the proposed protocol is uncertain.

Yang et. al. [52] proposed a way to use physical characteristics of the link to perform fast handover authentication in the 5G SDB-based network. Specifically, the proposal suggested using the Secure Context Information (SCI), as known as link signature, to perform re-authentication with new APs. First, UE measures the link signatures of the connection between UE and the old AP (H_{UE1}), and the connection between UE and the new AP (H_{UE2}). Then, we can take the average of them to calculate a MAC. Similarly, the new AP will also measure the same connections (i.e., the H_{AP1} and H_{AP2}) and take the average to calculate an XMAC. If the calculated MAC is equal to the XMAC, it is highly likely that the UE is physically located at the desired position. Hence, we can conclude that it is highly likely that the UE is legitimate. Although it is fast and it introduces only tiny computational overheads, the security of this scheme is susceptible to wireless channel conditions. Also, since the paper did not include the details about the authentication protocol, we could not determine if the proposed scheme has any potential logical attacks.

Han et. al. [53] proposed an efficient handover authentication mechanism for 5G networks using one-way hash functions and pseudonyms. Specifically, the protocol is based on the mobile-edge computing (MEC) system model: there are many powerful MEC servers at the edge of the network, and each of them is connected with multiple gNBs. There are three different handover scenarios: first-time entering to the 5G network, switching between gNBs connected to the same MEC server, and switching between gNBs connected to different MEC servers. Among the three scenarios, only the switching between gNBs connected to the same MEC server will run the newly proposed scheme. Other scenarios will run the conventional EAP-AKA scheme. To run the newly proposed scheme, by the completion of the EAP-AKA scheme, the MEC server will return five elements to UE: a symmetric key TK, a random number m , a counter indicating the number of times of re-authentication, a re-authentication ID, and a hash value $h^{\text{counter}}(m)$ which hash the m for many times. Then, if the UE wants to initiate re-authentication, it firstly sends the authentication ID to the MEC. After that, the MEC will return a random nonce and a $h^{\text{counter}}(m)$. UE will then need to supply the value of $h^{\text{counter}-1}(m)$ by doing hash operations to the m by “counter-1” times. Finally, both sides can be mutually authenticated with only hash functions used, such that it consumes much less computational overheads. Security evaluation with AVISPA shows that the protocol is secure in terms of mutual authentication, data confidentiality, and device anonymity. Performance evaluation shows that the processing delays and transmission delays of the newly proposed protocol are lower than EAP-AKA. However, since the proposal has used the same hash function and the same m all the time until the next full authentication, it could be vulnerable to birthday attacks that weaken the security of the protocol.

2.1.2.4. Blockchain-based Authentication Schemes

2.1.2.4.1. Overview of Blockchain

Linking data blocks into a long chain, blockchain technology is essentially a decentralized database solution that guarantees data immutability and practical consensus across multiple network nodes. There are three different types of blockchain platforms [54]: permission-less, permissioned, and consortium blockchains. For permission-less blockchain, this is a publicly accessible blockchain reaching consensus through a rigorous set of rules such as Proof-of-Work (PoW) and Proof-of-Stake (PoS). Some examples are cryptocurrencies like Bitcoin [55] and Ethereum. For permissioned blockchain, this is a private blockchain only accessible by the authorized users. It is generally faster than the permission-less blockchain because of its simpler consensus algorithm. For instance, Hyperledger Fabric is an enterprise private blockchain solution that can reach consensus much faster than most public blockchains [56]. For consortium blockchain, this is a hybrid private blockchain that is only accessible by authorized users, and multiple administrators are allowed to manage the consensus in the network. Among all three types of blockchains, it is envisioned that the private and consortium blockchains are the most suitable distributed solutions to solve the challenging problems in 5G because of their efficiency [57]. In recent years, blockchain has also improved security in many areas such as the 5G vehicular network [58], mobile edge computing [59], mobility management [60], [61], and so on.

2.1.2.4.2. Existing Solution: Authentication

J. Xu et. al. [62] have proposed an Identity Management and Authentication scheme based on redactable blockchain. The proposed system does not limit to the 3GPP system model, and it allows mobile users to upload and delete their own identity and public key information in the consortium blockchain managed by many operators. To upload his identity information, the user should generate their information and then send it to some issuers trusted by the network operators. Then, operators can add the entry into a redactable blockchain constructed with chameleon hash. Chameleon hash is a hash function with a trapdoor. Having the trapdoor enables easier hash collision searching. When the operators want to revoke the confidential information in that entry, he can calculate the hash collision using his trapdoor, and then change the entry without modifying the chameleon signature. However, this scheme is not an authentication protocol, it did not suggest how can mutual authentication be achieved.

Haddad et. al. [31] have proposed a blockchain-based authentication scheme for 5G networks. The proposal uses blockchain to store all UE public keys. For each block, it can store up to 6 records, and for each record, it contains a transaction number representing the pseudo-identity of the device, a UE public key for ECDSA signature operations, a timestamp, a transaction number of servicing AP (i.e. the identity of gNB), and a status code. To start mutual authentication, UE issues a request with an

ECDH public key, signed it with his own private key, and send it to the adjacent AP. Then, the receiving AP will countersign the message and forward it to the SN. Once the SN confirms the legitimacy by validating the digital signatures of AP and UE, it will respond to the request with another ECDH public key and sign it with SN private key. AP receiving the response will also countersign and forward it to UE. Finally, UE receiving the response will check the legitimacy by verifying the digital signatures in the response message, and then calculate the common secret using the ECDH public keys. The proposal did not mention what kind of blockchain (private or consortium) is used, but the use of blockchain did provide many advantages such as the immutability of data and prevention of single-point-of-failure. Security evaluation shows that it prevents eavesdropping, MITM, DoS, DDoS, and replay attacks. It also guarantees session key forward secrecy and device anonymity. Although the performance evaluation shows that its computational overheads could be lower than SEL-AKA [28] and 5G-AKA [23], it remains dubious because asymmetric key-based signature should be theoretically slower than the symmetric key-based KDF. Also, the size of the transaction number in the protocol does not fit well with the GUTI in the 3GPP 5G security architecture. As a result, more storage overheads are expected if it has to run in the 3GPP 5G core networks.

Yang et. al. [63] have proposed a blockchain-based anonymous access (BAA) scheme in the Cloud Radio over Fiber Networks (C-RoFN). The proposal suggested a blockchain-based trust architecture (BTA) that agrees between equipment manufacturers, users, and network operators. To start mutual authentication between the user and the operator, the device manufacturer must firstly register the public keys of the device into the operator's blockchain. Then, the device can request keying materials (i.e. materials for challenge-responses) from the manufacturer through the fiber links of operators. If the request is legitimate, the manufacturer will send responses to the device, so that it can initiate a further verification with the operator. Finally, some zero-knowledge proof steps will also run to make sure both the operator and the device possess the same materials given by the manufacturer. After successful authentication, a record will also be created in the blockchain. Simulation using Open vSwitch, VMware ESXi, OpenDaylight Beryllium, and Hyperledger Fabric shows that the design is feasible. However, since there is no detailed implementation and protocol design given in this paper, no further evaluation can be done. Hence, it can only be a proof-of-concept of the idea.

Zhang et. al. [59] have proposed a group signature scheme to let multiple parties assist the verification of blocks in the blockchain. Combining group signature into the blockchain, it can be used as a way of threshold-like data verification. It is noteworthy that the system model of this proposal is based on mobile-edge computing (MEC): adjacent devices are connected to a MEC edge server to share computational tasks. Consequently, it is different from 3GPP 5G system architecture, and it is not exactly a blockchain-based authentication protocol. The scheme suggested that a two-round process: in the first round of block validation, group members need to check and sign the block before broadcasting to other groups. In the second round, another group can receive the broadcast blocks to undergo another

round of validation. Finally, the block is verified by two groups, so it can be added to the longest chain in the MEC network. Security analysis shows that the protocol prevents common attacks in the Proof-of-Stake (PoS) blockchain, such as nothing-at-stake, bribe attacks, stake-bleeding attacks, and fake stake attacks.

Hewa et. al. [64] have proposed a blockchain-based certificate revocation mechanism for 5G IoT. Using smart contracts in blockchain as a private ledger to generate, store, update, and revoke all certificates, it makes sure that the certificate of potentially malicious devices can be revoked timely. All blockchain nodes in the scheme are the fog nodes so that they can also be fog servers for monitoring the traffic in the network. Moreover, to reduce its computational overheads of using asymmetric cryptosystems in IoT devices, the lightweight Elliptic Curve Qu Vanstone (ECQV) implicit certificate is chosen. Simulation using Hyperledger Fabric shows that certificate generation, access, and revoke are faster when blockchain is used. However, since it is not an authentication protocol, the way of using these certificates securely should be further researched.

2.1.2.4.3. Existing Solution: Handover / Re-Authentication

Chen et. al. [65] have proposed a security authentication scheme for the 5G ultra-dense network. The proposal has proposed an “AP Group Improved Practical Byzantine Fault Tolerant (APG-PBFT)” algorithm to solve handover problems in the 5G networks with many access points. Roughly speaking, the algorithm has 5 steps: first, UE should send an authentication request to the adjacent AP. Then, the AP will request the generation of an AP group. Second, AP will select some trusted APs based on their consensus history. Third, the AP group will be formed, and all participants will generate a private blockchain. Fourth, the AP will finish mutual authentication with UE using conventional authentication protocols (e.g., EPS-AKA). Fifth, the authentication results will be shared within the private blockchain. By doing so, every time the user has left the coverage of the original AP, the adjacent AP can use the information stored in the blockchain to pick up the connection immediately, without doing the time-consuming re-authentication. To extend the coverage, the APG can simply choose more APs to join the APG in the future. Performance evaluation shows that the APG-PBFT algorithm works better than the original PBFT with a slightly lower delay. However, the proposed scheme only employs blockchain for accelerating the handover. The authentication time can potentially take longer because of the APG creation.

Zhang et. al. [60] have proposed a robust and universal seamless handover authentication scheme for the 5G heterogeneous network. This proposal is a universal authentication scheme that works for both 3GPP access including intra-domain (X2-based), inter-domain (S1-based), and non-3GPP access. After initial authentication, each UE will be assigned with one chameleon hash trapdoor (i.e. the SK_{UE}), and the calculated chameleon hash will be uploaded to a consortium blockchain accessible by all APs. This record should be updated at every initial authentication because the pseudo-identity of the UE will also

update. Then, every time the UE re-authenticates with a new AP, the UE should provide a message containing the pseudo-identity, a randomly generated ECDH public key, a timestamp, and a chameleon hash collision generated using the trapdoor. Then, the AP will calculate the chameleon hash stored in the blockchain to verify the request and then respond to the request with an ECDH public key. Security evaluation with BAN logic and AVISPA (OFMC Checker) has proved that the protocol achieves injective agreement and mutual authentication. It is resistant to replay attacks and MITM, and it guarantees device anonymity, session key forward and backward secrecy, traceability, and free of key escrow problems. Performance evaluation also shows that it consumes much less computational overheads than the existing re-authentication schemes, and the storage cost of the blockchain remains controllable.

2.2. 5G-D2D Networks

For the solutions specifically designed for D2D ad-hoc connections, most of them can be categorized into two approaches, including physical layer security and cryptographic techniques.

2.2.1. Physical Layer Security (PLS)

Rather than utilizing the computational complexity in security algorithms, physical layer security is an approach exploiting the wireless channel characteristics to safeguard data transmission [14][66]. Since it does not rely on computational challenging problems, the level of data secrecy will not be affected by the processing speed of potential eavesdroppers. Also, it is believed that it can reduce the complexity in key management because it does not need any cryptographic keys [66]. By leveraging both randomness in wireless medium and imperfections in hardware [67], it is also believed that even in 5G core network which involves both HetNet, massive MIMO and millimeter wave, physical layer security can still be applied to worsen the signal quality of eavesdroppers.

2.1.1.1. Conventional Cooperative PLS

2.1.1.1.1. Protecting cellular links

In 2016, R. Zhang et. al. [68] proposed a secrecy-based Joint Power and Access Control (JPAC) scheme which makes use of physical layer security to enhance data secrecy of cellular links in any cellular network supporting D2D communication. In this paper, it is assumed that spectrum resources allocation of both cellular user equipment (CUEs) and D2D user equipment (DUEs) are managed by base stations, and all DUEs are willing to participate in this PLS cooperation because they need the assigned spectrum to transmit data. Then, to differentiate which D2D pairs should be chosen for improving the data secrecy of a CUE, all CUEs and DUEs will be assigned into some resource blocks. After that, transmission power of both CUEs and DUEs within the same resources block will be jointly calculated and controlled

to an optimal level, such that the optimal level of data secrecy of CUEs in that resource block can be achieved. Simulation showed that the proposed scheme can achieve higher data secrecy than the case without cooperation in various use cases. Besides, since the algorithm has a relatively low complexity of $O(K(M+N))$, it would be easy for mobile devices which have limited battery and computational power to deploy. However, it is also pointed that due to the interference from CUEs, the higher transmission power of CUEs will result in lower D2D transmission rate of the entire network. Also, this scheme has only focused on the data secrecy of CUEs, therefore, the data secrecy of DUEs is not covered.

Later, L. Wang et. al. [69] proposed another access selection scheme that allows multiple DUEs to reuse the same spectrum as CUEs to provide PLS to cellular links. Unlike the aforementioned scheme, this scheme does not control the power or channel gain of D2D pairs. Instead, using an iterative algorithm called “advance-and-retreat”, the cellular network can find the feasible optimal threshold that reflects the optimal distance between CUEs and DUEs. Then, multiple eligible D2D connection pairs will be selected as “protection pairs”, and only these pairs can use the same frequency spectrum as the CUE intended to protect. Therefore, during data transmission of these D2D pairs, their signals will also become jamming signals of CUE signals. As a result, eavesdroppers cannot easily listen to the original signals issued by CUE, so the data secrecy for that CUE can be improved. Since this scheme does not limit the transmission power of DUEs and CUEs, it is believed that the transmission capabilities of the D2D links and the CUE links will not decrease. However, since the proposed scheme has only covered the case that only one CUE is protected by multiple DUE pairs, feasibility of this scheme is yet to confirm because there would be a tremendous number of CUEs in 5G HetNet. Besides, this scheme has only discussed the data secrecy protection of CUEs, the data secrecy of D2D links is not guaranteed. Also, this scheme could be vulnerable to various attacks such as free-riding attacks by DUEs, resulting uncertainties in achieving high data secrecy level of the CUE.

2.1.1.1.2. Protecting both cellular links and D2D links

In recent years, D. Ying and F. Ye [70] another D2D PLS scheme which controls the transmitting power of the DUEs to achieve high data secrecy. In the paper, they proposed one D2D scenario which all students are sitting together and using their mobile phones to watch basketball games, so D2D relays are used to offload data flow from the base station. In this scenario, all DUEs are using the same frequency spectrum in the same area, so the transmission signals between D2D relays will also be the source of interference to other D2D relays. On the other hand, since we cannot identify the exact location of eavesdroppers, the paper assumes all eavesdroppers are distributed according to the Poisson Point Process (PPP). Finally, a model for computing optimal transmitting power based on this scenario is presented to adjust the transmitting power of DUEs and make sure all links can be protected equally at the edge of the D2D communication area. Simulation showed that if the transmission power of D2D relays is adequately low, an optimal balance between connection probability and the security probability

can be found, therefore high data secrecy of D2D communication can be ensured with guaranteed usability. Also, since this scheme does not require any extra jamming signal generation, no extra energy would be consumed to achieve this extraordinary data secrecy.

Apart from simply adjusting the transmitting power of D2D signals, H. He and P. Ren [71] recently also proposed another PLS scheme focusing on relay selection, jamming generation and network coding. In this scheme, instead of conventionally choosing one relay group to do D2D data transmission, two relay groups are selected at the same time. Then, one of the selected groups will be used to transmit messages, and another group will be used to lower the signal quality to eavesdroppers by generating jamming noise. For message transmission, Wyner's code is chosen to encode the message because it provides a certain amount of data secrecy when the eavesdropper's signal is weaker than the legitimate user's signal. To further tackle the case that sometimes eavesdroppers can also get good signal quality, this scheme also employed linear network coding method which separate the original message into many parts, and only the one who can receive all parts of the encrypted messages can decrypt the content. Therefore, only the legitimate user who has both high signal-noise ratio all the time and received all parts of messages properly can decrypt the message. Eavesdroppers with any parts of the message missed are impossible to decode. This scheme is particularly useful if the number of messages to be transmitted is long enough, such that network coding can be effectively employed to split the messages into many blocks. However, this scheme could be vulnerable to different attacks such as free-riding attack, because generating jamming signals requires additional power, and D2D relays do not have enough incentives to help others generously.

2.2.1.2. Social-Aware Cooperative PLS

Using social-domain knowledge to exploit performance gain of D2D data transmission secrecy, [72] proposed a scheme to provide efficient spectrum sharing, resource allocation and PLS to all CUEs and DUEs. Assuming that social trust relationship between users is unchanged (for example A and B are always friends), and all CUEs and DUEs are using a constant transmit power, a theoretical analysis calculating the ergodic rate of the proposed social-aided scheme is presented. It showed that the overall secrecy rate of the network can be improved by 63% compared with the conventional PLS scheme without considering social-domain factors. In order to attain the performance gain, a two-stage algorithm which uses the concept of matching theory model is then proposed. This two-stage algorithm consists of "Initial Stable Matching" and "Best Response Based Matching", and it can run on DUEs autonomously. After multiple times of iteration within these two stages, it is believed that the matching results will converge, and the optimal stable resource matching can be found. Simulation results showed that comparing with random selection, the proposed scheme can improve the overall secrecy rate by 50%. Also, comparing with exhaustive search, the scheme can find the optimal matching results faster

with lower computation complexity. In fact, since this scheme can work in a decentralized network environment, it can be used even when cellular coverage is absent. However, as the paper assumed that social trust relationship is a binary value (either trustworthy or untrustworthy) and the relationship value is stable, it may not be suitable to real life situation that some friends are closer than others (which is not binary value), and the relationship may change over time which may not be always stable.

Apart from that, S. Yan et. al. [73] proposed a jamming allocation scheme which utilizes both social and physical information to form D2D clusters. Like other PLS protocols, it confuses eavesdroppers by selecting some D2D nodes in the wireless network to generate jamming signals. In this scheme, all DUEs will be categorized into some multicast clusters depending on both social constraints (such as friendship) and physical constraints (such as physical D2D links) with a greedy iterative method. At the same time, if some D2D nodes opt not to join any multicast clusters, they will act as jammers. Then, to allocate jammers to the corresponding clusters effectively, two graph-based algorithms will be used. If the number of D2D clusters is equal or larger than the number of jammers, Kuhn-Munkres (KM) algorithm will be used to do one-to-one matching between jammers and D2D users, such that every D2D cluster can get at most one jammer. For the case that many jammers are available, Hungarian Matching (HM) algorithm will be chosen instead, such that each D2D cluster can get more than one jammer. After that, D2D jammers will calculate and apply the optimal power of jamming signals to confuse eavesdroppers. Simulation results showed that using the scheme, the overall performance of data secrecy is higher than random cluster allocation which does not take social-domain information into account. However, since this paper also assumed all jammers are friendly jammers, the system is vulnerable to free-riding attack which jammers do not want to generate jamming signals anymore because of power consumption (if it is a mobile device). Also, since the paper did not discuss any details about dynamic social graph generation, it may be difficult to implement because human social relationship changes over time.

2.2.1.3. Deep-learning PLS

L. Li et. al. [74] proposed a collaborative scheme for cellular D2D network to select antenna and find optimal D2D pair using deep learning neural network. The proposal assumes that in a disastrous situation, some temporary base stations are deployed to provide temporary cellular network. Since these temporary base stations have only limited power and service capacity, all user data are transmitted through the Public Safety D2D links. All temporary base stations only assist D2D pair selection and connection establishment. In the proposed scheme, Echo State Network (ESN) is firstly used for antenna selection. As the mobility of DUEs can be more accurately predicted by the ESN network, base stations can save some time slots and energy by not choosing the useless antenna directions. DUEs can also initiate the D2D connection faster since less waiting time slots are needed to connect to temporary

base stations. On the other hand, to better predict the best D2D transmitter for D2D pair connection establishment, the proposal employs the Long Short-Term Memory (LSTM) recurrent neural network (RNN) to find the best D2D pair based on antenna, position of DUEs, position of eavesdroppers, average speeds and so on. Simulation shows that with both two neural networks used, the theoretical data secrecy of D2D users can be improved because more D2D pairs can be selected and established in the network. Also, by making more accurate predictions, those temporary base stations consume less power and therefore can be used longer.

2.2.1.4. Shortcomings of PLS

Although physical layer security has been researched recently, there are still many shortcomings yet to be solved, resulting the paradigm remains impractical. For example, [12] raised a concern that users in D2D network may not want to contribute their own device batteries in exchange of data secrecy to other users (i.e. free-riding). Besides, since PLS depends on the randomness of the network, if all devices remain static, the secrecy of transmission may not be guaranteed. Furthermore, as applications nowadays may not have a standardized way to access to physical layer parameters directly, the cooperation of PLS with application layer becomes more difficult.

2.2.2. Cryptographic Techniques

On the other hand, cryptographic techniques have been the dominant solution to fulfill most of the security requirements. Different cryptographic functions are employed and combined to improve the security of D2D communication in various scenarios, including device discovery, mutual authentication, key exchange, group-based communication and so on. Some innovative applications such as profile matching, outsource decryption and so on are also built based on cryptographic techniques and 5G D2D network.

2.2.2.1. Privacy Preserving Device Discovery

In 2015, E. Abd-Elrahman et. al. [75] proposed a D2D device discovery and D2D communication encryption scheme leveraging Identity-based Encryption (IBE) and Elliptic Curve Cryptography (ECC). The proposed system model requires all DUEs to be authenticated and under the coverage of the core network, and multiple private key generators (PKG) are built across the network. The protocol has mainly three phases: the first phase is the system setup stage which allows PKGs to generate parameters for the ECC and send it to all DUEs. Next, in the restricted discovery state, all DUEs will generate one pseudo-identity and sign the discovery message with the private key derived from PKG using the elliptic curve digital signature algorithm (ECDSA) algorithm. After that, this message will be broadcasted to other DUEs and let other DUEs to ask for public keys derived from PKGs to verify the

discovery message. In case that two DUEs are not in the same domain (e.g., two DUEs are using different network operators), DUE1 will make another request to its local PKG (PKG1), and the PKG1 will access the PKG at DUE2's domain (PKG2) to get the key parameters. Therefore, DUE1 can sign and broadcast the message that is verifiable to DUE2. The final stage is direct communication. After verification of the discovery message, DUE1 will establish a connection with DUE2 using Elliptic Curve Diffie-Hellman (ECDH) key exchange. Security analysis showed that with ECDSA used in the broadcast message, this protocol can guarantee non-repudiation of the signaling messages. Also, this protocol ensured device privacy, data confidentiality, perfect forward and backward secrecy because the pseudo-identity is updated for each session and ECDH key exchange is used during connection establishment. Subsequently, another enhancement based on this protocol is also proposed in [76] which combined the Group Key Management (GKM) mechanism with the concept of PKG.

Sun et. al. [77] proposed a privacy-preserving device discovery and authentication mechanism for 5G D2D communications. In the paper, Identity-based prefix encryption is adopted to perform identification proof and encryption. This encryption mechanism is a simplified functionality of the hierarchical identity-based encryption. Given a ciphertext, it can then be decrypted with a key in which the binary string correlated to that key is the prefix of the ciphertext. During the authentication process, DUEs need to perform full access authentication with both visiting network and home cellular network first, then they can use identity-based prefix encryption to encrypt then broadcast device discovery messages. If any DUE wants to connect to the broadcasting DUE, it has to decrypt the broadcast message using its prefix encryption secret keys, encrypt the session establishment message with Elliptic Curve Diffie-Hellman Key Exchange (ECDH) and send it back to the broadcaster. Finally, the broadcasting DUE can decrypt the received message and perform the remaining handshaking with the desired DUE. This scheme is proven to secure in D2D device discovery, mutual authentication and session key agreement without losing privacy of both DUEs, because their identity information is never broadcasted to the public without encryption, and the broadcast message cannot be decrypted by unintended users. Also, with ECDH and different random nonce used, eavesdropping and man-in-the-middle attacks can be prevented. This scheme can only be applied to in-coverage cases since the first step in this scheme is to perform authentication to the visiting network and the home cellular network. Also, the computational overhead of this scheme is relatively high because of the bilinear pairing in the hierarchical identity-based encryption.

2.2.2.2. One-to-one D2D: Authentication and Key Agreement (AKA)

2.2.2.2.1. Diffie-Hellman Key Exchange

Three similar lightweight D2D protocols are proposed in [78]. Unlike most other schemes which involve many infrastructures in the core network during authentication and connection establishment,

the three D2D protocols use only base stations as a middleman to assist D2D network establishment. The first proposed protocol uses Diffie-Hellman Key Exchange (DHKE) to exchange the encryption key over public channel and authenticate each other with another random key using message authentication code (MAC) sent through cellular channel. The second protocol is similar to the first protocol, but the base stations act as a middleman to compare the MAC encrypted code with both parties. Both two DUEs will send the encrypted message and wait base stations to send back accept or reject acknowledgement. The third protocol is the most computational complex protocol. First, two DUEs will exchange their symmetric key using DHKE in a public channel and create one secret channel. Then, base stations will send the same random key to two DUEs through cellular channel, and two DUEs needs to encrypt and send the authentication messages with keyed-hash message authentication code (HMAC). Finally, they will both use base station as a middleman to send the accept or reject message to indicate if the authentication is success. Performance evaluation showed that all three protocols are having less communication overhead than SeDS, and the security evaluation showed that they are resilient to man-in-the-middle attacks. However, since these three protocols did not use any pseudo-identity or privacy-preserving device discovery for hiding device information, the proposed protocol may not fulfil the device privacy requirement. Also, as the D2D connection establishment and authentication procedures are still reliant to base station signal, this proposal is only suitable for in-coverage scenario.

In 2017, M. Schmittner et. al. [79] proposed a Secure Multi-hop D2D communication (SEMUD) scheme which focuses on the secure connection and message forwarding. In the proposal, it is assumed that the network follows the ProSe architecture which consists of ProSe Application, ProSe Function and ProSe Application Server. For the connection part, SEMUD is comprised of 5 phases. The first phase is the subscription phase which registers the UE and takes the public key from the ProSe Application Server. The second phase is the discovery and authentication phase that follows the standard ProSe discovery and authentication framework. The third phase is the key distribution phase which two DUEs exchange public keys for mutual authentication and a DHKE symmetric session key for all subsequent communication. The fourth phase is the communication phase which messages will be traversed in single or multiple hops. The final phase is the termination phase which deregisters the DUE from the network. In fact, the main focus of SEMUD is message forwarding - during the communication phase, all messages will be appended with Blake2b hashing and then encrypted using symmetric stream cipher algorithm XSalsa20/20 and parameters from Merkle Tree. Then, the message will be forwarded to other hops based on the per-neighbor reliability metrics computed in DUEs. When the receiver DUE can receive the message, an acknowledgment message will also be returned to the sender to ensure the reliability of the transmission. Security analysis showed that the proposed scheme can survive under blackhole attacks, also the message delay can remain steady even when the number of attackers is increasing.

Combining social domain knowledge and cryptographic techniques, C. Suraci et. al. [80] proposed a mechanism that determines the trustworthiness of adjacent D2D nodes with Social IoT paradigm and distribute public keys with Diffie-Hellman Key Exchange (DHKE). Social IoT parameters refers to the activity of the device, and the objective features such as manufacturing data, location and owner of the device. In this paper, activity of the device is employed to evaluate the trustworthiness of adjacent devices. Firstly, it assumed that all DUEs want to share the information distributed from Service Provider (SP) located at the core network. According to the proposed scheme, all base stations (eNodeB) are the trusted parties which stored a list of blocked nodes. These blocked nodes will never be chosen as D2D transmitters because their credibility is too low. For other nodes, DHKE will be used to exchange symmetric key between two DUEs, and all data relayed between DUEs are pre-signed by SP. To create the blocked list, during system initialization process, each new D2D node needs to send a real identity to core network, then the network will return a pseudo identity and a trust value to the DUE. For freshly connected DUEs, the trust value will always be 0.5 (half trustworthy). After that, during daily network traffic, two credibility counters called "Good Behavior Amount (GBA)" and "Malicious Behavior Amount (MBA)" will be counted by the SP according to their behaviors. As a result, a value called "Absolute Player Trust Value" will be derived from all the parameters above. Only the devices that with the trust value higher than a defined threshold can be treated as trustworthy device, otherwise they will be put into blocked node list. Therefore, all encrypted network traffic will only traverse in those socially and behaviorally credible nodes. Simulation results also show that malicious D2D nodes can be avoided with the proposed scheme, therefore the overall data secrecy can be improved. However, since this paradigm has only considered the past behavior of the D2D node, more social IoT parameters such as the social relationship of the owners should also be considered. Also, since the paper assumed all DUEs are just relays of SP, it may not be suitable to other D2D scenarios such as file sharing, P2P gaming and so on.

W. Shen et. al. [81], [82] proposed a D2D key exchange and authentication protocol based on Diffie-Hellman Key Exchange (DHKE) and out-of-band channel authentication. Although this scheme is not designed for 5G cellular network but Wi-Fi direct network, this protocol is designed to work without using any additional infrastructure such as base stations, so it is possible that it can be improved to support the out-of-coverage connection scenario in 5G D2D. In the proposal, after D2D device discovery and channel negotiation steps are completed, DUEs will start key exchange and authentication using Short Authentication String (SAS): First, both two DUEs will generate their random secret for DHKE and a random variable for SAS. After several times of exchange in the negotiated unencrypted channel, both DUEs can compute a common DHKE symmetric key and common SAS value which looks like a random nonce. After that, to authenticate if the connection is free of man-in-the-middle attack, DUE users need to confirm the common SAS value displayed on their DUEs. If the displayed SAS values are same, the connection should be safe enough. To further reduce

any unauthorized connections due to human error, this common SAS value is designed to be short enough, so that it is visually and verbally easy to compare. It is also suggested that this common SAS value can be transferred using near-field communication (NFC) or QR code to reduce human intervention. Demonstration using Wi-Fi Direct and Android shows the protocol is implementable on modern mobile devices, and the security level is much higher than using the existing WPS encryption in Wi-Fi Direct. As a result, both data confidentiality and authentication can be guaranteed in this scheme. However, since the scheme would send the identity of DUEs over unencrypted channel, privacy or anonymity of DUEs are still under threats.

In 2017, M. Wang et. al. [83] proposed an in-coverage D2D AKA protocol which works on both roaming/non-roaming and intra/inter-operator D2D communication in 5G cellular network. In the proposal, it is assumed that the core network is 4G LTE, and all DUEs can connect to their own HSS. The protocol has 4 stages. The first stage is system setup stage which initialize all parameters for cryptographic algorithms. The second stage is UE registration. During this stage, UEs need to create a fixed or random pseudo identity, then send this identity information with real identity to VN (for roaming case) or Authentication Centre (AuC, for non-roaming case). Then, VN or AuC will start normal EPS-AKA (the authentication and key agreement protocol used in 4G) to authenticate and register the UE. The third stage is device discovery and random nonce exchange. During this stage, any device discovery method can be used, and a random nonce should be exchange in any means such as face-to-face or social networking. The final stage is D2D session generation. After having all the parameters such as pseudo identity and the random nonce of each other, DUE1 need to initiate “key agreement request” to HSS/VN to verify DUE2, and then exchange their session key using a DHKE with validation check provided by HMAC. Finally, two DUEs can authenticate each other, and one common session key can be generated using the XOR result of DHKE common secret and the broadcasted random nonce. Unlike CP-ABE or some other IBE method, this protocol does not need any key generation centers, therefore it does not have any key management overhead. Also, since only HMAC and DHKE is used, the computational cost of the protocol remains practically low. Security analysis also showed that this protocol can provide mutual authentication, data confidentiality and device privacy. Formal analysis using AVISPA is also carried out to verify the protocol. However, since this paper is designed for 4G EPC network, updates is required for implementing the same protocol in 5G Core. Also, privacy-preserving device discovery should also be employed such that the protocol can be used securely.

Recently, T. Balen. et. al [84] proposed a Physical Unclonable Function (PUF) based key generation method. PUF is a unique identification of device or integrated circuit. Using RO PUF (Ring Oscillator Physical Unclonable Function) as a key generation source, an ECC-based asymmetric key pair can be generated uniquely for each DUE. Then, to start secured communication between DUE1 and DUE2, DUE1 and DUE2 will exchange their pre-generated public key to verify each other. After that, two

devices will exchange their session key using DHKE. Subsequently, the session key will be passed as a seed for Salsa20/20 algorithm to generate a pseudo random key, and all D2D messages will be encrypted by performing XOR operations with the Salsa20/20 PRNG output. As a stream ciphering encryption method, Salsa20/20 is believed to be more efficient and secure than other algorithms. The paper has also implemented the logic on SDR evaluation board to show the scheme is feasible. Although the proposed scheme can provide data confidentiality and provide non-repudiation, the degree of revocability is questionable: as the secret key is generated based on an unchangeable value in hardware, if that key is compromised, the device is unlikely to be safe anymore. Also, since this paper has only discussed data communication between two D2D devices, topics about using unique identifier for device identification, secured device discovery or privacy-preserving authentication are still yet to be researched.

In 2019, B. Seok et. al. [38] proposed a lightweight D2D protocol which uses tokens obtained from gNB to perform identity verification. Combining with DHKE for perform key exchange and Authentication Encryption with Associated Data (AEAD) ciphers for symmetric key encryption, the protocol is said to be lightweight but effective to provide data confidentiality and authentication. The protocol has 4 steps. The first step is called “D2D token generation” which all DUEs need to authenticate with cellular network using 5G-AKA, and then the gNB will distribute one unique “D2D token” for every device. This D2D token will be generated using ECDSA algorithm. The second step is device discovery stage which every DUEs will broadcast its own SUCI (Subscription Concealed Identifier) and D2D Token to adjacent devices. After receiving broadcast message from the target DUE, the third step is to send the received SUCI and D2D Token to gNB for verification. If the verification is valid, two devices will enter the final step of negotiation of symmetric key using ECDH. Finally, all subsequent messages will be encrypted using any AEAD cipher algorithms with the common symmetric key. Security analysis has shown that the protocol could provide authentication, data integrity and data confidentiality. It should also be able to mitigate various security attacks such as impersonation and eavesdropping. Moreover, thanks to the anonymity feature of SUCI, device privacy could also be ensured. Nevertheless, since the proposal did not include any formal verification nor logic rules derivation, it is possible that some loopholes of this protocol remain undiscovered.

2.2.2.2.2. Basic Public Key Infrastructure (PKI)

In-coverage compatible schemes

Zhang et. al. [85] proposed a secure D2D data sharing model (SeDS) which is based on cooperation of both gateway (GW), base stations (such as eNB), service provider (SP) and DUEs in cellular network. This model uses different types of cryptographic techniques such as public-key based signature and symmetric encryption (such as Diffie-Hellman Key Exchange) to ensure data confidentiality,

authentication and device privacy. Its non-repudiation characteristics can also help preventing free-riding attacks. For any authenticated DUEs in the same core network, base stations will assign one pseudoidentity for each DUEs, and they will exchange public/private key pairs with DUEs through secured channel. Then, GW will actively search for potential D2D pairs and assist DUEs to pair each other securely without revealing the real identity of another DUE. After that, these DUEs can share the information provided by SP using D2D communication. To secure integrity of data and ensure non-repudiation, all data shared by SP is pre-signed and all messages sent through DUEs will be re-signed with either public-key based signature or Hash-based Message Authentication Code (HMAC). Symmetric key encryption is also used for data sharing among DUEs to provide data confidentiality. Finally, to decrypt and verify the D2D messages, recipient DUE needs to send a key hint request message to base stations. Base stations will decide to either assist data decryption or punish the recipient DUE if it behaves maliciously. This scheme is particularly useful for P2P content sharing during in-coverage connection scenario such as video streaming in a highly dense area. However, since it involves support of many infrastructures in cellular network, availability and dependability of the D2D data transmission are heavily dependent to the performance of these infrastructure. Also, since this proposed model focused only on data sharing which downloads data from SP, it should be further enhanced to fit other D2D use cases.

In 2019, R. Melki et. al. [86] proposed a lightweight authentication and key exchange protocol based on public private key pairs, public key authority (PKA) and the physical characteristic of the shared channel. In the proposal, all DUEs need to generate its own public private key pair using ECC and register itself to PKA. PKAs are built using base stations (gNodeB), they are responsible for issuing certificates to bind public keys to the device identity. The paper also assumed that DUE1 and DUE2 are close to each other, therefore they can scan and receive similar physical characteristics in the shared channel. In case that DUE1 wants to connect to DUE2, DUE1 will first scan the shared channel to generate a nonce indicating the environment. Then, DUE1 will send a connection request message containing the hash of nonce XOR session key, encrypted session key, and a random token R. This connection request message will be encrypted with public key of DUE2 and then send to DUE2. When DUE2 receives the message, DUE2 can decrypt the message using its private key, and decrypt the session key using public key of DUE1. Since two devices are close to each other, DUE2 will also generate the same physical nonce, which can also be used to verify the hashed value. Finally, it will use the session key to encrypt the incremented R value and send back to DUE1 as a confirmation response. The paper mentioned that anonymity, device identity can be prevented, and various attacks such as impersonation, man-in-the-middle (MITM) and tracking can be prevented. However, since the proposed algorithm used public key to encrypt the session key, data forward secrecy may not be guaranteed if the PKAs are under attack and some keys are leaked in the future. Also, since the physical characteristic of the wireless channel is constantly changing, and different physical devices may

measure slightly different results, the generation of physical nonce could also be a challenging problem.

In-coverage and relay-coverage compatible schemes

In 2017, Y. Javed et. al. [33] proposed a D2D relay authentication scheme which use Elliptic Curve Cryptography (ECC), ElGamal (a variant of public key cryptography) and Public Key Infrastructure (PKI) to perform signature verification and encryption. In the proposed system model, different from usual 3GPP 5G or 4G network, they proposed a D2D network which is constituted by many small intelligent D2D relays, and all DUEs need to connect to these small relays to join D2D network. Also, some larger relays called “Multihop Relay Base Station (MRBS)” will connect to these small relays to manage them. In order to establish this large D2D relay network securely, the proposed scheme requires all relays to obtain a valid certificate from CA first. Then, during authentication state, all handshaking messages will be signed and encrypted with two algorithms called “EEoP Scheme for Signature Verification” and “EEoP Scheme for Encryption” which focus on signature verification and message encryption (for the signature verification, SHA3 V3 hash and PKI are jointly employed to generate signature, and for the message encryption, ECC and ElGamal are used). Compared to other authentication protocols, since this scheme uses computationally lightweight ECC to generate asymmetric keys, and it has only two authentication messages (one request and one response), it is particularly suitable for deployment in IoT and mobile devices which requires high speed and low complexity. Security analysis showed that the proposed scheme can prevent various attacks such as brute force attack, quantum attack, man-in-the-middle attack and interleaving attack. This scheme assumed all DUEs should connect to D2D relays or base stations, so it only works for in-coverage or relay-coverage connection scenario. Recently, A. Abro [34] also proposed an enhanced authentication scheme based on EEoP in [33] with improved hashing and encryption process using both ElGamal, ECC and Public Key Infrastructure (FHEEP). The proposed system model assumed all nodes in the D2D network have been registered into CA. Also, all DUEs are connected to some non-transparent relays (NTR), and these relays are connected to a large and computationally powerful MRBS relay to provide connectivity to all connected nodes. For each session, NTR needs to recompute the public/private key after a periodic interval, and SHA384 is used as the hash function during signature. Security analysis showed that since the encryption algorithm is session-based, even though if adversary can hack into one of the sessions, only that session will be compromised and not the remaining sessions. Also, if 192-bit elliptic curve is used in this scheme, it can effectively prevent adversary to solve the discrete logarithm problem (DLP) using “Baby Step, Giant Step” method.

In-coverage, relay-coverage and out-of-coverage compatible schemes

In 2017, Baskaran et. al. [39] presented a Lightweight Incognito Key Exchange (LIKE) which uses a

public key distribution entity called “Security Anchor (SA)” to assist authentication of D2D pairs. In the proposal, it is assumed that all DUEs are under in-coverage scenario, and a new entity SA is located at every base station. This SA will be responsible for generating public-private key pairs for all DUEs, and the ProSe Function and ProSe Application Server within the core network will help find and joint D2D peers within the network. The scheme has 3 main phases. The first phase is system setup phase which initializes all security parameters. The second phase is UE registration at SA which all UEs need to register itself to SA with real identity. They will be assigned a pseudo identity and a public private key pair afterwards. The third phase is the LIKE protocol communication. For this protocol to work, DUE1 firstly need to send a D2D attach request, then the SA in base station will form a list called “Service Candidate Set (SCS)” which contains all connectable D2D nodes. After that, the ProSe Application Function will take this SCS list and prune it based on the physical distance of DUEs to choose the best D2D pair. Next, ProSe Application Server will also receive the request and start assist pairing by sending D2D attach response to DUE1 and D2D initiation request to the chosen DUE2. Finally, DUE1 and DUE2 will find each other, use their assigned public private key pair to authenticate each other and perform session key exchange using DHKE. Security analysis showed that this protocol can prevent free riding attacks (since core network can reject the device request if that device always rejects D2D initiation request), man-in-the-middle attack and replay attack. Also, non-repudiation and device privacy can be guaranteed since the real identity of DUEs are recorded in SA, but pseudo identity is used during data transmission.

Furthermore, N. Dao et. al. [35] suggested the use of prefetched certificate of authority (CA) of D2D devices. In the proposal, all DUEs are certified by a common CA, and each DUE would save their own private key, CA signed own public key (with expiry time) and CA public key locally. Then, in order to perform authentication request, DUE1 will first send his CA signed own public key to DUE2, and DUE2 will verify the signature, and then send back the response message. The response should contain DUE2’s CA signed public key and a DUE1’s public key encrypted random nonce (Nonce A). After that, DUE1 will decrypt the random nonce and send another response which contains the decrypted random nonce, another new nonce (Nonce B) and some other variables. This response will be signed with DUE1’s private key and encrypted with DUE2’s public key. Next, DUE2 will decrypt the message, verify the signature and send back the DUE1’s public key encrypted Nonce B to DUE1. Finally, they can derive a common secret by XOR the Nonce A and Nonce B. To further against brutal force attack, the proposed mechanism has added an expiration timestamp for the secret key, therefore a new secret key is generated before attackers can find the key. This model demonstrated a commonly used authentication technique which promised authentication and data confidentiality. Also, it can be implemented in all connection scenarios given that the D2D certificates are downloaded in advance. However, this paper did not mention about the location and capacity of CA server. Also, without using Diffie-Hellman Key Exchange, there is also a potential risk of data leakage if the private keys of DUEs

are compromised, and the channel is being eavesdropped.

Besides, W. Boubakri et. al. [87] proposed a PKI and Zero Knowledge Proof (ZKP) based access control and authentication scheme specifically designed for 5G networks. In the proposal, hierarchical PKI architecture is built to delegate the issuance of certificates from root certification authority (Root CA) to terminal D2D devices. Specifically, network operators are the root CA which issue and sign “network certificates” to all its access nodes in the network. Then, these access nodes will also become intermediate CA which issue and sign “access certificates” to all connected devices such as CUEs and other deployed facilities. For some connected devices or CUEs can work as a virtual access node, they can also become the next intermediate CA which issue and sign “D2D certificates” for all relay-coverage DUEs. When there are any relay-coverage DUEs attempting to establish connection, ZKP based authentication and registration will start: First the DUE and the virtual access node (VAN) should have a pre-shared key. Then, DUE will issue a M2M communication request to the VAN. Next, VAN will then generate parameters for DUE to generate public-private key pair. After that, DUE will send back the generated public keys with a secret message related to the pre-shared key. Finally, VAN will verify the secret message with ZKP to determine if the DUE is legitimate. If the DUE is legitimate device, VAN will sign the authorization certificate and send back to the DUE. To further provide seamless handover experience, VAN (the D2D relay node) will also actively predict the next node and send a handover request which contains the certificate to target VANs. As a result, DUEs can simply reuse the existing certificate to communicate with the next targeted VANs. This scheme can be used for in-coverage and relay-coverage connection scenarios, and performance analysis showed that the proposed scheme can achieve higher performance than LTE handover. This scheme can also satisfy the security requirements of data confidentiality, authentication and device privacy, because no device related information nor messages are sent through the unencrypted channel.

2.2.2.2.3. Identity-based Cryptography (IBE / IBC)

In-coverage and relay-coverage compatible schemes

O. Hamoud et. al. [88], [89] suggested a certificateless public key cryptography (CL-PKC) based D2D authentication and key management system. This system consists of a new entity called “Regulatory Authority” which generate parameters for key centers, and two Key Generation Centers (KGCs) which generate master keys and public/private key pairs. Assuming all DUEs can connect to the cellular network (including in coverage and partial coverage), this scheme suggested DUEs should register themselves with two KGCs which located in cellular operators and ProSe Service Provider (ProSe SP). During setup stage, it is allowed that two KGCs to have different master keys, but they need to have the same sub-public key (which represents the DUE), so totally two different public/private key pairs will be generated. Then, during registration, these two pairs of keys and a pseudo identifier generated

by DUE will be combined into single public/private key pair. After registration, DUEs can now pair with others under any D2D coverage scenario (including out-of-coverage) because it should have all the parameters already. In order to pairing two DUEs, a triplet containing identifier, public key and a random number will be exchanged between two DUEs. Finally, after some validity check, symmetric key can be calculated for ongoing data confidentiality protection. Since multiple KGCs are used in this scheme, it is believed that the system is resistant to “Malicious but Active KGC Attack” and “Malicious but Passive KGC Attack”. Furthermore, this scheme provided device privacy using pseudo identifier but also remained traceability of devices due to the use of KGCs, so network operators or SP can still track and detect the misbehaving DUEs.

As a variant of IBC, Ciphertext Policy Attribute-Based Encryption (CP-ABE) is an encryption method which encrypts the message according to some access control policies, such that only devices satisfying the same policies and attributes can decrypt the message [90]. In 2015, H. Kwon et. al. [37] proposed an authentication scheme using CP-ABE and out-of-band personal identification number (PIN). Similar to Bluetooth pairing procedures, this scheme used out-of-band shared PIN code to prevent MITM attack. However, while connecting to a large D2D group, the pin may not be easily shared across many user devices. Hence, in this proposal, it presented two different protocols. The first protocol is called “Device to Device Authentication 1 (D2DA1)”. Assuming D2D discovery is finished, DUE1 needs to enter a PIN code first, then it will send the pin code encrypted with the attribute of DUE2 to DUE2. Then, DUE1 will also send a random nonce and a hashed secret to DUE2. Since DUE2 has the attribute of itself, it can decrypt the PIN, and further ask for the PIN from user to double confirm. DUE2 will also check if the hashed secret is correct. Finally, both devices will calculate the common secret key using all information aforementioned. For its second protocol D2DA2, it assumed that DUE is joining a D2D group, and there is one group manager (GM) UE. In order to create group D2D connection, DUE1 will first enter a PIN, the PIN will be encrypted with the attribute of the group, and it will be sent to GM DUE. GM DUE will relay the message to the group, and group members will decrypt the PIN and continue all the authentication procedure like D2DA1. For security analysis, although this scheme can prevent MITM attack, it did not mention about how the DUE1 get the attribute information of DUE2 or the group. If the adversary can also easily get the attribute information through unsecured D2D discovery procedure, the adversary could decrypt the message and get the PIN. Then, it could potentially be more dangerous than traditional Bluetooth pairing procedure which never sends PIN over the air. Also, since the DUE2 should be able to decrypt the PIN, PIN verification becomes a de-facto procedure to detect uninformed connection establishment. It shows that it is possible to further polish the scheme such that it can run safely without user intervention.

Later in 2018, a D2D access control scheme which uses CP-ABE algorithm to encrypt the symmetric keys is suggested in [90]. In order to cater different connection scenarios, two level of trust, namely

“General Trust (GT)” for in-coverage and “Local Trust (LT)” for relay-coverage/out-of-coverage are proposed. The paper assumed that the core network should have both ProSe Function Server and ProSe App Server, which record and evaluate the GT trust levels of all DUEs. For GT only case, DUE needs to register with core network first, then it will be assigned with parameters such as pseudo identity, GT trust levels and attribute keys. After that, a symmetric key will be encrypted by the aforementioned attribute keys and all D2D communication data will be encrypted by that symmetric key. If the targeted DUE also have the same attribute value, it should be able to decrypt the symmetric key and use it to decrypt the D2D communication data. Finally, DUEs can send LT values and feedback vote to core network for future trust value evaluation. For LT only case, attribute keys and local trust values are controlled by DUEs instead of core network, and the remaining steps are similar with GT case. In fact, when the core network is available, GT and LT can also be used together to cater both in-coverage and relay-coverage scenarios. Security analysis showed that the proposed scheme can provide fine-grained access control (i.e., only specific trust level can access the data), data confidentiality and device privacy. However, since this protocol introduced two level of trusts and mixed with CP-ABE cryptography, the overall complexity of this protocol is quite high.

Wang et. al. [91] proposed an anonymous authentication and key agreement protocol (AAKA-D2D) based on the assumption of 3GPP EPS system model, and it is also suitable for 5G cellular network. Using pseudo-identity, identity-based signature (IBS) and Diffie-Hellman key exchange, the proposed protocol works in the in-coverage scenario to provide device anonymity, mutual authentication and session key agreement. There are totally four steps in the proposal, including system initialization, user registration, D2D discovery and D2D session establishment. For the initialization and user registration stages, CN initializes the elliptic curve parameters for computing bilinear pairings and distribute pseudo-identities to all DUEs. For D2D discovery process, DUEs broadcast their own pseudo-identity over-the-air for other discoverers to find them. For a discoverer to initiate connection, it has to send one request including the received DUE’s pseudo-identity to CN. Then, CN distributes a common intermediate key and a session ID to both two DUEs. Subsequently, DHKE messages between two devices are encrypted using the intermediate key, and it is digitally signed using IBS such that the message recipient can verify the signature without having to request public keys from CN. Finally, a session key can be obtained using DHKE messages between two devices. For security analysis, this scheme can provide mutual authentication, perfect forward secrecy, device privacy protection and it should prevent message fabrication attacks. Also, since the pseudo-identity is appended with an expiry timestamp, it provides key revocation to prevent key reuse if the public key is leaked to the adversary. However, with no formal verification proof provided in the proposal, the security of this protocol is yet to be assured. Simulation showed that the performance of the scheme is 20% faster than other IBE-based D2D schemes, which makes it more practical to be deployed in mobile devices than other IBE-based schemes.

2.2.2.3. One-to-many D2D: Group-based Authentication and Key Agreement (Group AKA)

2.2.2.3.1. Basic Public Key Infrastructure (PKI)

In 2015, R. Hsu et al. [92] proposed two PKI based D2D authenticated key exchange protocols, namely “Group Anonymity for D2D communication with CN Assistance (CN-GD2C)” for in-coverage scenario and “Group Anonymity for Network-absent D2D communication (NA-GD2C)” for out-of-coverage scenario in LTE-A network. For CN-GD2C protocol, it is assumed that both DUEs have registered themselves to the core network with their own unique key, and all DUEs have installed the public/private key of home subscriber service (HSS). First, DUE1 will encrypt the D2D request message containing Diffie-Hellman variable and other parameters using public key of HSS and send to DUE2. Then, DUE2 will also encrypt its DH variable and other parameters and concatenate it with the received request message from DUE1 and send to HSS. As HSS and ProSe Server has all group and device information, if DUE1 and DUE2 are found in the same group, after several times of validation, DUE1 and DUE2 will finally authenticate each other and have their shared session key generated by DHKE key exchange for future communication. For NA-GD2C, it is assumed that in a network, there are different D2D groups, and each group will have its own certificate issued by CAs. For every UE, it has stored certificates of different groups in advance, but each UE only belongs to one specific group. During D2D authentication of two DUEs, since cellular network is absent, these certificates will be used instead of the public key of HSS to encrypt request messages. To make sure that eavesdroppers will not be able to know their group, DUE1 will send a list of certificates to DUE2, and DUE2 will respond with another list of certificates and a ciphertext if DUE2 is also within the listed groups. After various of message exchanges, they will be able to confirm if both devices belong to the same group. Security analysis of these two protocols showed that they can provide secure mutual authentication, group anonymity, data confidentiality, device anonymity and traceability. Simulation on Android device also showed that the proposed algorithm is computationally implementable on modern mobile devices. Subsequently, an enhancement based on this protocol is also proposed in [36].

Wang et. al. [93] proposed a constant-round D2D group communication protocol designed for both in-coverage, partial coverage and out-of-coverage scenario. This paper focused on authentication, key exchange and group key update policy. During the protocol setup, one sponsor will be chosen in the D2D network. In case that no cellular network (for example, eNB) is available, this sponsor DUE in the D2D network will be chosen to generate a timestamp and corresponding parameters. After D2D device discovery is completed, the sponsor will assign all DUEs with their own public/private key pairs and pseudo identity for joining the D2D network. These keys and identity will be used to ensure proper authentication and privacy of DUEs. Then, DUEs will enter a two-round process which makes use of Computational Diffie-Hellman (DHKE) to generate session keys and establish sessions with other DUEs. While encountering scenarios that some group members are joining or leaving the D2D group,

the group session key of that D2D network will be updated with the two-round process accordingly, such that forward and backward data secrecy of that group can be protected. Simulation showed that with elliptic curve cryptography (ECC) used, all the keys can be generated with low computational burden of D2D UEs. It is also believed that the protocol is resilient to denial-of-service attacks because it can work without the cellular network. However, since the paper did not focus on D2D device discovery before group formation and data encryption scheme after group formation, privacy and data integrity can only be ensured if proper privacy-preserving discovery process and data encryption scheme are also employed.

Shang et. al. [94] proposed a certificateless authentication and key agreement protocol for D2D group communication in 5G cellular network (CAKA-D2D). For certificateless system, the complete set of public-private key pair consists of two parts, namely the “first part” generated by trusted entity (i.e., the CA), and the “second part” generated by the users registered into the trusted entity. It guarantees that even if the keys in trusted entity (i.e., the “first part”) are leaked, the data is still secure without the private keys from the “second part”. In the proposal, all DUEs are connected to a 3GPP 5G cellular network, and there are totally 5 stages for establishing secure group authentication. The first stage is system setup and user registration. The Access and Mobility Function (AMF) in 5GC generates a master key and publicizes the elliptic curve parameters to all DUEs, and every DUE register itself to AMF to get a pseudo-identity and the first part of public-private key pair. The DUE will also generate the second part of the public-private key pair locally. The second stage is D2D discovery, all DUEs broadcast their pseudo-identity and their complete set of public keys to other DUEs. The fourth stage is group session request, after receiving broadcast from other intended group member DUEs, every participant has to send the aggregated pseudo-identities of other group members to AMF. AMF will return them a new session identity of that group. The fourth stage is key agreement. Each DUE generates a random key hint and send the digitally signed encrypted key hint to their group members. Each DUE also needs to verify the key hint from other group members and aggregate them as one group session key. Finally, in the fifth stage of group activation, each DUE sends the hashed group session key to AMF for verification. If AMF finds all the received hash keys are identical, it will claim the D2D group as activated, and all group members can now start to communicate using that common group session key. For security analysis, the proposed scheme is tested using the AVISPA tool to guarantee that it prevents common active and passive attacks including eavesdropping, man-in-the-middle and desynchronization attacks. To further protect grouped forward and backward secrecy, the proposal also mentioned about the key update procedures of different connection scenarios. Also, the ECDSA signature used in the scheme guarantees prevention of impersonation attacks, and the pseudo-identity makes sure device anonymity. The use of certificateless system has also prevented key escrow problem which allows the single point of failure. Simulation shows that the computational cost of the scheme is lower than most IBE-based schemes, but more analysis is yet to be carried out to understand its performance compared

with other group AKA schemes.

2.2.2.3.2. Identity-based Cryptography (IBE / IBC)

As an enhancement of [75], E. Abd-Elrahman et. al proposed the concept of combining Group Key Management (GKM) mechanism with Identity-based Encryption (IBE) to secure multicast group communication [76]. In the paper, it also assumed that the connection between UE and core network is pre-secured, and multiples private key generators (PKG) are built in the core network. Additionally, each D2D key generator will only distribute keys to its corresponding D2D user group, and all D2D communication is based on a smaller subgroup which only has 2 members to share one symmetric key. After D2D device discovery and discovery message verification phase which is similar to [75], each D2D UEs will be assigned into some groups based on Logical Key Hierarchy (LKH) protocol. For example, in the 4-ary key tree structure, 4 D2D communication pairs (subgroups) will be assigned into one larger group, and 4 groups will be managed by one KGC. After that, each subgroup will generate their own symmetric subgroup key, and KGC will generate another symmetric group key for multicast messages encryption. If there is anyone joining or leaving the multicast group, subgroup and KGC will do re-keying together to achieve perfect forward and backward secrecy. This protocol inherits all the security characteristics of [75], also it does not suffer from the disadvantages of most IBC-based protocols such as key escrow problem and identity disclosure problem. However, since coordination of core network is required in this protocol, it can be applied only at in-coverage connection scenario which may not benefit the public safety applications mentioned in the paper.

M. Wang et. al. [95] proposed two authentication and key agreement protocols (PPAKA-HMAC and PPAKA-IBS) for D2D group communication. These two algorithms work in similar ways, while PPAKA-IBS uses identity-based signature rather than HMAC to provide stronger resistance to internal attacks by malicious users in the group. In the proposal, it is assumed that all DUEs are under the coverage and registered to the cellular network (i.e., in coverage scenario). Both protocols have 6 stages. The first stage is system setup, serving network will initialize all cryptographic parameters for group formation. The second stage is user registration, serving network will distribute pseudo identity to all UEs, and for PPAKA-IBS, an extra public/private key pair will also be distributed. The third stage is D2D discovery, UEs will discover each other with their pseudo identities. The fourth stage is session request, UEs will ask serving network for D2D group formation and the serving network will reply with a list of pseudo identities of the same group. For PPAKA-HMAC, an extra HMAC key will also be sent to all group members for future message signing. The fifth stage is session establishment. For PPAKA-HMAC, there will be a two-round algorithm which the first round uses HMAC to authenticate all users and the second-round computes session keys. For PPAKA-IBS, the two round algorithm is more complicated since it involves verification of many identity-based signatures from different UEs,

however, the procedure is still similar. The final stage is group session activation, session key computed by UEs will be hashed as a seed, and it will be sent to the serving network for verification. If all seeds sent from UEs in the same group are same, the serving network will store the seed into a management table, and the group formation is completed. For scenarios such as session key is expiring, and new user is joining or leaving, new session key will be calculated and updated within the group immediately to make sure group backward and forward secrecy of data. Security analysis also showed that the proposed protocols can fulfil the requirements of data confidentiality, authentication and device privacy.

Improving the two protocols (CN-GD2C and NA-GD2C) mentioned in [92] with Boneh and Franklin Identity-Based Encryption (BF-IBE), k-anonymous secret handshake, DHKE and zero-knowledge proof, R. Hsu et. al. [36] further proposed another group D2D communication and key exchange protocols in 2018, namely “Group Anonymous and Accountable D2D Communication in Mobile Network (GRAAD)”. To recall, BF-IBE scheme has totally four algorithms: **Setup** is to initialize master key and its parameters. **Extract** is to return the private key given the identity ID. **Encrypt** is to encrypt a message with given encryption parameters and identity. **Decrypt** is to decrypt the ciphertext using the private key. For CN-GD2C (cellular network presence) scenario, using BF-IBE, two DUEs firstly choose a random session identity, then DUE1 will send an encrypted D2D request message containing Diffie-Hellman variable and other parameters to DUE2. Then, DUE2 will also send the similar information concatenated with DUE1 request message and send to HSS and ProSe Server. Since HSS (the KGC of BF-IBE) and ProSe Server has all device information and group belonging information, after several times of validation, if DUE1 and DUE2 are in the same group, they will finally be able to authenticate each other. For NA-GD2C (cellular network absence) case, although the procedure is similar to, it is a more complicated mechanism because four new functions including gSelect, uSelect, gSelectVer and uSelectVer are introduced to generate encryption parameters to perform IBE-based encryption locally instead of exchanging a list of certificates. Security analysis showed that the two schemes can provide D2D group anonymity, data confidentiality (with IND-CCA and forward secrecy proven), device privacy while preserving device traceability and revocability.

As an advancement of [77], Sun et. al. [96] proposed an Efficient Anonymity Proximity Device Discovery and Batch Authentication Mechanism (EAP-DDBA) for one-to-many D2D connection establishment in 3GPP 5G network. The proposed scheme combines the privacy preserving discovery procedure in [77] that encrypts the broadcast message with identity-based prefix encryption. Therefore, only authorized D2D users can read the content of the broadcast message and initiate connection request using the ECDH parameters in the encrypted broadcast message. Subsequently, certificateless batch signature (CLBS) is employed such that the broadcasting UE can verify multiple signatures from different connection initiating UEs in a single batch. After that, to respond the connection initiators that all incoming connections are approved in just one single message, the broadcasting UE combines all hashed identities of connection initiating UEs into one single number using Chinese Remainder

Theorem (CRT). Finally, these connection initiating UEs can calculate the remainder using his own parameters and the received number to verify if the broadcasting UE has approved the connection. The scheme has totally three phases, including the system parameter generation phase that generates all encryption parameters, initial access authentication phases that uses 5G-AKA or EAP-AKA' to build secure channels between home network and UEs, and the device discovery and authentication phase that does not rely on home network to establish connections. Security analysis shows that this scheme can provide mutual authentication, device anonymity and perfect forward secrecy. It can also resist many attacks including stolen private key attacks, replay attacks, desynchronization attacks and battery depletion attacks. Performance analysis shows that the scheme performs faster than [36], [83], [95]. However, unlike most other schemes that share the same session key among all group users, this scheme is designed for one-to-many D2D communication. As a result, users within the same group uses their own session key to connect to a single point, and it is using star topology instead of mesh topology.

2.2.2.4. Continuous Authentication

Some schemes focused on authenticating the DUE continuously after successful initial authentication and key agreement are also proposed to secure D2D communications in 5G network:

In 2016, I. Abualhaol et. al. [97] proposed the idea of using legitimacy patterns to continuously authenticate the D2D connections. Legitimacy patterns in this paper refers to one secret pattern agreed on both D2D senders and receivers. During daily data transmission, this pattern will be inserted at the end of one packet, while the location of that packet can be in anywhere. If this pattern is found altered or disappeared in the D2D receiver side, it is likely that the D2D communication is under any physical layer attacks including unauthorized attack or data alternation. Furthermore, traditionally to evaluate one security protocol, researchers need to revisit the whole standard. But with legitimate pattern introduced, a mathematical representation called security score (SeS) of the protocol can be found to evaluate how secure the system is. By definition, SeS is the probability that given the physical attack has happened, the chance that the system can detect this attack successfully. Security analysis showed that although this protocol may not easily against eavesdropping attacks, other attacks such as jamming, restricting access and packet injection can be detected. Simulation also showed that by increasing the percentage of legitimate pattern in the message, the SeS improvement can also increase. However, with longer legitimate patterns, it is expected that attackers may also find and replicate the pattern. Also, the overall throughput of the network could also decrease. As a result, the proposed scheme may not be feasible and still need more improvement.

Recently, Tan et. al. [98] proposed a D2D authentication scheme which uses certificateless authentication for initial connection setup and smartphone sensors data for continuous authentication. In the proposal, it is assumed that all DUEs should be in in-coverage or relay coverage scenario, and all

DUEs are smartphones equipped with both accelerometer and gyroscope. For connection establishment, the scheme has 3 main phases. The first phase is “Offline Registration Phase” which registers the DUE to serving network (SN) with user identification information. After that, a unique license ID will be assigned to the DUE and saved securely for the usage in next phase. The second phase is “Authentication Phase”. During this phase, decisional bilinear Diffie-Hellman (DBDH) will be used to authenticate the DUE. The final stage is “Group Key Distribution State”. If the DUE can be verified by SN, DUE will join a D2D group, and SN will construct a new group key and distribute to all group members in this stage. Unlike most authentication schemes which finished the work, this scheme also introduced continuous authentication. By making use of the activity data collected from accelerometer and gyroscope, DUE will also use machine learning techniques including SVM (Support Vector Machine) and Radial Basis Function (RBF) to classify the behavior of the user and send to SN. Finally, SN will compare the list of activities done by many users to determine if the DUE is trustworthy or it needs reauthentication. Security analysis showed that by using certificateless authentication, key escrow problems in IBE can be solved and it also provided device privacy since license ID is used. However, collecting behavioral data of users and sending it to SN will introduce new privacy concerns, and it is not mentioned in the paper.

2.2.2.5. Content Sharing Protocols

Unlike the traditional way of building a secured communication channel with symmetric key exchange schemes, there are some other proposals that uses different strategies for encrypting messages in D2D systems. These schemes could potentially provide more efficient content sharing and data relaying. It could also provide new features such as message aggregation and manipulations.

A D2D-assist data transmission protocol for M-Health systems is proposed by A. Zhang et. al. [99] and revised by C. Zhou [100]. This protocol consists of two parts: the first part is an algorithm called “Certificateless Generalized Signcryption (CLGSC)” which performs both encryption and signature at the same time. In short, CLGSC is the algorithm which combines key generation centers, public/private key encryption and hash functions to provide both key generation and data protection. The second part is a protocol called “Lightweight and Robust Security-Aware (LRSA)” D2D-assist data transmission protocol which transmit the data through source, relays and destinations securely. In order to start data transmission from the health device to physician, health device needs to undergo 4 phases including system initialization, data formulation, data transmission and data receiving and processing. For system initialization stage, a network manager server (NM) will generate all encryption parameters such as public/private key pairs and secure hash functions. Then all health devices with pseudo identities and all physicians need to register to the NM to get some keys. After that, health device and physician will start initial session key agreement to generate symmetric key using DHKE. The second phase is data

formation that health device would encrypt the content using CLGSC with the physician public key. The third phase is data transmission. Since data may pass through many relays, the signature of the data will be verified in each relay and signed with the relay's public key using CLGSC. After the message is relayed to NM, NM will verify all the signatures and decrypt the relay content. At this moment, the content encrypted with physician's public key will be stored. The final stage is data receiving and processing. When physician receives the data from NM, he will verify and signature and decrypt the content using his own private keys. Security analysis showed that the scheme provided data confidentiality, data integrity, mutual authentication and device privacy. Although the papers assumed that data senders are healthcare devices and data receiver is a server for physician, this scheme can be applied to any D2D client-server model which NM or KGC exists. Therefore, it can be used for connection scenarios including in-coverage, partial coverage and out-of-coverage.

Homomorphic encryption scheme can also be found in some existing D2D content sharing protocols. For instance, L. Wang et. al. [101] proposed a secure content sharing protocol (SCSP) which does the D2D matching using social network and shares common hot topics automatically. In the paper, it is assumed that both DUEs are within the same core network (in-coverage scenario), and both are visiting the same social network server (SNS). The proposed protocol has 3 main phases. The first phase is system initialization which allows trust authority to generate encryption parameters, register SNS with a public/private key pair and register all DUEs. The second phase is user profile matching which every DUEs will submit their public key encrypted interest index table (a table indicating preferred topics or content of a user) to SNS. After that, SNS will calculate their common interest score and credibility score using homomorphic multiplication, decryptions and other operations. The final phase is the secure content sharing phase that DUE can get the common interest topics by requesting base stations with a request message. This message is encrypted, and its integrity is protected by HMAC code. Finally, the base station will assist connection and secure message sharing between two adjacent DUEs using ECDH key exchange. Although the proposed scheme can protect data confidentiality, data integrity, device privacy, non-repudiation and provide mutual authentication, this protocol does not fully utilize the potential of Paillier homomorphic encryption scheme. The proposed scheme only does homomorphic multiplication in SNS, while SNS owned the private key of the encrypted content. This could be further improved if the SNS does not have private key, and all credibility is calculated locally on DUEs.

B. Jin et. al. [102] proposed a D2D data privacy protection mechanism based on choosing a reliable central node in a D2D network and using it to perform homomorphic data aggregation. Specifically, some DUEs in the network want to share private data to one data receiving DUE, but they do not want to reveal their identities to that DUE. Since the proposed scheme assumed that all DUEs are under out-of-coverage scenario, and no central server exists in the D2D network, a security mechanism is needed to aggregate all private data into single data and send back to the data receiving DUE. The proposed

mechanism has two main steps, the first step is to choose one reliable D2D node as a central node, and the second step is to use that central node to receive and send back the aggregated private data to the data receiving DUE. Regarding the central node selection, criteria such as device node density, computation speed, availability, transmission speed and service success rate will be considered and normalized into single performance factor called reliability. Then, reliability of all DUEs will be sorted and the DUE with highest reliability will be selected as central node. After selecting central node, all DUEs will generate one common public/private key pair, and they will encrypt the data with public key and send to the central node without relaying through the data receiving DUE. After that, the central node will aggregate the received values using Paillier homomorphic process, and finally, the aggregated data will be transmitted to data receiving DUE. It can now use the common private key to decrypt the content, without knowing who has contributed which part of the data. Simulation results showed that the proposed mechanism is feasible, however it does not mention about how common public/private key pairs are generated and how to make sure the data can be securely transmitted to central node without relaying through the data receiving DUE. Also, since the central node is chosen voluntarily, it has not enough incentive to assist data aggregation. As a result, more research should be conducted in order to make this mechanism deployable.

2.2.2.6. 5G-D2D Futuristic Applications

Cryptographic techniques used in 5G D2D network could also foster many new and futuristic applications.

2.2.2.6.1. Privacy-Preserving Device Profile Matching

Like privacy-preserving device discovery, device profile matching refers to finding the suitable candidate of common interests. For example, we can find the DUEs which own the same content or have the same properties. Secure Multiparty Computation (MPC), the concept of allowing multiple devices to compute the same task without any central authority and without revealing the content [103], is one of the cryptographic concepts which could be the right solution.

In order to find the users with common interests and create a point-to-point communication, M. Li et. al. proposed a privacy preserving profile matching scheme called FindU in [104], [105]. Unlike common profile matching scheme which uses a central server to collect and match profile information of all users, this paper exploited the concept of MPC, so it does not require any central server to run. This paper proposed two schemes, which are the basic scheme and advanced scheme which involves more computations. Regarding the basic scheme, there are totally 3 phases. The first phase is data share distribution phase which UEs will generate its profile, separate it into many shares, and distribute them to other UEs using Shamir Secret Sharing. The second phase is computation phase which all UEs will do homomorphic computations locally to the received shares. The third phase is called reconstruction

phase. In this phase, all UEs will reveal their shares. For the UE who wants to find its profile intersection to another UE, it has to aggregate a certain number of shares and perform Lagrangian interpolation to get back the result. Although this scheme is believed to be able to find the device with most profile intersection to connect with in an ad-hoc network, it requires all the devices to be connected using ad-hoc network in advance which seems to be a chicken-or-the-egg question. Therefore, the paper mentioned that it should be used with some proper user discovery protocols (e.g., Service Discovery Protocol in Bluetooth) and key establishment procedures (e.g., DHKE) such that a secure communication channel can be provided for initiating the protocol. In fact, in 5G network, since adjacent UEs can communicate through cellular network if the base station signals exist, this could potentially solve this problem and it can be used to search the suitable DUEs in in-coverage scenario without adding new infrastructure in 5GC. As a result, this scheme should be feasible in 5G network, but more research is still required.

Similarly, R. Zhang et. al. [106] also proposed another privacy-preserving profile matching protocol which performs dot-product on homomorphic encrypted profile vector to find common interest. In the paper, it is assumed that both two parties (Alice and Bob) have downloaded the same proximity-based mobile social network (PMSN) app, and they can discover each other through the same network. First, both Alice and Bob need to generate a personal profile factor which showed their characteristics or the degree of interest in many topics. Second, they need to encrypt their vector using Paillier homomorphic encryption scheme. In case that Alice wants to know her interest intersection with Bob, she needs to send this encrypted vector and her public key to Bob and let Bob do the dot-product multiplication for her. To secure both Alice and Bob, this protocol has 3 levels of security. Level 1 refers to the lowest security which Alice can know her similarity to Bob, but Bob knows about her similarity estimation function. Level 2 refers to medium security which Alice knows her similarity to Bob, but Bob does not know about how Alice evaluates it. Level 3 refers to maximum security which Alice can only know if her similarity to Bob has reached a certain threshold, and Bob knows nothing. Implementation and experiment were done in Bluetooth ad-hoc network, and it showed that the proposed protocols are implementable but slow due to the limited network transmission speed, low processing power in mobile device and unoptimized Java Paillier libraries. Also, the proposed protocol did not consider the integrity of the messages since no signature is provided in the protocol. As a result, in order to make it suitable to 5G D2D device matching, this proposal may need further improvement in both security aspect and performance aspect.

2.2.2.6.2. Decryption Outsourcing in D2D Network

Since message encryption and decryption takes time to complete, there are some security application proposals using D2D network to provide fast data security to all DUEs. By delegating the computation loads to other free DUEs or computationally powerful CN, some complicated encryption schemes can now be deployed in mobile network easily with reasonably good performance.

In 2018, Q Li et. al. [107] proposed a D2D-oriented data encryption scheme called “Robust and Scalable Data Access Control Scheme (RSDAC)”. This protocol is using an improved multi-authority CP-ABE (MA-CP-ABE) mechanism which allows each base station to generate intermediate attribute keys and verify the DUE independently. In the paper, it is assumed that there is one core network server (CNS) being the trusted central authority (CA) and several base stations which connects all UEs, verify their attributes and generate intermediate keys (IK) for them. Besides, it assumed that each DUEs would have different roles. For example, data owner (DO) DUE will use a symmetric key to encrypt his data, and then encrypt the symmetric key with some access structure, so that only the users having that attribute can find back the symmetric key. Data user (DU) are the DUEs receiving data from DO and use the data. There are also some DUEs called Outsourced Decryption Service Providers (ODSP) which translate ciphertext into “Partial Decryption Ciphertext (PDC). It helps adjacent DUs to run some of the decryption work such that the overall system performance can be faster. This proposal has totally 4 main phases. The first phase is initialization which creates encryption parameters (e.g., bilinear groups) in CNS and allows all base stations to join and register to CNS. The second phase is data outsourcing stage which allows DO to encrypt the data using “Key Encapsulation Mechanism (KEM)”, the technique which encrypts the content using a symmetric key, and encrypt the key using CP-ABE. The third phase is user key generation phase. In order to allow DU to read the ciphertext, DUs need to register themselves to CNS, and they will be given a public key, private key and user decryption key (UDK), such that they can encrypt any content or decrypt the received ciphertext using these keys. The final phase is decryption and verification phase. In this phase, ODSP will ask DU for his private key so that they can decrypt the ciphertext into PDC and send it back to DU. After DU has received PDC from ODSP, he will further decrypt the PDC into cleartext using his UDK. Data verification can also be done using CNS optionally. As a result, this proposal used MA-CP-ABE to provide a way to outsource the decryption work, allow multi authority by delegating intermediate key generation tasks to base stations and provide data verification. It is useful for 5G environment which tremendous number of DUEs is expected. Security analysis showed that it allows fine-grained access control, while it can also prevent collusion attacks of DUEs and ultra vires of base stations. However, since formal verification of the protocol is absent, it would be questionable if any security concerns would be unveiled during actual deployment of this protocol.

In 2019, H. Wang et. al. [108] proposed an outsourcing threshold decryption (OTD) scheme based on CP-ABE encryption. The paper mentioned that since all ABE-based encryption requires bilinear pairing and other computations which are computationally complex, some computational work should be outsourced to adjacent mobile devices using D2D mobile network. The proposed scheme has three main entity including authority, user 1 (data sender) and user 2 (data receiver). During setup phase, authority would run setup algorithm which creates master key and public parameters. It will also generate and distribute secret keys for data receivers, each secret key has one predefined attribute set. For data sender

to send one message M , he needs to encrypt the message using the received public parameters and one attribute set (as known as access structure) which matches the decryption threshold of data receiver. When data receiver has received the ciphertext, he will firstly generate the outsourced key and retrieve key. Then, he will send the ciphertext with the outsourced key to other adjacent DUEs, such that these DUEs will decrypt part of the content for him. Finally, the cleartext M can be retrieved by receiving all the responses from the DUEs and combine them using the retrieve key. Verification of the received data will also be done during the data combination to make sure adjacent DUEs have computed correct results. With CP-ABE used, security analysis showed that the proposed scheme is strong enough to against chosen plaintext attacks (CPAs). However, the proposal did not provide any incentives for other DUEs to join the cooperative decryption, so it may suffer from various attacks such as free-riding attack which DUEs do not join the network intentionally to save their battery, making this cooperative decryption ineffective.

2.3. Conclusion of the Literature Review

To conclude, in chapter 2.1 of the literature review, we found that the latest 5G authentication protocol 5G-AKA is vulnerable to many cyber-attacks such as linkability attacks, denial-of-service (DoS) attacks, and distributed denial-of-service (DDoS) attacks. Also, 5G-AKA fails to provide perfect forward secrecy and post-compromise secrecy due to the use of the long-term symmetric key and the sequence number. Although many proposals tried to overcome these issues, most of them failed to provide high resistivity to DDoS attacks because of their centralized protocol design. Also, many other proposals are unsuitable for resource-limited 5G devices because they are computationally expensive.

To further extend our discussion of the 5G security, we studied 5G D2D, one of the most promising technologies in the future 5G networks in chapter 2.2. We found that since 5G D2D communication is still yet to be standardized by 3GPP, many researchers proposed novel ways to enhance its security. We broadly classified them into two categories: the physical layer security (PLS) and the cryptographic techniques. Although PLS involves much less computational overhead, it remained impractical because of its infant development. While cryptographic techniques such as authentication protocols are highly feasible, many of them have introduced too much computational overhead for resource-limited 5G devices. Also, many proposals failed to provide adequate security to fight against free-riding and other common attacks.

Chapter 3. A Secure Blockchain-based Authentication and Key Agreement Scheme for 3GPP 5G Networks (5GSBA)

3.1. Introduction

In recent years, the continuous growth of mobile subscribers and connected devices has fostered an active development of the fifth-generation cellular network (5G). 5G networks aim to simultaneously connect as many devices as possible in the same network. To cater to the stringent requirements, the 5G network is constructed by many tiny femtocells to serve many users [4]. In this way, limited spectrum resources can reuse effectively to provide services to more devices simultaneously. Also, having more base stations installed, 5G wireless networks can alleviate traffic congestions in the wireless channels.

While the 5G network is providing many benefits, it also poses more security challenges. Acknowledged by the potential security issues, the Third Generation Partnership Project (3GPP) has standardized authentication and key agreement (AKA) protocol as 5G-AKA in TS 33.501 [23]. 5G-AKA can mutually authenticate and create secure connections between base stations, 5G core networks, and user equipment (UE). It has resolved some pre-existing security issues found in the 4G LTE networks. For example, by encrypting the permanent identity of the UE using the Subscription Concealed Identifier (SUCI), 5G-AKA can prevent the International Mobile Subscriber Identification (IMSI) catching attacks and rogue base stations attacks [41]. However, as some of the authentication procedures in 5G-AKA are inherited from the 4G EPS-AKA [44], some security issues remain unsolved. For example, 5G-AKA suffers from linkability attacks which can track a specific device by using synchronization error messages [42]. Also, it lacks perfect session key forward secrecy that guarantees data confidentiality even if the long term key is stolen in the future [41]. What is more, 5G-AKA is a centralized protocol that relies heavily on two functional entities, namely the Authentication Server Function (AUSF), and the Authentication credential Repository and Processing Function (ARPF) located inside the Unified Data Management (UDM) server. As there will be a tremendous number of devices connecting to the same 5G core network, 5G-AKA would be highly vulnerable to the Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks to paralyze the functional entities in the core network. Consequently, the existing 5G-AKA protocol poses many security threats that affect the robustness and reliability of the 5G networks.

On the other hand, blockchain is a new technology for decentralized applications. Initially proposed by the creator of the Bitcoin cryptocurrency, blockchain is a practical way to construct and manage a trustworthy decentralized ledger database across the network. By storing transactions into data blocks and linking them together using cryptographic hash functions, blockchain ensures all blocks in the chain reaching the consensus effectively. Also, blockchain ensures the data in the database becomes computationally infeasible to mutate. In recent years, researchers have envisioned that blockchain can

be a distributed way to solve many challenging problems in the 5G networks [57]. As the number of 5G network infrastructures and mobile devices is growing exponentially, the benefits of blockchain would become more prominent in the future.

In this chapter, we propose a secure blockchain-based authentication and key agreement scheme for the 5G networks (5GSBA). Our proposed 5GSBA protocol offers the following advantages. First, it provides a distributed and safe way to store all subscribers' information privately. By employing the one-time hash secret, authentication-related entries are only the hashed digests that work similarly as public keys. Hence, even if the database is disclosed in the future, adversaries cannot use it to impersonate any UEs. Moreover, 5GSBA prevents not only the typical network attacks such as eavesdropping, man-in-the-middle attacks, replay attacks, IMSI-catching attacks, but also prevents the DoS and DDoS attacks effectively and provides perfect session key forward/backward secrecy. The main contribution of this paper are as follows:

- 1) We design a novel authentication and key agreement protocol for the 3GPP 5G networks. Our protocol works based on the improvement of the existing system architecture of 5G Core Network. It can be easily adopted to the 3GPP access scenario, which all UEs are connected to the home network via the nearby gNBs.
- 2) Our proposed 5GSBA protocol is secure and efficient. Using blockchains and other state-of-the-art cryptographic functions, 5GSBA can guarantee device unlinkability, mutual authentication and data confidentiality in low computational and energy costs. Most importantly, it not only can prevent all typical network attacks but also deters DoS and DDoS attacks.
- 3) The security of the protocol has been verified with BAN logic and formal verification tool Scyther. The performance evaluation and simulations also demonstrated its resistivity to DoS and DDoS attacks.

3.2. System and Security Model

3.2.1. System Model

Our system model follows the 3GPP 5G system architecture listed in TS 23.501 [5] and the security architecture listed in TS 33.501 [23], with some new features added to the functional entities. Figure 7 shows the 3GPP 5G Core Network (5GC) consisting of many functional entities.

Figure 7 - System and Security Model for 5GSBA

In our system model, a private blockchain network is running across different entities in the 5GC network. First, Next Generation Node B (gNB) is the base station that provides data transmission between UEs and the core network. All gNBs in the 5G network are also the members of the blockchain storing subscribers' information issued by the network operator. Second, Access and Mobility Function (AMF) is the server that verifies and forwards private key related messages from the Authentication Server Function (AUSF) to gNBs. AUSF is the authentication server that manages the access rights and distributes the secret keys for gNBs to access the private blockchain. Network operators can also add new subscriber records through this server. Third, Unified Data Management (UDM) is the server that stores private keys of the blockchain network and all other sensitive information. In the 5GC, one AUSF serves multiple AMFs across the core network, and each AMF serves multiple neighboring gNBs.

For all UEs under the coverage of 3GPP access (i.e., under the signal coverage of gNBs), they can access the 5GC via the nearby gNBs. In the conventional authentication protocols such as 5G-AKA, all UE finds the nearest gNB to start authentication and key agreement with the same AUSF. But in our proposed scheme, authentication work is delegated to the gNBs because they all possess the subscribers' information locally. A UE uses the one-time hash secret stored in their Universal Subscriber Identity Module (USIM) to initiate an authentication request. Then, the gNB hashes the secret and compares it with the records saved in the private blockchain to grant or deny UE's access to the network immediately.

Considering that 5G devices can connect to the 5GC by many different approaches such as non-3GPP access, 3GPP access in the home network, and 3GPP access in the visiting network, designing a universal authentication protocol with optimized performance could be very complicated. In this work, we focus on the most common scenarios to simplify the designed authentication protocol. It is assumed that all UEs use 3GPP access to connect to the gNBs in the home network. These gNBs are connected

to the 5GC. And the connections between gNBs and 5GC are secured by wired connections protected by IPsec tunnels. Thus, if a gNB can mutually authenticate with UEs using the private key of the home network and information stored in the blockchain, it can be regarded that the UEs have joined a legitimate 5GC network.

3.2.2. Security Model

Our security model is also shown in Figure 7. It is assumed that the wireless channels in the 5G environment follow the Dolev-Yao model, which implies that there could be many active and passive attackers in the surroundings. Passive attackers eavesdrop and interpret the messages sent from both UEs and gNBs. Then, they can analyze the intercepted data to figure out the messages sent from both parties. Active attackers not only eavesdrop on the wireless channels but also modify the intercepted messages, replay them, or even fabricate new messages to impersonate legal UEs or gNBs to disrupt the network. All these attackers are labeled as “DY Attackers” in Figure 7. Moreover, although it is assumed that the connections within 5GC are always trustworthy, some accidents or misconfiguration could happen in the future. For example, the databases stored within the functional entities in 5GC could expose to the public in some rare situations. In this scenario, these passive attackers would try to use the exposed permanent keys to decrypt previous communication data. There could also be a lot of compromised botnet UEs in the wireless network. These UEs could launch DoS and DDoS attacks at any scheduled time attempting to flood or paralyze the authentication-related functional entities in the 5GC.

By having all assumptions above, a desired 5G authentication protocol should provide security functionalities including mutual authentication, secure data transmission, session key perfect forward secrecy, and device anonymity. Besides, it should be able to prevent active attacks such as impersonation attacks, linkability attacks [42], replay attacks, man-in-the-middle (MITM) attacks, and DoS attacks. Passive attacks, including eavesdropping and location tracking, should also be deterred.

3.3. Details of the 5GSBA Protocol

3.3.1. Preliminary

To satisfy all security requirements such as mutual authentication, session key perfect forward secrecy, device anonymity, and active and passive attacks prevention, the 5GSBA combines one-time hash function, Elliptic Curve Integrated Encryption Scheme (ECIES), Elliptic Curve Diffie-Hellman (ECDH), and keyed-hash message authentication code (HMAC) at different stages of the protocol. Furthermore, the 5GSBA employs a private blockchain across all gNBs as a distributed subscription repository. Therefore, every gNBs connected to 5GC can approve authentication requests from UEs

without the direct involvement of AUSF and UDM. In this section, we briefly describe the theory of the chosen cryptographic techniques, including the one-time hash, ECDH, HMAC, ECIES, and private blockchains.

3.3.1.1. One Time Hash Function

One-time hash is also known as Lamport's one-time signature. It refers to using the hash digest from a cryptographic hash function (CHF) as the public key and using the original message as the private key. Specifically, CHF needs to fulfill the following requirements to be secure:

1. Deterministic: given a fixed input, it should always generate the same output.
2. Infeasible collision: it is hard to find two different messages generating the same hash digest.
3. One way function: it is hard to find back the original message given a hash digest.
4. Avalanche effect: changing slightly in a message will change the hash value significantly.

For example, the Secure Hash Algorithm 2 (SHA256) is a famous CHF used extensively in Bitcoin [55] and many other applications. It is proposed by the National Security Agency (NSA) in the United States. In this chapter, we assume the CHF used in our 5GSBA is SHA256. Also, the original message (i.e., the private key) will change frequently to reduce the risk of birthday attacks.

3.3.1.2. Elliptic Curve Diffie-Hellman (ECDH)

ECDH can be divided into two parts, the Elliptic Curve Cryptography (ECC) and the Diffie-Hellman Key Exchange (DHKE) algorithm.

For the ECC, it is based on a mathematical equation $y^2 = (x^3 + ax + b) \bmod p$ where $(4a^3 + 27b^2) \bmod p \neq 0$. Basically, for any arbitrary point P , if we list the results of its self-addition (i.e., $P, P + P, P + P + P \dots, n \cdot P, (n + 1) \cdot P$), it is found that there will always be one sequence that repeats for every n number. By having this subgroup order n , we can calculate its order N using Schoof's algorithm, randomly choose a cofactor $h = N/n$, and the point $G = h \cdot P$. Finally, we can publicize the materials (a, b, p, P, G) as the public key and keep the cofactor h as the private key. Therefore, if any adversaries want to find back h from G , the only way would be to add P for x times until $x \cdot P = h \cdot P$ (i.e., brute force attack). However, as the value of h could be very large, and the elliptic curve point addition is computationally expensive, it becomes infeasible to carry out.

To use ECC safely, we must choose the elliptic curve values (a, b, p) carefully, because choosing simple values will make the public key becomes easy to guess. Hence, our protocol only uses the elliptic curve values recommended in the Standards for Efficient Cryptography (SEC) [109]. For example, in 256-bit, we recommend secp256k1 or secp256r1. Although secp256k1 is more widely used because of the popularity of Bitcoin, both are proven to be secure by security researchers.

For the DHKE, it applies the Computational Diffie-Hellman (CDH) problem to generate a common

key. To simply put, CDH means given a value $G = h \cdot P$, it is computationally infeasible for attackers to find back the h by using G . Since ECC satisfies the CDH problem, if we have two users (A and B). User A can send a value $G_A = a \cdot P$, to A, and User B can send a value $G_B = b \cdot P$ to A. In this way, both user A and user B can generate $G_{AB} = a \cdot b \cdot P$ easily, but adversaries cannot: because $G_A \cdot G_B = (a + b) \cdot P$, but not $a \cdot b \cdot P$.

3.3.1.3. Keyed-hash Message Authentication Code (HMAC)

HMAC is a specific type of message authentication code (MAC) that combines a cryptographic hash function with a symmetric key. It accepts two inputs (message, symmetric key), and generates one output (hash digest). HMAC concatenates the message with the block-sized symmetric key and hashes it with a cryptographic hash function such as SHA256. Since HMAC is using the symmetric key, it is relatively simpler than the digital signature that uses asymmetric keys. Consequently, HMAC is an ideal solution for building lightweight network protocols that require message authentication.

3.3.1.4. Elliptic Curve Integrated Encryption Scheme (ECIES)

ECIES combines ECDH and any key derivation function (KDF) to encrypt a message simply and efficiently. In this chapter, we always assume the KDF to be SHA256. Then, the following ECIES steps will run encrypt the message:

1. Alice generates a random private key a , and then a public key $G_A = a \cdot P$.
2. When Bob receives Alice's public key:
 - a. He generates one ephemeral private key b , and then a public key $G_B = b \cdot P$.
 - b. He generates a common secret $G_{AB} = a \cdot b \cdot P$ and the symmetric key $KDF(G_{AB})$.
 - c. He uses the symmetric key to encrypt message m to m' .
 - d. He generates a MAC σ based on m for integrity checking.
 - e. Finally, he sends (G_B, m', σ) back to Alice.
3. When Alice receives the message from Bob:
 - a. She generates the common secret $G_{AB} = a \cdot b \cdot P$ and the symmetric key $KDF(G_{AB})$.
 - b. She decrypts the message m' to m using the symmetric key.
 - c. Finally, she verifies the message integrity by checking if $MAC(m) = \sigma$.

In this way, we can reuse the cryptographic functions used in ECDH and HMAC parts to do asymmetric encryption. This can reduce the complexity of the proposed system with proven security methods.

3.3.1.5. Private Blockchains

A private blockchain is a decentralized database that stores any record privately. There are many

implementations such as Hyperledger Fabrics (which is also reviewed in Section 2.1.2.4). In this chapter, we assume at the beginning of each data block, it contains a key-hashed message authentication code, a digital signature, or simply a cryptographic hash digest of the previous block. Once the new blocks are downloaded, their signatures should be validated immediately. In this way, we guarantee all data stored in the longest blockchain is complete and always immutable.

3.3.2. Details of the Protocol

This section presents the Secure Blockchain-based Authentication and Key Agreement Protocol (5GSBA) in detail, which is a two-step protocol designed for UEs and gNBs to authenticate mutually in the 5G networks. The 5GSBA consists of four phases: system initialization phase, USIM registration phase, gNB broadcast phase, and mutual authentication phase. The notations used in this scheme are listed in Table 2, and Figure 8 shows the workflow of 5GSBA with a sequence diagram.

Table 2 – Notations in 5GSBA

Notation	Description
SUPI	Subscription Permanent Identifier of the UE
ID _{gNB}	Permanent Identifier of gNB
P	Elliptic Curve Generator
Y / Y ₂	One-Time Hash Secret
H(msg)	Cryptographic Hash Function
HMAC(msg, K)	Keyed-hash Message Authentication Code
σ	HMAC code
PK _{core}	Public Key of 5GC
SK _{core}	Private Key of 5GC
TS	Timestamp
K _{hmac}	HMAC Symmetric Key
E _{PKCore} (msg)	Message M is encrypted with Public Key of 5GC

Figure 8 - Sequence Diagram of 5GSBA

Phase 1 – System Initialization: in this phase, let p be the modulus, $E(\mathbb{F}_p)$ be the elliptic curve over the finite field $\mathbb{F}_p$, P be the generator point on $E(\mathbb{F}_p)$ with order n , and $\mathbb{G}$ be the subgroup generated by multiplying the generator point P . Let the cryptographic hash function be $H \subseteq \mathbb{Z}_n^*$. Then, AUSF and gNBs run the following procedures:

- 1) AUSF generates an ECIES private key SK_{core} representing the 5GC by choosing a random number k . Then, the ECIES public key $PK_{core} = k \cdot P$ is stored at UDM.
- 2) Whenever a new gNB attaches to the 5GC, they are mutually authenticated by any existing approach such as IPsec tunnels. Then, AUSF takes a copy of ECIES private key SK_{core} from UDM and install it into the secure enclave of the authenticated gNB.
- 3) Finally, the authenticated gNB keeps a local copy of the latest private blockchain. It should also index the transactions in the blockchain locally for faster access.
- 4) Whenever the blockchain has any updates, the gNB will download and index the new blocks accordingly.

For the private blockchain managed by AUSF, Figure 9 illustrates the structure of one block. For each block, it contains the following elements:

- **Block Header:** It contains the block version for future maintenance and upgrades.
- **Previous Block Hash:** It is the hash of the previous block. It guarantees the immutability of the blockchain.

- **Timestamp:** It is the block creation time for tracking purposes. All transaction timestamps within the block should never be larger than this timestamp.
- **Transactions:** Each transaction is a subscription record for one device. To reduce the storage overhead of the blockchain, each block contains multiple transaction records. This size should be adjustable according to the preference of network operators. By default, we follow the block size of bitcoin as 1MB [110].

Since it is a private blockchain, any efficient algorithms can be used such as Practical Byzantine Fault Tolerance (PBFT) [111] to reach consensus for all gNBs. The details about blockchain consensus implementation are omitted here.

Figure 9 - Structure of A Block

Phase 2 – USIM Registration: during the production of the USIM, the operator should install a one-time hash secret Y and the ECIES public key of the 5GC PK_{core} to the USIM, and the one-time secret hash digest $H(Y)$ to the private blockchain. In this way, when UE shows the collision of the hash function (i.e., the secret Y) to gNB, it can prove to the gNB that it is the legitimate UE. The following shows the procedures:

- 1) The operator generates a one-time hash secret Y and the digest of the one-time hash secret $H(Y)$.
- 2) USIM stores the SUPI, elliptic curve parameters, one-time hash secret Y , and the ECIES public key of 5GC PK_{core} into its non-volatile storage.
- 3) AUSF creates a new transaction including the SUPI, $H(Y)$, timestamp, and a status code. The status code is an operator-configurable number representing the activation status of the SUPI. For

example, “activated” can be 1, “suspended” can be 2, “revoked” can be 3, and so on. The format of one transaction in the blockchain is shown in Figure 10.

SUPI Value	H(Y)	Timestamp	Status
------------	------	-----------	--------

Figure 10 - Structure of A Transaction

Phase 3 – gNB Broadcast: after initialization, gNBs broadcast their own identities ID_{gNB} over the air. Since 5GSBA has made SUCI mandatory during the UE authentication, the identity request procedures in 3GPP TS 33.501 (Section 6.12.4) [23] are no longer needed. Hence, UEs can freely choose when to start authentication and when to prepare of SUCI without having to respond to the identity requests from gNBs.

Phase 4 – Mutual Authentication: whenever a UE is powered up to start authentication with the 5G network, the following two-step protocol will execute:

- 1) **UE → gNB:** UE sends an authentication request to gNB with the following steps:
 - a. Generate a random HMAC key K_{hmac} , a random ECDH private key a , a random ECDH public key $a \cdot P$, and a timestamp TS.
 - b. Generate SUCI by encrypting $\{SUPI, Y, K_{hmac}\}$ with PK_{core} .
 - c. Generate the next one-time hash secret Y_2 .
 - d. Calculate $\sigma_1 = \text{HMAC}(\{ID_{gNB}, SUPI, Y, H(Y_2), TS, a \cdot P\}, K_{hmac})$
 - e. Send the authentication request = $\{SUCI, H(Y_2), TS, a \cdot P, \sigma_1\}$ to gNB.
- 2) **gNB** verifies the received authentication request with the following steps:
 - a. Check the validity of the timestamp TS, and then decrypt the SUCI into $\{SUPI, Y, K_{hmac}\}$ using the private key SK_{core} .
 - b. Verify the HMAC of the message $\sigma_1 = \text{HMAC}(\{ID_{gNB}, SUPI, Y, H(Y_2), TS, a \cdot P\}, K_{hmac})$
 - c. Search the latest transaction record of the SUPI in the private blockchain.
 - d. Compare the hash of the received Y with the $H(Y)$ value stored in the blockchain. If they are equal, continue to send an authentication response. Otherwise, abort the protocol.
- 3) **gNB → UE:** gNB sends an authentication response to UE with the following steps:
 - a. Generate a random ECDH private key b and a ECDH public key $b \cdot P$
 - b. Calculate $\sigma_2 = \text{HMAC}(\{SUPI, TS, b \cdot P\}, K_{hmac})$, where TS is the received timestamp.
 - c. Send the authentication response = $\{TS, b \cdot P, \sigma_2\}$
- 4) **UE** verifies the received authentication response with the following steps:
 - a. Calculate $\sigma' = \text{HMAC}(\{SUPI, TS, b \cdot P\}, K_{hmac})$. If σ_2 equals to σ' , calculate the ECDH session key using formula $a \cdot b \cdot P$.

Similarly, gNB also calculates the ECDH session key using formula $b \cdot a \cdot P$. Since $a \cdot b \cdot P = b \cdot a \cdot P$, a session key is derived and both parties are mutually authenticated. Hence, the UE and the gNB can establish a one-to-one secured communication channel using the derived session key.

3.4. Security Evaluation

Designing a security protocol that looks logically sound is not sufficient as it could involve some hidden bugs hardly be found by human. One of the most famous examples would be the Needham-Schroeder public key protocol [112] designed in 1978. This protocol was widely accepted, until G. Lowe [113] found an attack to it after almost 20 years. Thus, formal verification that proves the logical soundness and exhausts all possible attacks is needed. Since this thesis will design new security protocol, in this section, we will use two formal evaluation tools, the Burrows-Abadi-Needham (BAN) logic, and the Scyther Tool. First, we justify the logical correctness of the 5GSBA using BAN logic. Then, we provide formal verification on security of the 5GSBA using the Scyther Tool. Moreover, we present an extensive qualitative security analysis based on the discussion in [Section 3.2.2](#) to show that the 5GSBA is secure to fight against various malicious attacks.

3.4.1. Logical Proof with Burrows-Abadi-Needham (BAN) Logic

3.4.1.1. Introduction to BAN Logic

Burrows-Abadi-Needham (BAN) logic is a set of rules that verify the logical correctness of a protocol systematically [114]. In order to use BAN logic to verify our protocols, we need to formalize the protocol into idealized form. After that, we can use a set of rules including message meaning rule, nonce verification rule, jurisdiction rule, belief rule, see rule and freshness rule to validate if our protocol can achieve the intended security goals. Here show all the logical rules in BAN Logic.

Notations in BAN Logic

Some frequently used symbols for shortening the logical reasoning in BAN Logic are listed here:

Parties Interactions

- All **parties** or **messages** are represented using capital letters (e.g., P , Q , X).
- For P **believes** Q , it is written as: $P \equiv Q$.
- For P **has said** X , it is written as: $P \mid \sim X$.
- For P **sees** message X , it is written as: $P \triangleleft X$.

- For P **has control over** Q, it is written as: $P \mid\Rightarrow Q$.

Symmetric and Asymmetric Keys

- Q's public key K is written as: $\overset{K}{\mapsto} Q$.
- Q's private key K is written as: K^{-1} .
- Symmetric key K shared among P and Q is written as: $Q \overset{K}{\leftrightarrow} P$.

Message Operations

- To **concatenate** messages X and Y, it is written as: (X, Y) .
- To **encrypt** message X using key Y, it is written as: $\{X\}_K$.
- To **sign** message X using private key Y, it is written as $\{X\}_{K^{-1}}$.
- To **append and encrypt** message X using key Y, it is written as $< X >_Y$.
- To **append a nonce** in message X (such that it is a **recent** message), it is written as: $\#(X)$.

Message Meaning Rule (MMR)

If the message recipient believes the key is secure and trustworthy, it should automatically agree that the message / signature issuer should also be trustworthy.

For Shared Key Model (e.g., AES), if P believes that Q and P shared key K, and P sees a message X encrypted using key K, P believes that Q has said the message X:

$$\frac{P \mid\equiv Q \overset{K}{\leftrightarrow} P, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X}$$

Equation 1 - Message Meaning Rule (Shared Key Model)

For Public Key Model (e.g., DSA) (the arrow is called mapsto), if P believes that Q has public key K, and P sees a message X signed with the private key of K, P believes that Q has said the message X:

$$\frac{P \mid\equiv \overset{K}{\mapsto} Q, P \triangleleft \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X}$$

Equation 2 - Message Meaning Rule (Public Key Model)

For Shared Secret Model (e.g., HMAC), if P believes that Q and P shared secret Y, and P sees a message X encrypted and appended with Y, P believes that Q has said the message X:

$$\frac{P \mid\equiv Q \overset{Y}{\leftrightarrow} P, P \triangleleft < X >_Y}{P \mid\equiv Q \mid\sim X}$$

Equation 3 - Message Meaning Rule (Shared Secret Model)

Nonce Verification Rule (NVR)

If the message recipient P believes that a message X is recent, and P believes that it is issued by Q, P implicitly believes Q also believes the message X in the current session.

$$\frac{P \models \#(X), P \models Q \mid \sim X}{P \models Q \models X}$$

Equation 4 - Nonce Verification Rule

Jurisdiction Rule (JUR)

If the message recipient P believes that the content of a message X is controlled by message issuer Q, and P also believes that Q believes X, then P should implicitly believe X.

$$\frac{P \models Q \mid \Rightarrow X, P \models Q \models X}{P \models X}$$

Equation 5 - Jurisdiction Rule

Belief Rule (BEL)

If the message recipient P believes the whole message, he should also believe parts of it.

$$\frac{P \models X, P \models Y}{P \models (X, Y)}$$

Equation 6 - Belief Rule 1

$$\frac{P \models (X, Y)}{P \models X}$$

Equation 7 - Belief Rule 2

$$\frac{P \models Q \models (X, Y)}{P \models Q \models X}$$

Equation 8 - Belief Rule 3

Belief rule can also combine with see rules:

$$\frac{P \models Q \mid \sim (X, Y)}{P \models Q \mid \sim X}$$

Equation 9 - Belief Rule 4

Sees Rules / Saying Rule

If the message recipient P sees the whole message, it means he also sees parts of the message. Also, if P owns and believes the secret keys, he can decrypt and see the ciphertext.

For generic model:

$$\frac{P \triangleleft (X, Y)}{P \triangleleft X}$$

Equation 10 - Sees Rule (Generic)

For shared key (symmetric) model:

$$\frac{P \models Q \stackrel{K}{\leftrightarrow} P, P \triangleleft \{X\}_K}{P \triangleleft X}$$

Equation 11 - Sees Rule (Symmetric Key)

For public key (asymmetric) model:

$$\frac{P \models \stackrel{K}{\mapsto} P, P \triangleleft \{X\}_K}{P \triangleleft X}$$

Equation 12 - Sees Rule (Public Key)

$$\frac{P \models \stackrel{K}{\mapsto} Q, P \triangleleft \{X\}_{K^{-1}}}{P \triangleleft X}$$

Equation 13 - Sees Rule (Signature)

Freshness Rule

If the message recipient P believes that the message X is fresh for the current session, it implicitly agrees a message containing X should also be fresh.

$$\frac{P \models \#(X)}{P \models \#(X, Y)}$$

Equation 14 - Freshness Rule

By combining all the schemes above, for communication between A and B, if it fulfills the following goals:

$A \models A \stackrel{K_{AB}}{\leftrightarrow} B$ (A believes the symmetric key between A and B)

$B \models A \stackrel{K_{AB}}{\leftrightarrow} B$ (B believes the symmetric key between A and B)

$A \models B \models A \stackrel{K_{AB}}{\leftrightarrow} B$ (A believes that B also believes the symmetric key between A and B)

$B \models A \models A \stackrel{K_{AB}}{\leftrightarrow} B$ (B believes that A also believes the symmetric key between A and B)

Then we can conclude that two parties are “mutually authenticated”.

3.4.1.2. Formalized 5GSBA Protocol

In our idealized protocol, U refers to the UE, C refers to one of the gNBs in the 5G cellular network

(CN). All cleartext and identities in the protocol are omitted as they can be easily forged. For the notations, SUCI can be regarded as a message encrypted by the public key of 5GC (i.e., the K_C). Besides, all message content protected by the HMAC key K_{hmac} can be viewed as a message encrypted by the HMAC key (i.e., K_H). Therefore, the idealized 5GSBA protocol is shown below:

Message 1) $U \rightarrow C: C \triangleleft \left\{ \langle U \xleftrightarrow{K_H} C, Y \rangle_Y \right\}_{K_C}, \{TS, a \cdot G\}_{K_H}$

Message 2) $C \rightarrow U: U \triangleleft \{TS, b \cdot P\}_{K_H}$

3.4.1.3. Logical Assumptions

The following assumptions are made according to the nature of our protocol: First, CN should safely assume that UEs control the HMAC key issued by themselves:

$$\mathbf{A1: } C \models U \mid \Rightarrow U \xleftrightarrow{K_H} C$$

Second, since both CN and UE check the timestamp in the protocol, they should believe the timestamps are recent:

$$\mathbf{A2: } C \models \#(TS) \quad \mathbf{A3: } U \models \#(TS)$$

Third, CN and UE should also believe their locally generated keys are trustworthy to themselves:

$$\begin{aligned} \mathbf{A4: } U \models U \xleftrightarrow{K_H} C & \quad \mathbf{A5: } U \models a & \quad \mathbf{A6: } U \models a \cdot P \\ \mathbf{A7: } C \models b & \quad \mathbf{A8: } C \models b \cdot P \end{aligned}$$

Fourth, UE should believe that the key generated by CN is controlled and trusted by himself. Similarly, CN should also believe the keys generated by UE are controlled by himself:

$$\begin{aligned} \mathbf{A9: } U \models C \mid \Rightarrow b \cdot P & \quad \mathbf{A10: } C \models U \mid \Rightarrow a \cdot P \\ \mathbf{A11: } U \models C \models b & \quad \mathbf{A12: } C \models U \models a \end{aligned}$$

Fifth, CN should believe the secret of the one-time hash sent from UE, by validating with the records in the blockchain. Also, since it is only valid once, it can be viewed as a fresh nonce:

$$\mathbf{A13: } C \models U \xleftrightarrow{Y} C \quad \mathbf{A14: } C \models \#(Y)$$

Finally, since we are using ECDH, we can assume for UE (U), session key $U \xleftrightarrow{K_{UC}} C = a \cdot b \cdot P$ can be calculated with received $b \cdot P$ and the locally generated a . Similarly, for CN (C), session key $U \xleftrightarrow{K_{UC}} C = b \cdot a \cdot P$ can be calculated with received $a \cdot P$ and the locally generated b .

3.4.1.4. Protocol Goals

The goal of the protocol is to achieve mutual authentication of two parties. In other words, we need to create a mutually trusted session key after running the protocol. We can express this idea with four equations:

$$\begin{aligned} U &| \equiv U \xleftrightarrow{K_{UC}} C & C &| \equiv U \xleftrightarrow{K_{UC}} C \\ U &| \equiv C &| \equiv U \xleftrightarrow{K_{UC}} C & C &| \equiv U &| \equiv U \xleftrightarrow{K_{UC}} C \end{aligned}$$

3.4.1.5. Protocol Verification

The detailed verification is listed as follows:

Using see rule with **Message 1**, we have **D1**:

$$\frac{C | \equiv \xrightarrow{K_C} U, C \triangleleft \{ \langle U \xleftrightarrow{K_H} C, Y \rangle_Y \}_{K_C}}{C \triangleleft \langle U \xleftrightarrow{K_H} C, Y \rangle_Y}$$

Using message meaning rule with **A13** and **D1**, we have **D2**:

$$\frac{C | \equiv U \xrightarrow{Y} C, C \triangleleft \langle U \xleftrightarrow{K_H} C, Y \rangle_Y}{C | \equiv U | \sim (U \xleftrightarrow{K_H} C, Y)}$$

Using freshness rule with **A14** and **D1**, we have **D3**:

$$\frac{C | \equiv \#(Y)}{C | \equiv \#(U \xleftrightarrow{K_H} C, Y)}$$

Using nonce verification rule with **D3** and **D2**, we have **D4**:

$$\frac{C | \equiv \#(U \xleftrightarrow{K_H} C, Y), C | \equiv U | \sim \#(K_H, Y)}{C | \equiv U | \equiv (U \xleftrightarrow{K_H} C, Y)} = C | \equiv U | \equiv (U \xleftrightarrow{K_H} C, Y)$$

Using belief rule with **D4**, we have **D5**:

$$\frac{C | \equiv U | \equiv (U \xleftrightarrow{K_H} C, Y)}{C | \equiv U | \equiv U \xleftrightarrow{K_H} C} = C | \equiv U | \equiv U \xleftrightarrow{K_H} C$$

Using jurisdiction rule with **A1** and **D5**, we have **D6**:

$$\frac{C | \equiv U | \Rightarrow U \xleftrightarrow{K_H} C, C | \equiv U | \equiv U \xleftrightarrow{K_H} C}{C | \equiv U \xleftrightarrow{K_H} C} = C | \equiv U \xleftrightarrow{K_H} C$$

As a result, CN believes the received HMAC key, so CN continues to process the Message 1.

Using message meaning rule with **D6** and **Message 1**, we have **D7**:

$$\frac{C \models U \xleftrightarrow{K_H} C, C \triangleleft \{TS, a \cdot P\}_{K_H}}{C \models U \mid \sim (TS, a \cdot P)} = C \models U \mid \sim (TS, a \cdot P)$$

Using freshness rule with **A2** and **D7**, we have **D8**:

$$\frac{C \models \#(TS)}{C \models \#(TS, a \cdot P)} = C \models \#(TS, a \cdot P)$$

Using nonce verification rule with **D7** and **D8**, we have **D9**:

$$\frac{C \models \#(TS, a \cdot P), C \models U \mid \sim (TS, a \cdot P)}{C \models U \models (TS, a \cdot P)} = C \models U \models (TS, a \cdot P)$$

Using belief rule with **D9**, we have **D10**:

$$\frac{C \models U \models (TS, a \cdot P)}{C \models U \models a \cdot P} = C \models U \models a \cdot P$$

Using jurisdiction rule with **A10** and **D10**, we have **D11**:

$$\frac{C \models U \Rightarrow a \cdot P, C \models U \models a \cdot P}{C \models a \cdot P} = C \models a \cdot P$$

As a result, CN believes the received ECDH public key from UE, and the protocol continues with Message 2.

Using message meaning rule with **A4** and **Message 2**, we have **D12**:

$$\frac{U \models U \xleftrightarrow{K_H} P, P \triangleleft \{TS, b \cdot P\}_{K_H}}{U \models C \mid \sim (TS, b \cdot P)} = U \models C \mid \sim (TS, b \cdot P)$$

Using freshness rule with **A3** and **D12**, we have **D13**:

$$\frac{U \models \#(TS)}{U \models \#(TS, b \cdot P)} = U \models \#(TS, b \cdot P)$$

Using nonce verification rule with **D12** and **D13**, we have **D14**:

$$\frac{U \models \#(TS, b \cdot P), U \models C \mid \sim (TS, b \cdot P)}{U \models C \models (TS, b \cdot P)} = U \models C \models (TS, b \cdot P)$$

Using belief rule with **D14**, we have **D15**:

$$\frac{U \models C \models (TS, b \cdot P)}{U \models C \models b \cdot P} = U \models C \models b \cdot P$$

Using jurisdiction rule with **A9** and **D15**, we have **D16**:

$$\frac{U \models C \Rightarrow b \cdot P, U \models C \models b \cdot P}{U \models b \cdot P} = U \models b \cdot P$$

As a result, UE also believes the received ECDH public key from gNB.

For UE, since we know $U \models a$ in **A5** and $U \models b \cdot P$ in **D16**, the common key $U \xleftrightarrow{K_{UC}} C$ can be derived:

$$U \models a \cdot (b \cdot P) = U \models a \cdot b \cdot P = U \models U \xleftrightarrow{K_{UC}} C$$

For CN, since we know $C \models b$ in **A7** and $C \models a \cdot P$ in **D11**, the common key $U \xleftrightarrow{K_{UC}} C$ can be derived:

$$C \models b \cdot (a \cdot P) = C \models b \cdot a \cdot P = C \models U \xleftrightarrow{K_{UC}} C$$

Moreover, to finish the protocol, CN has to believe **Message 1** to continue the protocol and send **Message 2**. Hence, we can say if UE has received **Message 2**, UE can be sure that CN has believed **Message 1**, and therefore $U \models C \models a \cdot P$. Combining this with **A11**, we can conclude that:

$$U \models C \models b \cdot (a \cdot P) = U \models C \models a \cdot b \cdot P = U \models C \models U \xleftrightarrow{K_{UC}} C$$

Similarly, UE has to believe **Message 2** to start the subsequent data transmission. Hence, we can say if CN received subsequent data correctly, CN can be sure that UE has believed **Message 2**, and therefore $C \models U \models b \cdot P$. Combining this with **A12**, we can conclude that:

$$C \models U \models a \cdot (b \cdot P) = C \models U \models a \cdot b \cdot P = C \models U \models U \xleftrightarrow{K_{UC}} C$$

Consequently, all the aforementioned security goals are satisfied, and we can conclude that the 5GSBA is logically verified.

3.4.2. Formal Verification with Scyther Tool

Scyther Tool [115] is an automatic formal verification tool for security protocols. It analyses various of security vulnerabilities of protocols based on Dolev-Yao adversary model and perfect cryptography assumption. By modelling our protocol with Security Protocol Description Language (SPDL), Scyther can read and find potential security issues effectively based on the aforementioned assumptions. Figure 11 shows the screenshot of the latest stable version (v1.1.3) of Scyther Tool.

Figure 11 - Scyther Tool

Scyther Tool provided different security claims: **Secrecy** and **SKR** claims refer to ensuring the variable or session key keep secret throughout the running of protocol. For authentication claims, **Aliveness** refers to ensuring the protocol initiator can successfully complete a run with active responders, which refers to active replies instead of replay messages sent from adversary. **Nisynch** refers to the protocol can complete a run without interruption, but it has no requirement of one-to-one relationship (i.e., non-injective). **Niagree** means the protocol should receive the expected variables after running the protocol without consideration of one-to-one relationship. **Weakagree** refers that all protocol runners communicate with the same expected set of initiators and responders (i.e., injective). **Reachable** refers to the protocol can run to a specified point, which is particularly useful for logic error checking. With appropriate combinations of all these security claims, Scyther can detect many common attacks including MITM attack, message fabrication, replay attack and so on.

Our formal verification result is shown in Figure 13. Our SPDL model in Figure 12 has two roles, the gNB, and the UE. First, the 5GSBA can achieve a mutual key agreement using ECDH. To emulate ECDH key exchange between two parties, we define functions g_1 and g_2 , and then set α to be $g_1(a)$ and β to be $g_1(b)$. We firstly check the secrecy of ECDH private keys using the claims of `Secret a` and `Secret b`. Then, we emulate the ECDH common key using $g_2(\beta, a)$ (i.e., $a \cdot b \cdot G$) and $g_2(\alpha, b)$ (i.e., $b \cdot a \cdot G$) and check its secrecy using **SKR** claims. Second, the 5GSBA uses HMAC to ensure the integrity of all messages. To make sure adversaries cannot get the

HMAC key K_{hmac} (i.e., only gNBs can receive the key from UEs), we check it with `Secret Khmac` claims at both parties. Third, the 5GSBA can guarantee device anonymity by encrypting the SUPI into SUCI. To make sure the adversaries have no way to reveal the SUPI of the UE, we check the secrecy of SUPI using `Secret MSIN` claim. MSIN refers to Mobile Subscriber Identification Number. It is chosen because SUPI contains mobile country code (MCC), mobile network code (MNC), and MSIN. While MCC and MNC are not the unique identifiers to the device, we only mandate MSIN to be secret. Fourth, our protocol assumes that UE needs to use his one-time hash secret Y to prove his identity to gNB. To ensure that the adversaries cannot get the secret Y with any means, we check its secrecy using `Secret Y` claims. Also, the secrecy of $Y2$ is checked to ensure that only UE cannot reveal the next one-time hash secret at any means. Fifth, to further ensure the correctness of our SPDL model, a *Reachable* claim is put at the end of every role. It makes sure that every line of our code has been executed. Finally, by testing all security claims including *Aliveness*, *Niagree*, *Nisynch*, and *Weakagree* in both parties, it shows that our protocol can guarantee injective agreement with active initiator and responders. In other words, we can conclude that there is no attack found in our protocol.

```
// Hash functions
hashfunction h1,h2,g1,g2;
hashfunction h;

// Self Defined Types
usertype String;
usertype Timestamp;

// Global variables
const MCC: String;
const MNC: String;
macro SUPI = (UE,MCC,MNC,MSIN);

/*
 * Hack to simulate  $g^{ab} = g^{ba}$ .
 * '@' prefix of protocol name denotes helper protocol, which is used by
 * Scyther for displaying, and such protocols are ignored in
 * auto-generation of protocol modifiers.
 */
protocol @exponentiation(RA)
{
    role RA
    {
        var alpha,beta, T1,T2: Ticket;

        recv_!1(RA,RA, g2(g1(T1),T2) );
        send_!2(RA,RA, g2(g1(T2),T1) );
    }
}

protocol 5GSBA(gNB,UE)
{
    role gNB {

        var TS: Timestamp;
        var MSIN: Nonce;

        fresh b: Nonce;
        var alpha: Ticket;

        var Khmac: SessionKey;
        var Y: Nonce;
        var HY2: Ticket;

        # Emulate received SKEY from UE before
    }
}
```

```

recv_!1(UE,gNB, {SUPI,h(Y)}k(gNB,UE));

# Actual Protocol
recv_2(UE,gNB,
{SUPI,Y,Khmac}pk(gNB),HY2,TS,alpha,{h(gNB,SUPI,Y,HY2,TS,alpha)}k(Khmac));
send_3(gNB,UE, TS,g1(b),h({SUPI,TS,g1(b)}k(Khmac)));

# Secret Claims
claim_C1(gNB,SKR,g2(alpha,b));
claim_C2(gNB,Secret,b);
claim_C3(gNB,SKR,Khmac);
claim_C4(gNB,Secret,MSIN);
claim_C5(gNB,Secret,Y);

# Attack Claims
claim_C5(gNB,Nisynch);
claim_C6(gNB,Niagree);
claim_C7(gNB,Alive);
claim_C8(gNB,Weakagree);
claim_C9(gNB,Reachable);

}

role UE {

    var TS: Timestamp;
    fresh MSIN: Nonce;

    fresh a: Nonce;
    var beta: Ticket;

    fresh Khmac: SessionKey;
    fresh Y: Nonce;
    fresh Y2: Nonce;

    recv_!T1(UE,UE, TS);

    # Emulate received SKEY from UE before
    send_!1(UE,gNB, {SUPI,h(Y)}k(gNB,UE));

    # Actual Protocol
    send_2(UE,gNB,
{SUPI,Y,Khmac}pk(gNB),h(Y2),TS,g1(a),{h(gNB,SUPI,Y,h(Y2),TS,g1(a))}k(Khmac));
recv_3(gNB,UE, TS,beta,h({SUPI,TS,beta}k(Khmac)));

    # Secret and DHKE Claims
    claim_U1(UE,SKR,g2(beta,a));
    claim_U2(UE,Secret,a);
    claim_U3(UE,SKR,Khmac);
    claim_U4(UE,Secret,MSIN);
    claim_U5(UE,Secret,Y);
    claim_U6(UE,Secret,Y2);

    # Common Security Claims
    claim_U7(UE,Nisynch);
    claim_U8(UE,Niagree);
    claim_U9(UE,Alive);
    claim_U10(UE,Weakagree);
    claim_U11(UE,Reachable);

}

}

```

Figure 12 - SPDL Model of 5GSBA

Scyther results : verify						
Claim				Status	Comments	Patterns
5GSBA	gNB	5GSBA,C1	SKR g2(alpha,b)	ok	No attacks within bounds.	
		5GSBA,C2	Secret b	ok	No attacks within bounds.	
		5GSBA,C3	SKR Khmac	ok	No attacks within bounds.	
		5GSBA,C4	Secret MSIN	ok	No attacks within bounds.	
		5GSBA,C5	Secret Y	ok	No attacks within bounds.	
		5GSBA,gNB1	Nisynch	ok	No attacks within bounds.	
		5GSBA,C6	Niagree	ok	No attacks within bounds.	
		5GSBA,C7	Alive	ok	No attacks within bounds.	
		5GSBA,C8	Weakagree	ok	No attacks within bounds.	
		5GSBA,C9	Reachable	ok	Verified	At least 1 trace pattern.
UE		5GSBA,U1	SKR g2(beta,a)	ok	No attacks within bounds.	
		5GSBA,U2	Secret a	ok	No attacks within bounds.	
		5GSBA,U3	SKR Khmac	ok	No attacks within bounds.	
		5GSBA,U4	Secret MSIN	ok	No attacks within bounds.	
		5GSBA,U5	Secret Y	ok	No attacks within bounds.	
		5GSBA,U6	Secret Y2	ok	No attacks within bounds.	
		5GSBA,U7	Nisynch	ok	No attacks within bounds.	
		5GSBA,U8	Niagree	ok	No attacks within bounds.	
		5GSBA,U9	Alive	ok	No attacks within bounds.	
		5GSBA,U10	Weakagree	ok	No attacks within bounds.	
		Done.	5GSBA,U11	Reachable	ok	Verified

Figure 13 - Formal Verification of 5GSBA with Scyther Tool

3.4.3. Security Analysis

We qualitatively justify that our protocol provides the following security properties:

Mutual Authentication and Key Agreement: By the 5GSBA, a UE uses a one-way hash secret Y stored in the USIM to prove that it is the legitimate party. A gNB verifies it by comparing it to the one-way hash value $H(Y)$ stored in the private blockchain. Since finding the original secret of a hash is a computationally infeasible discrete log problem, the UE can effectively prove to the gNB that it is a legitimate sender of the authentication request message. Besides, the UE encrypts the one-way hash secret Y and a randomly generated HMAC key K_{hmac} using the public key PK_{core} . Since only the legitimate gNBs holding the private key SK_{core} in their secure enclaves can read the one-way hash secret and the HMAC key, the gNB proves to the UE that it is a legitimate sender of the authentication response by issuing the correct HMAC code σ_2 . Thus, both parties can be mutually authenticated to derive a common key using ECDH parameters in the authentication messages.

Secure Data Transmission: The 5GSBA assumes that all subsequent user data will be encrypted with the session key generated in the authentication procedure. For the session key generation, a UE sends an ECDH public key $a \cdot P$ to a gNB, and the gNB replies to another ECDH public key $b \cdot P$ to the UE. Hence, the actual session key $a \cdot b \cdot P$ is never transmitted through the insecure free channel. In fact, the ECDH relies on Computational Diffie-Hellman (CDH) problem, which is difficult to find the a

from $a \cdot P$. Even if an adversary captures all the authentication messages, it is computationally infeasible for him to recover the session key by finding back either a or b to calculate the $a \cdot b \cdot P$. Consequently, the 5GSBA can ensure only legitimate parties (i.e., UE and gNB) can read the user data.

Session Key Perfect Forward / Backward Secrecy: The 5GSBA uses the public key PK_{core} of the 5GC, the randomly generated HMAC key K_{hmac} , and the one-way hash secret Y for authentication purposes only. The session key generation relies on randomly generated ECDH parameters. Hence, even if all the permanent keys are stolen in the future, adversaries still could not recover the content of the session. Also, since the newly generated ECDH parameters for each session are irrelevant to the previous or future sessions, compromising the current session key will not affect the secrecy of the previous or future sessions.

Device Anonymity: Since the SUCI is made mandatory by the 5GSBA, the permanent identifier of requesting UEs are always concealed with ECIES encryption. Hence, eavesdroppers cannot identify a specific device or use the authentication request messages to trace the device.

Protocol Attack Resistance: The 5GSBA outperforms most proposals including the standardized 5G-AKA with the stronger resistibility to many attacks. For example, thanks to the aforementioned security properties, the 5GSBA prevents common attacks such as eavesdropping, location tracking, and man-in-the-middle (MITM) attacks. Moreover, it is highlighted that some critical attacks can be prevented including DoS/DDoS attacks [116], linkability attacks, UE impersonation attacks, rogue base station attacks, and replay attacks.

- **DoS Semantic Attacks Prevention:** By the 5GSBA, semantic attacks exploiting the weaknesses of the protocol can be prevented because an authentication request contains a timestamp and a one-time hash secret Y . Specifically, when the gNB receives an authentication request, it first checks if the timestamp is fresh. Then, it decrypts the SUCI and compares the Y with the records stored in the private blockchain. If the calculated hash value does not match, it will reject the session immediately. By doing so, adversaries cannot hoard extra sessions in the gNB by replaying old authentication messages. Also, since the authentication request involves only the computationally inexpensive ECIES and hash functions, adversaries cannot exhaust the gNBs with dummy requests easily.
- **DDoS Flooding Attacks Prevention:** By decentralizing the authentication work from the AUSF/UDM to all gNBs, the 5GSBA can alleviate the effects of flooding attacks that paralyze the network with authentic requests. In the 5G era, the inter-site distance (ISD) of gNBs is getting smaller, and the number of gNBs deployed is increasing rapidly. Hence, the total computational power of gNBs is growing steadily, making it more difficult for adversaries to flood and paralyze the entire 5G network. Furthermore, since the 5GSBA shifts the authentication work from AUSF/UDM to gNBs, it can also prevent the single-point-of-failure. One gNB failure or one AUSF

failure will not affect the entire 5G network. Thus, DDoS attacks in the 5G authentication can be prevented, and the quality of service (QoS) across the 5G network can be maintained. The performance analysis will show that the 5GSBA accommodates much more incoming requests than conventional centralized schemes.

- **Linkability Attack Prevention:** Unlike conventional symmetric key-based protocols such as 5G-AKA, the 5GSBA generates session keys using ECDH instead of the sequence number. Hence, the 5GSBA does not have the MAC failure or synchronization failure commonly found in symmetric key-based AKA protocols. Adversaries can no longer use these error messages as a loophole to trace a specific device.
- **UE Impersonation Attack Prevention:** By the conventional 5G-AKA protocol, users have to trust the network operator implicitly. Since the network operator owns a copy of users' symmetric keys and sequence numbers, insider attackers in the network can impersonate the UE by abusing these keys. By the 5GSBA, since the one-time hash secret is only stored at the USIM, there is no way for network operators to impersonate the UE using the data stored in the private blockchain. Hence, if the network operators cannot provide the one-time hash secret used in the authentication, users can simply deny all malicious behavior for that session.
- **Rogue Base Station Attack Prevention:** By the 5GSBA, since only the legitimate gNBs can decrypt the SUCI, rogue base stations cannot produce genuine authentication responses by generating the correct HMAC code σ . Thus, it can prevent UEs from establishing connections with rogue base stations.
- **Replay Attack Prevention:** The authentication requests and responses by the 5GSBA are all tagged with a timestamp TS, and the HMAC key K_{hmac} should also work once only. Therefore, by checking the timestamp in both UEs and gNBs, replayed messages can be easily identified and discarded.
- **Battery Depletion Attack Prevention:** By the 5G-AKA, a UE must respond to the identity requests from the serving network by generating a fresh SUCI. If some rogue base stations frequently send the identity requests, the battery of UEs could deplete faster. By the 5GSBA, only UE can take the initiative to generate a fresh SUCI for authentication. Hence, the serving network cannot force the UE to create a fresh SUCI, and it prevents the battery depletion attacks effectively.

3.5. Performance Evaluation

In this section, we analyze performance of the 5GSBA from four different aspects including computational overheads, communication overheads, energy consumption for UE, and effectiveness against DoS attacks. In the comparison, 5G-AKA [44] is selected because it is the 3GPP standardized protocol. SE-AKA [24] is chosen because it supports perfect forward secrecy with the similar security

functions as the 5GSBA. Another blockchain-based 5G authentication protocol, BB-AKA 5G [31] is also included to show that the 5GSBA can achieve a better performance than the existing blockchain-based scheme.

3.5.1. Computational Overheads

3.5.1.1. Theoretical and Experimental Delays

In this subsection, computation overheads are evaluated through simulation experiments. All simulations are conducted on a computer with Intel® Core™ i5-3210M CPU @ 2.5 GHz CPU and 16GB of RAMs. The Charm Crypto v0.5 with Python 3.7.5 is used to create the simulation code. Charm Crypto [117] is a Python wrapper of Pairing-Based Cryptography (PBC) libraries. It allows easy prototyping of cryptographic functions based on elliptic curve cryptosystems. Among all the selected protocols, each of them uses various cryptographic functions, and each of them has a unique key size requirement. To holistically and equally evaluate their performance at the same security level, we follow the NIST recommendation [118] to use 256-bit equivalent key strength throughout all simulations. Hence, for all elliptic-curve cryptography (ECC) based operations, secp256k1 is chosen as the default elliptic curve. For HMAC operations, HMACSHA256 is selected. For each cryptographic function, we run it by 2000 times to measure its average time. Moreover, to evaluate the data access delays incurred in blockchains, we have followed the specifications in Section IV.B to create a blockchain prototype. This blockchain has many 1MB-sized blocks. Every block is stored as one file, and all transactions in the blockchain are indexed using Python Dictionary. To compare the performance difference between blockchain and traditional database, we have also built another centralized database with MariaDB v10.4.14. Finally, Table 3 shows the experimental time of executing different functions.

Then, for the theoretical computational time, the authentication steps of all four schemes are carefully studied. For example, in 5GSBA, during the authentication request, UE generates a new ephemeral ECDH key (which consumes one $T_{ECIES.enc}$), encrypts the identity into SUCI using ECIES (which consumes one $T_{ECDH.gen}$), hashes Y using a cryptographic hash function (which consumes one T_{hash}), and calculates the HMAC σ_1 (which consumes one T_{HMAC}). Then, after receiving the authentication response from gNB, UE verifies the HMAC key σ_2 (which consumes one $T_{HMAC.ver}$). Hence, the total authentication computational overhead for UE would be $T_{ECIES.enc} + T_{ECDH.gen} + T_{hash} + T_{HMAC} + T_{HMAC.ver} = 2.8499$ milliseconds. Furthermore, since both UE and gNB need to derive a session key using ECDH (which consumes one $T_{ECDH.CK}$) in parallel, we should also count the time once. Finally, we evaluate all schemes in the same fashion and derived Table 4. Finally, all the information with the theoretical computational time in Table 4 is combined to find the experimental computational overheads in Figure 14.

It is noteworthy that for all blockchain-based schemes, we have omitted the blockchain indexing time $T_{BC.index}$ in the computational overhead calculation. Blockchain indexing is a slow process for gNBs to categorize the recently downloaded blocks. It can improve the blockchain random access speed to a constant time. Since the indexing can be done in the background parallelly at any time, it would not impose a negative effect on the authentication significantly. Hence, only the average blockchain transaction read time $T_{BC.read}$ and write time $T_{BC.write}$ are included in the calculation.

For the experimental computational delays shown in Figure 14, the 5GSBA is slower than the standardized 5G-AKA by $5.1377 - 3.4683 = 1.6694$ milliseconds. It is because that the 5GSBA employs asymmetric ECDH to replace the less secure symmetric-key-based key derivation function (KDF) in the 5G-AKA. By introducing this tiny computational overhead, the 5GSBA can ensure session key perfect forward secrecy (PFS), backward secrecy, and linkability attack prevention. On the other hand, the computational time of the 5GSBA and the SE-AKA are comparable because both schemes employ ECDH, but the 5GSBA offers an enhanced security function in terms of DoS attack prevention and DDoS alleviation. For the BB-AKA 5G, it consumes much more time than the 5GSBA because the BB-AKA 5G needs two ECDSA signatures and verifications at the gNB and the AMF, while in contrast, the 5GSBA only requires one lightweight one-way hash secret verification in the gNB. Thus, the 5GSBA saves $(10.4107 - 5.1377) / 10.4107 = 50.6\%$ of computational overheads than the BB-AKA 5G. Consequently, although the 5GSBA protocol has unavoidably introduced tiny computational overhead, it is still the most efficient protocol in terms of balancing between security and performance.

Table 3 - Experimental Time for Different Functions

Notation	Description	Time (ms)
$T_{ECDSA.sign}$	ECDSA Sign	0.7286
$T_{ECDSA.ver}$	ECDSA Verify	1.3442
$T_{ECIES.enc}$	ECIES Encryption	2.0572
$T_{ECIES.dec}$	ECIES Decryption	0.7851
T_{HMAC}	KDF / HMAC Calculation	0.0495
$T_{HMAC.ver}$	HMAC Verification	0.0281
$T_{ECDH.gen}$	ECDH Key Generation (1 Exp)	0.6945
$T_{ECDH.CK}$	ECDH Common Key	0.7099
T_{hash}	SHA256 Calculation Time (Hash Time)	0.0206
$T_{sym.enc}$	Symmetric Encryption	0.0925
T_{xor}	XOR Operation	0.0084
$T_{BC.index}$	Blockchain Block Indexing	236.0909
$T_{BC.create}$	Blockchain Block Creation	774.3669
$T_{BC.read}$	Blockchain Transaction Read	0.2914
$T_{BC.write}$	Blockchain Transaction Write	0.0434
$T_{DB.read}$	Centralized Database Read	0.4956

Table 4 - Theoretical Computational Overheads

Protocol	Entity	Authentication Computational Overhead	Execution Time (ms)
5GSBA	UE	$T_{ECIES.enc} + T_{ECDH.gen} + T_{hash} + T_{HMAC} + T_{HMAC.ver}$	2.8499
	CN	$T_{ECIES.dec} + T_{ECDH.gen} + T_{hash} + T_{HMAC} + T_{HMAC.ver} + T_{BC.read} + T_{BC.write}$	1.9126
	Both	$T_{ECDH.CK}$	0.7099
5G-AKA	UE	$T_{ECIES.enc} + 2T_{sym.enc} + T_{xor} + 2T_{HMAC}$	2.3496
	CN	$T_{ECIES.dec} + 2T_{sym.enc} + T_{xor} + 2T_{HMAC} + 2T_{hash} + T_{DB.read}$	1.6143
SE-AKA	UE	$T_{ECIES.enc} + 4T_{HMAC} + T_{HMAC.ver} + T_{ECDH.gen} + T_{ECDH.CK}$	3.6877
	CN	$T_{ECIES.dec} + 2T_{HMAC.ver} + 3T_{HMAC} + T_{ECDH.gen} + T_{ECDH.CK} + T_{DB.read}$	2.8898
BB-AKA 5G	UE	$T_{ECDSA.sign} + 2T_{ECDSA.ver} + T_{ECDH.gen}$	4.1115
	CN	$3T_{ECDSA.sign} + 2T_{ECDSA.ver} + T_{ECDH.gen} + T_{BC.read} + T_{BC.write}$	5.9035
	Both	$T_{ECDH.CK} + T_{hash}$	0.7305

Figure 14 – Experimental Total Computational Delays

3.5.1.2. Average Delays under Unknown Attacks

Although the 5GSBA can resist several malicious attacks shown in the security analysis, it is possible to have some new unknown attacks in the 5G networks interrupting the authentication process. To evaluate the performance under the unknown attacks, it is assumed that for each step of the protocol, it is facing either known or unknown attacks randomly. Hence, there will be a percentage of the chance that the protocol will encounter an unknown attack at every step. If the attack is known, the protocol continues to the next step without interruption. However, if it is an unknown attack, the protocol would be interrupted immediately, so that it must restart from the first step. Based on the assumptions above, we build a simulation model in MATLAB 2020b to measure the average delays for the successful execution of the protocols in Figure 15. It shows that although the 5GSBA runs with slightly higher delays (from 1.67 to 4.17 milliseconds) than those of the 5G-AKA in all scenarios, its performance is still much better than another blockchain-based protocol of the BB-AKA 5G. Considering that the 5GSBA ensures the best security among all protocols, we can still conclude that it can achieve a reasonable balance between high security and reasonable delays.

Figure 15 - Average Delays under Unknown Attacks

3.5.2. Communication Overheads

As bandwidth resources in the 5G network are invaluable, smaller communication overhead is always preferred. We further evaluate the communication overheads in terms of bandwidth consumption and transmission overheads.

For the bandwidth consumption, we derive the total message size of the related protocols based on five different security levels suggested by NIST [118]. Specifically, for 160-bit calculation, elliptic curve secp160r1 + SHA1withECDSA are chosen. For 224 bits, the secp224k1 + SHA224withECDSA are chosen. For 256 bits, the secp256k1 + SHA256withECDSA are chosen. For 384 bits, the secp384r1 + SHA384withECDSA are chosen. For 512 bits, the secp521r1 + SHA512withECDSA are chosen.

Using 5GSBA as an example, to calculate its total message size, we need to measure the size of the authentication request, authentication response, and the average traffic incurred due to blockchain transactions. For 160 bits, we found that its ECIES encrypted string sizes 1136 bits, the hashed Y_2 sizes 160 bits, the ECDH public key sizes 336 bits, the cryptographic hash / HMAC digest sizes 160 bits, and the timestamp fixes at 64 bits. Hence, the size of an authentication request would be $1136+160+64+336+160=1856$ bits, and the size of an authentication response would be $64+336+160=560$ bits. For blockchain transactions, we assume that one successful authentication consumes two blockchain transactions. One is for downloading the old entry, and the other one is for uploading the new entry. For each transaction, it contains one message tag (fixed 8 bits), one 5G-SUPI (fixed 120 bits), one hash (sized 160 bits), and one timestamp (fixed 64 bits). Hence, the total blockchain transaction costs would be $(8+120+160+64) * 2 = 704$ bits. As a result, the total message size of 5GSBA in 160-bit key size would be $1856+560+704=3120$ bits. Using the same way, we derive the message size of all schemes in Figure 16.

Consequently, the results in Figure 16 shows that our 5GSBA outperforms all other schemes by achieving the lowest bandwidth consumption. It is because that the 5GSBA authenticates the incoming UEs locally using the two-step protocol, which can save redundant forwarding messages.

Figure 16 – Total Bandwidth Consumption

For communication delays, there are two components including the propagation delays and the transmission delays. For propagation delay calculation, it consists of both wired and wireless connections. On the wired connections, it refers to the link between the gNB and the nearest 5GC functional entities. It is assumed that optical fiber is deployed for all wired connections, and the 5GC functional entities are geographically distant to the gNB. To simulate a real-life situation, we assume the distance between gNB and the nearest functional entity to be 1 kilometer, and the distance between two functional entities within the 5GC to be less than 0.5 kilometers. On the wireless connection, according to the 5G dense urban service requirement in TS 22.261 [2], it is assumed that the inter-site distance (ISD) between two gNBs is 200 meters, and the requesting UE is located to the edge of the coverage of the gNB. Therefore, the distance between the gNB and UE is about 100 meters. For transmission delay calculation, it also consists of wired and wireless connections. On the wired connections, due to the link of optical fiber, it is assumed that the uplink and downlink speed are both 1Gbit/s. On the wireless connections, similarly by following TS 22.261 [2], it is assumed that the uplink speed for UE is 50Mbit/s and the downlink speed for UE is 300Mbit/s. For blockchain-based schemes, it is assumed that blockchain nodes can synchronize new blocks parallelly during the idle time of gNBs. To better reflect the extra communication overheads introduced by blockchains, we included the average delays of downloading one old transaction and uploading one new transaction into the calculation. Finally, by combining all the assumptions above, the sums of propagation and transmission delays are derived in Figure 17.

Figure 17 - Total Communication Delays

In Figure 17, it shows that the 5GSBA achieves similar total communication delays from 160-bit to the 256-bit key length. Although the total delay of the 5GSBA is slightly higher than that of the 5G-AKA in the 384-bit to 512-bit, the 5GSBA can still consume much less time than that of the BB-AKA 5G. It is because that the 5GSBA has saved some propagation delays by consuming the lengthy authentication requests locally in gNBs. Considering that the 256-bit ECC key is accepted in the NIST recommendation [35], the communication delays of the 5GSBA is very close to that of the 5G-AKA. Overall, the 5GSBA has achieved the right balance between strong security and relatively low communication overheads.

3.5.3. Storage Overheads

3.5.3.1. Storage Cost for UE

To enable UE authentication, three types of information have to be stored in the non-volatile memory of the UE or USIM including key parameters, a permanent identity, and public/private key materials. In this subsection, we compare the total required storage for all protocols in different key lengths. First, for the key parameters, all schemes need to store ECC curve parameters sextuple $T_{EC} = (p, a, b, G, n, h)$. Five elliptic curves, namely the secp160r1, secp224k1, secp256k1, secp384r1, and secp521r1 have been taken for the comparison. The size of these elliptic curve sextuples can be found in [109]. Second, for the permanent identity, the 5GSBA and the 5G-AKA use the SUPI sized 120 bits, the SE-AKA uses permanent ID PID and temporary ID TID sized 128 bits, and the BB-AKA 5G uses transaction number Tx sized 64 bits. The sizes of these identities are independent of the key length. Third, for the private key materials, by the 5GSBA, each UE needs to store an ECIES Public Key PK_{core} and a one-time hash secret Y . By the 5G-AKA, each UE needs to store an ECIES Public Key PK_{core} , a symmetric key K , and a sequence number SQN . By the SE-AKA, each UE stores only the ECIES Public Key PK_{core} and a symmetric key K . By the BB-AKA 5G, each UE stores the public key of the adjacent access point

PK_{AP} , the public key of SN PK_{SN} , and the private key of UE SK_{UE} . The size of these parameters varies according to the different key lengths. Specifically, the sizes of ECIES Public Key PK_{core} are 336 bits in secp160r1, 464 bits in secp224r1, 528 bits in secp256k1, 784 bits in secp384r1, and 1056 bits in secp521r1. Besides, the size of one-time secret Y , the hashed one-time secret $H(Y)$, and the symmetric key K also varies from 160 bits to 512 bits correspondingly. However, since the strength of a symmetric key is independent of the size of the sequence number, the size of sequence number SN remains constant to be 48 bits.

By combining all these parameters, the comparison results are demonstrated in Figure 18. It shows that 5GSBA consumes the least space at UE in all key lengths. On the other hand, the BB-AKA 5G consumes the most storage space because it needs to store the lengthy ECDSA public keys for message verification. Consequently, the proposed 5GSBA is more feasible because it would not introduce excess storage requirements to the memory-limited USIM module.

Figure 18 - Storage Overhead for UE

3.5.3.2. Storage Cost for gNB within a decade

Since the 5GSBA employs a blockchain at gNBs, the storage cost of the private blockchain has to be carefully studied to ensure the cost of infrastructure to be feasible. So, the evaluation on the storage requirement for a blockchain node to store all records within ten years is required.

For the size of parameters in the blockchain, we follow the block and transaction structure of the 5GSBA in Fig. 3 and Fig. 4. The default block size is assumed to be following the block size of Bitcoin as 1MB [110] for convenience. Also, it is assumed that the block header is sized 8 bits, SUPI is sized 120 bits, the one-time secret hash $H(Y)$ is sized 256 bits, the timestamp is sized 64 bits, and the status code is sized 8 bits. Consequently, each transaction costs 1248 bits. For the BB-AKA 5G, it is assumed that the transaction ID Tx is sized 64 bits, the public key PK_{UE} is sized 528 bits, the timestamp is sized 64 bits, and the status code is sized 8 bits. For the number of mobile users, according to the Newzoo Global Mobile Market Report in 2019 [119], the expected number of mobile users at the end of 2020 is 3.2 billion with an annual increase of 7.8%. Finally, by following the estimation method in [60], we estimate

the size of blockchain after i years with Equation 15:

$$\text{Size of blockchain in } i \text{ years} = \left\lceil \frac{N_0 \cdot (1 + R)^i \cdot l_{trans}}{l_{block} - l_{header}} \right\rceil \cdot l_{block}$$

Equation 15 - Size of all records in i years

Where N_0 is the initial number of mobile users, R is the growth rate of mobile users, i is the number of years, l_{trans} is the length of one transaction, l_{block} is the length of one block, l_{header} is the length of one block header. Consequently, the expected storage cost for gNB within the next decade is plotted in Figure 19. It shows that the 5GSBA uses significantly less memory than that of the BB-AKA 5G. Also, by 2030, the storage requirements for the 5GSBA can still be satisfied using a single hard drive easily. Hence, it shows that the 5GSBA is more feasible to implement in the gNBs in a controlled storage cost.

Figure 19 - Size of Blockchain within A Decade

3.5.4. Energy Consumption for UE

Since mobile devices have limited battery life, a security protocol with the strongest security and the least energy consumption is always preferable. To evaluate the energy consumption for a UE, two factors should be considered including the data transmission energy and the energy for execution of the cryptographic functions. For the data transmission energy, the data transfer power model in [120] is adopted to estimate the energy consumption. Thus, the energy cost for uplink transmission of the UE is $E_{ul} = (\alpha_u t_{udr} + \beta) \cdot t_{ul}$ and the energy cost for the downlink is $E_{dl} = (\alpha_d t_{ddr} + \beta) \cdot t_{dl}$, where $\alpha_u = 438.39$ mW/Mbps, $\alpha_d = 51.97$ mW/Mbps, $\beta = 1288.04$ mW, t_{udr} is the uplink throughput, t_{ddr} is the downlink throughput, t_{ul} is the transmission time spent on the uplink, and t_{dl} is the transmission time spent on the downlink. We further assume that the uplink and downlink throughput follow the dense urban 5G service requirement in TS 22.261 [2], so that $t_{udr} = 50$ Mbps and $t_{ddr} = 300$ Mbps.

On the other hand, for the energy consumption for the execution of cryptographic functions, the measure of energy cost approximation in [121] is taken. In [121], all experiments have been conducted using a

battery-powered Compaq iPAQ H3670 PDA with an Intel SA-1110 StrongARM processor clocked at 206MHz, 64MB of RAM, and 16MB of FlashROM. Then, the energy cost of ECDH public key generation $E_{ECDH.gen}$ is 276.7 mJ, the ECDH common key derivation $E_{ECDH.ck}$ is 163.5 mJ, the ECDSA signature generation $E_{ECDSA.sign}$ is 134.2 mJ, and the ECDSA signature validation $E_{ECDSA.ver}$ is 196.23 mJ. For the energy cost of symmetric key-based operations, the symmetric key-based encryption $E_{sym.}$ is $9.92 + 2.29l$ uJ, where l is the number of bytes of the cleartext. The HMAC operation E_{HMAC} is 1.16 mJ, and hash operation E_{hash} is 0.76 mJ. For the ECIES operations, since it is a hybrid encryption mechanism combining both CDH problem and key encapsulation mechanism (KEM), the cost of the encryption is estimated as $E_{ECDH.gen} + E_{ECDH.ck} + E_{sym} = 440.20992 + 0.00229l$ mJ.

Finally, by combining all parameters above with the unknown attack model stated in Section 3.5.1.2, the average UE energy consumption for different protocols is simulated in Figure 20. It shows that all the schemes providing perfect forward and backward secrecy including the 5GSBA, SE-AKA, and BB-AKA 5G consume a similar amount of energy in the situation without any unknown attacks. When the unknown attack probability increases, the 5GSBA is unavoidably using more energy because of more frequent SUCI generation. However, the 5GSBA is still consuming less energy than the BB-AKA 5G. On the other hand, although the 5G-AKA uses the least energy, its power-saving symmetric key-based cryptosystem makes it vulnerable to many other known attacks.

Figure 20 - Energy Consumption for UE

3.5.5. Resistivity to DoS and DDoS Attacks

To show that our 5GSBA can fight against DoS and DDoS attacks, we further design two simulations to compare all related protocols. All simulation results show that the 5GSBA can achieve a high level

of security functionality with reasonable delays.

3.5.5.1. Average Delays under DDoS Flooding Attacks

One of the outstanding features of the 5GSBA is the alleviation of the flooding of DDoS attacks. By the 5GSBA, since all authentication tasks are performed distributed at various gNBs, it avoids congestion and processing at the single authentication entity AUSF. Hence, to illustrate the performance of the protocols under the flooding of many authentication requests, all centralized protocols such as the 5G-AKA and SE-AKA can be modeled as a M/M/1 queuing model, while the blockchain-based protocol of the 5GSBA can be modeled as a M/M/c queuing model (c is the number of gNBs). For simplicity, it is assumed that the arrivals and the locations of UEs follow Poisson distribution, such that all devices within the 1km^2 are located uniformly and can initiate the authentication requests randomly. For the centralized protocols, since the AUSF in the 5GC should be a computationally powerful server, it is assumed that they run on the server with 80% faster than the performance listed in Table 3. For blockchain-based protocol, as the computational resources for each gNB are limited, it is assumed that its computational time is the same as that in Table 3. Furthermore, one AUSF only serves the gNBs within the 1km^2 area, and the ISD of every gNBs is 200m according to the 5G dense urban service requirement in TS 22.261 [2]. Consequently, each server needs to handle the requests from $1\text{km}^2 / \pi 0.1^2 = 31.8 \approx 32$ gNBs (i.e., $c=32$).

Finally, the expected average time of completing the protocol is shown in Figure 21. It shows that although the decentralized 5GSBA takes a bit longer time than the other centralized schemes when the network traffic is low, as the number of requests increases, the 5GSBA outperforms by maintaining an almost constant delay. In other words, the 5GSBA can serve more authentication requests with the same amount of delay than that of the centralized schemes. In future, as the number of gNBs keeps increasing, it is expected that the 5GSBA can perform even better by accommodating more users. Therefore, it shows that the 5GSBA is more robust in terms of accommodating more users, and thus, to alleviate the effects of flooding attacks much better. Moreover, since the 5GSBA can accommodate more requests, it also helps network operators to have more time buffer to detect the DDoS attacks and deploy countermeasures. For example, network operators can use the time to analyze network traffic and block potential DDoS attackers without impacting overall service quality. By doing so, attackers would find it more challenging to perform a successful DDoS attack in the 5G networks.

Figure 21 - Average Delays under Many Requests

3.5.5.2. Average Successful Authentications under DoS Attacks

The 5GSBA can alleviate the effect of DoS attacks by verifying the timestamp and the HMAC code in the authentication request. To prove that the 5GSBA can provide the best DoS attack resistivity among all related schemes, a simulation model is built based on MATLAB 2020b with the following assumptions. There are 1000 mobile devices sending authentication requests to a server at every one millisecond, which can hold at most 1000 sessions simultaneously without any performance degradation. Then, for each authentication request, there could be a percentage of chance that it is a fabricated DoS request. In case that the request is legitimate, the server will complete the protocol operation normally following the protocol specifications. However, if the request is a DoS attack, either of the following results could happen. If the server can identify it as a DoS request, it will terminate the session immediately to accept another new request. If the server could not identify it, the session will hold until it reaches the protocol expiry time at 3 seconds, which is a common default value for RADIUS server timeout. This simulation is run for 60 seconds, the number of successful authentications of different schemes has been recorded in Figure 22.

Figure 22 - Successful Authentications under DoS Attacks

In the figure, it shows that when there is no DoS attack in the network, the 5G-AKA performs the best among all schemes with the highest number of successful authentications. However, when the percentage of DoS attacks increases, the performance of the 5G-AKA drops drastically, while the 5GSBA can always maintain a high number of success authentications. It is because that the 5G-AKA has no way to identify if the authentication requests are fabricated, so the server wastes some sessions for holding until their expiry. On the other hand, the SE-AKA can also provide DoS attack prevention similar to our 5GSBA. But the 5GSBA runs slightly faster, so it has a higher number of successful authentications.

3.7. Conclusion of the 5GSBA

In the 5G era, the explosive growth of mobile penetration makes DoS and DDoS attacks more feasible. To secure 5G networks from these major network attacks, we proposed a Secure Blockchain-based 5G Authentication and Key Agreement (5GSBA) protocol in this chapter. Our 5GSBA protocol combines the one-time hash to a private blockchain and adopts the ECIES, ECDH, and HMAC to ensure authentication security between gNBs and UEs. The security evaluation has proved that 5GSBA is safe under the Dolev-Yao model, and it is resistant to various network attacks. Performance evaluation shows that 5GSBA has improved the security in the 5G network with only a tiny amount of extra overhead. Finally, simulations also show that 5GSBA provides high resistivity to DoS and DDoS attacks.

Chapter 4. A Lightweight Traceable D2D Authentication and Key Agreement Scheme in 5G Cellular Networks (LT-AKA)

4.1. Introduction

With the rapidly growing number of mobile phones and connected smart devices, the development of the fifth-generation cellular network (5G) aims to provide a full-fledged network that allows larger capacity, higher speed, lower latency, and a more efficient network with better resource allocation. Apart from the conventional cellular network topology discussed in the previous chapter, Device-to-Device (D2D) communication is a promising technology in 5G allowing adjacent mobile devices to communicate directly without relaying user data over the base stations. D2D communication provides several benefits over the conventional cellular network, such as improving network efficiency, enabling smart-city applications, and facilitating location-based applications. Most importantly, if all IoT devices in the future are equipped with the 5G D2D function, mobile users can communicate with IoT devices much easier without additional hardware in their phones. For example, smart meters with 5G D2D can relay their readings through 5G mobile phones hassle-free. Consequently, D2D technology could bring more mobile users to a faster, more efficient network with more exciting new applications.

In recent years, the Third Generation Partnership Project (3GPP) has published the feasibility study of D2D communication which covered some basic requirements such as optimized path, device discovery flow and public safety applications [8]. 3GPP has also put the D2D communication under the umbrella of Proximity-based Services (ProSe) [6]. However, since D2D in 5G network has a unique structure of connecting both ad-hoc network and centralized network, it is facing more security challenges than the conventional cellular network. For example, there are eavesdropping, message fabrication, freeriding behavior and so on. In view of these security risks, 3GPP has also specified a general security framework for different ProSe use cases in [122]. This framework defined different levels of device discovery such as open and restricted discovery and specified some basic security requirements for establishing various kinds of D2D connections. Although it provided some directions about what kind of signaling messages should be sent during D2D connection setup, however, it did not provide any recommendations about how mutual authentication, device anonymity, and message non-repudiation are guaranteed during the key agreement process. Consequently, the security risks in 5G D2D networks are still yet to be resolved. To overcome all the security challenges in 5G D2D network, there is a need to design a secure way to protect D2D network traffic: it should mutually authenticate two parties and exchange information without revealing anything to adversaries. It should be able to resist various kinds of attacks, and it should also be computationally lightweight such that it can run on resource-constrained 5G smart devices and battery constrained mobile devices.

In this chapter, we propose a novel lightweight traceable authentication and key agreement protocol (LT-AKA) for one-to-one D2D communication in the 3GPP 5G network. Unlike most existing schemes

that only focus on authentication, our proposed scheme provides both authentication between D2D users and traceability support for network operators. This enables network operators to trace and punish any suspicious D2D users, therefore mitigating possible free-riding attacks in the 5G D2D network. Also, our proposed scheme is computationally lightweight, so it is preferable to all resource-constrained 5G devices such as connected sensors, 5G IoT devices, and low-end mobile phones. The contributions of this chapter can be summarized as follows:

- 1) We propose a novel LT-AKA protocol that provides device discovery, mutual authentication, and key agreement mechanisms for one-to-one D2D communication under the 3GPP 5G network.
- 2) LT-AKA employs key-hashed message authentication code (HMAC) to protect the authenticity of broadcast messages and key exchange messages. Then, it also adopts the Elliptic Curve Diffie-Hellman (ECDH) to exchange the secret session key safely. Finally, it uses the authenticated encryption with associated data (AEAD) cipher to protect the subsequent symmetric key based communication.
- 3) We conduct a formal security analysis using Scyther Tool and Burrows-Abadi-Needham (BAN) logic to show that LT-AKA is secure and resistant to several network attacks, including replay attacks, man-in-the-middle (MITM) attacks, and free-riding attacks.
- 4) We provide a comprehensive performance analysis to show that LT-AKA is the most cost-effective solution compared with other existing schemes.

4.2. System and Security Model

4.2.1. System Model

Our system model is based on the 3GPP 5G architecture listed in TS 23.501 [5]. As shown in Figure 1, 5G Core Network (5GC) is the 3GPP 5G cellular network consisting of various elements. For instance, the Next Generation Node B (gNB) is the signal tower which provides data transmission between UEs and the core network. Access and Mobility Function (AMF) / Security Anchor Function (SEAF) is the server located in either the home or visiting network, which forwards authentication requests to the Authentication Server Function (AUSF) and provides basic access control to the connected UEs. AUSF is the authentication server located in the home network which handles authentication requests and grants permissions to eligible mobile devices. Unified Data Management (UDM) / Authentication credential Repository and Processing Function (ARPF) is the server in the home network which stores private keys and other sensitive information of all Universal Subscriber Identity Module (USIM) issued by the network operator. Session Management Function (SMF) is the server which controls the issuance of all data sessions of UEs. For all UEs under the coverage of 3GPP Access (i.e., under the coverage of gNBs), they can access the 5GC via the nearby gNBs.

In this chapter, we assume that all UEs are connected to the gNB, and the gNBs are connected to the 5GC. Then, we assume that the connection between gNB and 5GC is secured by any means (e.g., cable connection protected with adequate security protocols). Hence, UE and 5GC can mutually authenticate each other with 5G-AKA protocol to create a secure channel between UE, gNB and 5GC. We further assume that both two UEs are under the coverage of gNBs, and both are successfully authenticated to the same 5GC. In our protocol, D2D connect request message, D2D connect request accepted message, D2D discoverer found message, and D2D success indications are transmitted through the secured channel between UEs and gNB, while D2D broadcast messages and D2D key exchange messages will be transmitted through the free channel labelled with “D2D Link”

Figure 23 - System Model of LT-AKA

4.2.2. Security Model

The security model of our proposal is also shown in Figure 23. We assume that since all D2D UEs run 5G-AKA during registration state to mutually authenticate UEs, gNBs and 5GC, these UEs have been connected to reliable gNBs and registered to a trustworthy 5GC. Although 5GC is assumed to be trustworthy, in some rare situation, it is possible that the data stored in 5GC may be exposed to passive attackers due to accidents, misconfigurations or insiders in network operators. Therefore, we must prevent these passive attackers from recovering the past communication data in case of any key leakage in 5GC in the future. On the other hand, for the attackers outside 5GC, due to nature of wireless network, we assume that D2D communication between two UEs is vulnerable to many security attacks. Therefore, we assume that there are many active attackers in the surroundings following Dolev-Yao model. These attackers may eavesdrop, intercept, modify and fabricate any messages over the wireless channel. Finally, for the free riders among all UEs, we assume that all broadcasting UEs shares their data resources, and all discoverer UEs are data users. Although we assume all registered UEs should runs the protocol honestly, there could be some selfish free riders which always connect to other UEs to get data resources, but seldom let other UEs to connect themselves for the sake of power saving. By

having all the assumptions above, our proposed protocol should achieve the following security features:

- (1) **Privacy-Preserving Device Discovery:** UEs can discover any available D2D UEs nearby by scanning the broadcast messages, but adversary cannot use these broadcast messages for identity tracing nor location tracking.
- (2) **Mutual authentication:** the legitimacy and identity of two UEs should be confirmed mutually so that no MTIM attack can be found.
- (3) **Secure data transmission:** after running the protocol successfully, both parties should get a session key. All messages should be encrypted with AEAD cipher using that session key, such that adversary know nothing about the cleartext.
- (4) **Session key perfect forward secrecy:** the session key generated in the protocol should be independent to the keys known to the 5GC. Therefore, even if the keys in 5GC is exposed in the future, adversary cannot recover the old messages.
- (5) **Traceability:** all D2D connection success events should be reported to 5GC, such that 5GC can trace the statistics of all D2D UEs. It can prevent free-riding and facilitate billing if necessary.
- (6) **Protocol attack resistance:** the protocol should be able to prevent both passive and active attacks. Passive attacks are eavesdropping and location tracking. Active attacks are replay attack, MITM attack and free-riding attack.

4.3. Details of the LT-AKA Protocol

4.3.1. Preliminary

To satisfy security requirements such as mutual authentication, session key perfect forward secrecy, device anonymity, traceability support, and active and passive attacks prevention, our scheme combines the hash-based message authentication code (HMAC), elliptic curve Diffie-Hellman (ECDH), and random nonce at different stages to provide secure mutual authentication and key exchange. In this section, we briefly describe the theory of the chosen cryptographic techniques including the random nonce. For the details about HMAC and ECDH, please refer to Section 3.3.1.

4.3.1.1. Random Nonce

Random nonce in this chapter refers to a randomly generated value that uses once only. To ensure a high level of security, this random nonce must satisfy the following requirements:

1. **Unpredictability:** the pseudo-random number generator (PRNG) must have high entropy, such that it generates different results even if two identical devices are running in a very similar environment.
2. **Lengthy:** the range of the random nonce must be long enough, such that collisions or brute force attacks become infeasible.

In this chapter, we always assume the PRNG to be safe, and the random nonce sizes at least 32 bits.

4.3.2. Details of the Protocol

This section presents our proposal, namely the Lightweight Traceable Authentication and Key Agreement Protocol (LT-AKA) for one-to-one D2D communication in detail. Our protocol is a lightweight protocol designed for all mobility devices to discover and establish 5G D2D connections securely. Our scheme has totally 4 stages, including initialization, device discovery, link setup and secure data transmission. Table 5 defines all parameters in our scheme, and Figure 24 shows the detailed working flow of LT-AKA using sequence diagram.

Table 5 - Parameters in LT-AKA

Parameter	Description
GUTI	5G Global Unique Temporary Identifier of UE
K	HMAC Symmetric Secret Key
$\sigma_1, \sigma_2, \sigma_3$	Generated HMAC Code
TK	Unique one-time token (nonce)
TAG	Message tag for message synchronization
G	Key Generator of ECDH
$a \cdot G, b \cdot G$	Public key for ECDH. The a and b are private keys

Figure 24 - Our Proposed LT-AKA Protocol

Stage 1 – Initialization: In this stage, UE and 5GC mutually authenticate each other using 5G-AKA

or 5GSBA discussed in Chapter 3. Each UE is allocated with a 5G-GUTI after establishing secure channel according to the 3GPP TS 33.501 [23]. This 5G-GUTI is a temporary pseudo identity which updates periodically. For every D2D UE, it firstly sends a D2D initialization request to the 5GC to indicate that the UE supports D2D functionality, then 5GC will generate a unique HMAC key K_A and send it back to the UE. We also assume that the elliptic curve parameters required for the subsequent ECC-based operations are preinstalled in the USIM or mobile equipment (ME), following the similar strategies for the SUCI operations stated in 3GPP TS 33 501 [23].

Stage 2 – Device Discovery: In device discovery stage, assuming a UE has some data to share to other UEs, it can make itself discoverable by broadcasting its own 5G-GUTI with HMAC code over insecure free channels. As 5G-GUTI changes periodically over time, the UE will also have to generate new σ_1 locally and update the broadcast messages accordingly. It makes sure the legitimacy and freshness of the broadcast message. The format of $\sigma_1 = \text{HMAC}(\text{GUTI}_{\text{UEA}}, K_A)$, and the broadcast message $M = \{\text{TAG}_{\text{broadcast}}, \text{GUTI}_{\text{UEA}}, \sigma_1\}$.

Stage 3 – Link Setup: After UE_B received the broadcast message from UE_A , they proceed to the link setup stage. The following steps are executed:

- (1) **$\text{UE}_B \rightarrow \text{AMF}$:** UE_B sends the D2D connect request message to AMF via the secured channel. It should include the discoverer message tag, UE_B 's GUTI, UE_A 's GUTI, and the received HMAC code σ_1 from UE_A 's broadcast message.
- (2) Upon receiving the connect request message from UE_B , AMF checks if the GUTIs in the connect request are legitimate. If they are valid, AMF further verifies the HMAC code σ_1 with the stored HMAC key for UE_A . Otherwise, it refuses the request.
- (3) **$\text{AMF} \rightarrow \text{UE}_B$:** AMF allows UE_B to connect to UE_A by sending a connect request accepted message. It contains a message tag, GUTI of UE_A , a random nonce TK which will be sent to UE_A during key exchange, and a common HMAC key K_{AB} shared among UE_A and UE_B . UE_B can now wait for the DHKE request message from UE_A .
- (4) **$\text{AMF} \rightarrow \text{UE}_A$:** AMF notifies UE_A that there is a discoveree UE_B and it wants to connect with UE_A . In this discoveree found message, it includes a message tag, GUTI of UE_B , and a common HMAC key K_{AB} shared among UE_A and UE_B .
- (5) **$\text{UE}_A \rightarrow \text{UE}_B$:** UE_A runs the following step to send DHKE request message:
 - a. Choose a random number $a \in \text{Zn}^*$ and calculate $a \cdot G$.
 - b. Send the DHKE Request Message = $\{\text{GUTI}_{\text{UEB}}, \text{GUTI}_{\text{UEA}}, b \cdot G, \sigma_2\}$ to UE_A , where $\sigma_2 = \text{HMAC}(\{\text{GUTI}_{\text{UEA}}, \text{GUTI}_{\text{UEB}}, a \cdot G\}, K_{AB})$.
- (6) **$\text{UE}_B \rightarrow \text{UE}_A$:** After receiving the DHKE Request Message from UE_A , UE_B runs the following:
 - a. Verify the message with the received K_{AB} with $\text{HMAC}(\{\text{GUTI}_{\text{UEA}}, \text{GUTI}_{\text{UEB}}, a \cdot G\}, K_{AB})$. If the generated HMAC code is different to the received code, it refuses the connection.
 - b. Choose a random number $b \in \text{Zn}^*$ and calculate $b \cdot G$.

- c. Send the DHKE Response Message = $\{GUTI_{UEB}, GUTI_{UEA}, b \cdot G, TK, \sigma_3\}$ to UE_B , where $\sigma_3 = \text{HMAC}(\{GUTI_{UEB}, GUTI_{UEA}, b \cdot G, TK\}, K_{AB})$.
- (7) For UE_A upon receiving the DHKE Response Message from UE_B , it verifies the HMAC code with the received K_{AB} from 5GC with $\text{HMAC}(\{GUTI_{UEB}, GUTI_{UEA}, b \cdot G, TK\}, K_{AB})$. If the calculated HMAC code is different, UE_A refuses the connection.
- (8) $UE_A \rightarrow \text{AMF}$: After UE_A has successfully authenticated the DHKE response message, UE_A sends the D2D success indication to 5GC over the secured channel. The indication message contains a message tag, UE_A 's GUTI, UE_B 's GUTI, and TK. Finally, 5GC can check its validity by comparing the value of received TK with the TK distributed to UE_B before.

Stage 4 – Secure Data Transmission: The final stage of our scheme is to generate the common symmetric session key and start symmetric key based communication. For the session key, it can be derived using ECDH with the formula $K = a \cdot b \cdot G$. For the subsequent symmetric key based communication, a sequence number and timestamp should be concatenated with the message before encryption to prevent possible replay attacks. Then, this concatenated message should be encrypted using common AEAD ciphers such as AES-GCM or any other algorithms, according to the requirement of different use cases.

4.4. Security Evaluation

This chapter evaluates the security of LT-AKA using the security evaluation tools in Section 3.4 to verify that it is logically correct and safe enough to fight against different kinds of attacks.

4.4.1. Logical Proof with Burrows-Abadi-Needham (BAN) Logic

4.4.1.1. Formalized LT-AKA Protocol

Some messages in our protocol are neglected because they do not contribute to the logical properties for validating the protocol. For example, all clear text content, identities and message tags are omitted because these messages can be forged by others, which makes them insufficient for security logic proving. In the idealized form of our protocol, A refers to UE_A , B refers to UE_B and C refers to 5GC. Also, the K_{HAB} refers to the temporal HMAC key for communication between A and B.

Message 3) $C \rightarrow B$: $B \triangleleft \{K_{HAB}, \#(K_{HAB})\}_{K_{BC}}$

Message 4) $C \rightarrow A$: $A \triangleleft \{K_{HAB}, \#(K_{HAB})\}_{K_{AC}}$

Message 5) $A \rightarrow B$: $B \triangleleft \{a \cdot G, \#(a \cdot G), K_{HAB}\}_{K_{HAB}}$

Message 6) $B \rightarrow A$: $A \triangleleft \{b \cdot G, \#(b \cdot G), K_{HAB}\}_{K_{HAB}}$

4.4.1.2. Logical Assumptions

The following assumptions are made according to the nature of our protocol: First, since Message 3 and 4 are sent through the secured channels between UEs and the 5GC, we can safely assume that 5GC issues the HMAC key K_{HAB} recently, and the content is recent:

$$\begin{aligned} \mathbf{A1: } A &| \equiv C \sim (K_{HAB}, \#(K_{HAB})) & \mathbf{A2: } A &| \equiv \#(K_{HAB}, \#(K_{HAB})) \\ \mathbf{A3: } B &| \equiv C \sim (K_{HAB}, \#(K_{HAB})) & \mathbf{A4: } B &| \equiv \#(K_{HAB}, \#(K_{HAB})) \end{aligned}$$

Second, since we fully trust that the keys issued by 5GC is controlled by 5GC, therefore we may assume that:

$$\begin{aligned} \mathbf{A5: } A &| \equiv C \Rightarrow K_{HAB} & \mathbf{A6: } A &| \equiv C \Rightarrow \#(K_{HAB}) \\ \mathbf{A7: } B &| \equiv C \Rightarrow K_{HAB} & \mathbf{A8: } B &| \equiv C \Rightarrow \#(K_{HAB}) \end{aligned}$$

Third, we also assume that the private DHKE key generated by UE itself is recent, and the UE itself trusts this key:

$$\begin{aligned} \mathbf{A9: } A &| \equiv a & \mathbf{A10: } A &| \equiv a \cdot G & \mathbf{A11: } A &| \equiv \#(a \cdot G) \\ \mathbf{A12: } B &| \equiv b & \mathbf{A13: } B &| \equiv b \cdot G & \mathbf{A14: } B &| \equiv \#(b \cdot G) \end{aligned}$$

Fourth, we assume that the received public key from another UE is controlled by that UE:

$$\mathbf{A15: } A | \equiv B \Rightarrow b \cdot G \quad \mathbf{A16: } B | \equiv A \Rightarrow a \cdot G$$

Finally, since we are using DHKE, we assume that for UE_A , session key $A \xleftrightarrow{K_{AB}} B = a \cdot b \cdot G$ can be calculated with received public key $b \cdot G$ and private key a . Similarly, for UE_B , the same session key can also be calculated with received public key $a \cdot G$ and private key b .

4.4.1.3. Protocol Goals

The goal of our protocol is to achieve mutual authentication of two parties. In other words, we need to have a mutually trusted session key for both parties after running the protocol. We can express this idea as follows:

$$\begin{aligned} \mathbf{G1: } A &| \equiv A \xleftrightarrow{K_{AB}} B & \mathbf{G2: } B &| \equiv A \xleftrightarrow{K_{AB}} B \\ \mathbf{G3: } A &| \equiv B | \equiv A \xleftrightarrow{K_{AB}} B & \mathbf{G4: } B &| \equiv A | \equiv A \xleftrightarrow{K_{AB}} B \end{aligned}$$

4.4.1.4. Protocol Verification

Based on the assumptions above, the detailed explanation of the verification is listed as follows:

Using nonce verification rule with **A1** and **A2**, we have **D1**:

$$\frac{A \models \#(K_{HAB}, \#(K_{HAB})) , A \models C \mid \sim (K_{HAB}, \#(K_{HAB}))}{A \models C \models (K_{HAB}, \#(K_{HAB}))}$$

Using nonce verification rule with **A3** and **A4**, we have **D2**:

$$\frac{B \models \#(K_{HAB}, \#(K_{HAB})) , B \models C \mid \sim (K_{HAB}, \#(K_{HAB}))}{B \models C \models (K_{HAB}, \#(K_{HAB}))}$$

Using belief rule with **D1**, we have **D3** and **D4**:

$$\frac{A \models C \models (K_{HAB}, \#(K_{HAB}))}{A \models C \models K_{HAB}} \quad \frac{A \models C \models (K_{HAB}, \#(K_{HAB}))}{A \models C \models \#(K_{HAB})}$$

Using belief rule with **D2**, we have **D5** and **D6**:

$$\frac{B \models C \models (K_{HAB}, \#(K_{HAB}))}{B \models C \models K_{HAB}} \quad \frac{B \models C \models (K_{HAB}, \#(K_{HAB}))}{B \models C \models \#(K_{HAB})}$$

Using jurisdiction rule with **A5** and **D3**, we have **D7**:

$$\frac{A \models C \mid \Rightarrow K_{HAB} , A \models C \models K_{HAB}}{A \models K_{HAB}}$$

Using jurisdiction rule with **A6** and **D4**, we have **D8**:

$$\frac{A \models C \mid \Rightarrow \#(K_{HAB}) , A \models C \models \#(K_{HAB})}{A \models \#(K_{HAB})}$$

Using jurisdiction rule with **A7** and **D5**, we have **D9**:

$$\frac{B \models C \mid \Rightarrow K_{HAB} , B \models C \models K_{HAB}}{B \models K_{HAB}}$$

Using jurisdiction rule with **A8** and **D6**, we have **D10**:

$$\frac{B \models C \mid \Rightarrow \#(K_{HAB}) , B \models C \models \#(K_{HAB})}{B \models \#(K_{HAB})}$$

Using message meaning rule with **D7** and message 5, we have **D11**:

$$\frac{A \models A \xrightarrow{K_{HAB}} B , B \triangleleft \{a \cdot G, \#(a \cdot G), K_{HAB}\}_{K_{HAB}}}{B \models A \mid \sim (a \cdot G, \#(a \cdot G), K_{HAB})}$$

Using freshness rule with **A14** and **D11**, we have **D12**:

$$\frac{B \models \#(K_{HAB})}{B \models \#(a \cdot G, \#(b \cdot G), K_{HAB})}$$

Using message meaning rule with **D9** and message 6, we have **D13**:

$$\frac{B \models A \xleftrightarrow{K_{HAB}} B, A \triangleleft \{b \cdot G, \#(b \cdot G), K_{HAB}\}_{K_{HAB}}}{A \models B \sim (b \cdot G, \#(b \cdot G), K_{HAB})}$$

Using freshness rule with **A11** and **D13**, we have **D14**:

$$\frac{A \models \#(K_{HAB})}{A \models \#(b \cdot G, \#(b \cdot G), K_{HAB})}$$

Using nonce verification rule with **D13** and **D14**, we have **D15**:

$$\frac{A \models \#(b \cdot G, \#(b \cdot G), K_{HAB}), A \models B \sim (b \cdot G, \#(b \cdot G), K_{HAB})}{A \models B \models (b \cdot G, \#(b \cdot G), K_{HAB})}$$

Using nonce verification rule with **D11** and **D12**, we have **D16**:

$$\frac{B \models \#(a \cdot G, \#(a \cdot G), K_{HAB}), B \models A \sim (a \cdot G, \#(a \cdot G), K_{HAB})}{B \models A \models (a \cdot G, \#(a \cdot G), K_{HAB})}$$

Using belief rule with **D15**, we have **D17**:

$$\frac{A \models B \models (b \cdot G, \#(b \cdot G), K_{HAB})}{A \models B \models b \cdot G}$$

Using belief rule with **D16**, we have **D18**:

$$\frac{B \models A \models (a \cdot G, \#(a \cdot G), K_{HAB})}{B \models A \models a \cdot G}$$

Using jurisdiction rule with **A15** and **D17**, we have **D19**:

$$\frac{A \models B \Rightarrow b \cdot G, A \models B \models b \cdot G}{A \models b \cdot G}$$

Using jurisdiction rule with **A16** and **D18**, we have **D20**:

$$\frac{B \models A \Rightarrow a \cdot G, B \models A \models a \cdot G}{B \models a \cdot G}$$

Since we know $A \models a$ and $A \models b \cdot G$, the common session key $A \xleftrightarrow{K_{AB}} B$ can be derived:

$$A \models a \cdot (b \cdot G) = A \models a \cdot b \cdot G = A \models A \xleftrightarrow{K_{AB}} B$$

Similarly, since we know $B \models b$ and $B \models a \cdot G$, the common session key $A \xleftrightarrow{K_{AB}} B$ can also be derived:

$$B \models b \cdot (a \cdot G) = B \models b \cdot a \cdot G = B \models A \xleftrightarrow{K_{AB}} B$$

Finally, since we assume that the protocol continues only if the message 5 from UE_A is verified by UE_B , if the protocol continues (i.e., UE_A can receive response message 6 from UE_B), we can conclude that UE_A believes UE_B believes the content of message 5. In other words, it is $A \models B \models a \cdot G$. Similarly, since we assume that the protocol continues only if the message 6 from UE_B is verified by UE_A

successfully, if the UE_A continues (i.e., CN can receive the success indication message 7 from UE_A), we can also conclude that UE_A believes message 6. In other words, it is $B \models A \models b \cdot G$. Consequently, we can conclude that:

$$A \models B \models A \xleftrightarrow{K_{AB}} B \quad B \models A \models A \xleftrightarrow{K_{AB}} B$$

All required security goals are satisfied, and we can conclude that our protocol is logically verified.

4.4.2. Formal Verification with Scyther Tool

Our formal verification result is displayed in Figure 3. In our SPDL model in Figure 25, we created 3 different roles including CN, UE_A and UE_B . We first emulate the secured channel between UE and CN with the secret session keys “sidA” and “sidB”. Then, we issue the fresh HMAC secret key $hmac_{AB}$, and check the secrecy of this key using the security claim “SKR” in both CN, UE_A and UE_B . It makes sure that the secret key is not leaked to adversaries from the perspective of all parties. For the Diffie-Hellman key exchange, the common secret key is emulated using “ $g2(\text{beta}, a)$ ” (i.e., $a \cdot b \cdot G$) and “ $g2(\text{alpha}, b)$ ” (i.e., $b \cdot a \cdot G$), and their secrecyes are also verified using the security claim “SKR”. To further ensure the correctness of the simulation, **Reachable** is put at the final line of every role, to make sure that every line of the SPDL code is executed. Finally, by testing all security claims including *Secrecy*, *Nisynch*, *Niagree*, *Aliveness*, *Weakagree*, and *Reachable* in all three parties, it shows that our protocol can guarantee a safe injective connection with the active initiators and responders. In other words, we can conclude that there is no attack found in our protocol.

```
// Hash functions
hashfunction h,g1,g2;

// Global variables
usertype String;
const REQUEST: String;
const ACCEPT: String;
const DISCOVEREE: String;
const SUCCESS: String;

// macros for simulating GUTI
macro GUTI-A = (CNTK,CN,UEA);
macro GUTI-B = (CNTK,CN,UEB);

/*
 * Hack to simulate g^ab = g^ba.
 * '@' prefix of protocol name denotes helper protocol, which is used by
 * Scyther for displaying, and such protocols are ignored in
 * auto-generation of protocol modifiers.
 */
protocol @exponentiation(RA)
{
    role RA
    {
        var alpha,beta, T1,T2: Ticket;

        recv_!1(RA,RA, g2(g1(T1),T2) );
        send_!2(RA,RA, g2(g1(T2),T1) );
    }
}

protocol ECCHMAC(CN,UEA,UEB)
```



```

{

role CN {

    # Simulate safe connection to same core network
    var sidA,sidB: SessionKey;
    fresh CNTK: Nonce;

    # For one-to-one D2D connection
    fresh hmacAB: SessionKey;
    fresh TK: Nonce;

    # Simulate safe connection and HMAC distribution
    recv_S1(UEA,CN, {sidA}k(UEA,CN));
    recv_S2(UEB,CN, {sidB}k(UEB,CN));
    send_S3(CN,UEA, {CNTK}k(sidA));
    send_S4(CN,UEB, {CNTK}k(sidB));

    # Receive D2D Requests
    recv_5(UEB,CN, {REQUEST,GUTI-B,GUTI-A}k(sidB));
    not match(GUTI-A,GUTI-B);

    # Respond with tokens and common hmac
    send_6(CN,UEB, {ACCEPT,GUTI-A,TK,hmacAB}k(sidB));
    send_7(CN,UEA, {DISCOVEREE,GUTI-B,hmacAB}k(sidA));

    # Receive success indication from UEA and B
    recv_10(UEA,CN, {SUCCESS,GUTI-A,GUTI-B,TK}k(sidA));

    # Secret Claims
    claim_CN1(CN,SKR,sidA);
    claim_CN2(CN,SKR,sidB);
    claim_CN3(CN,SKR,hmacAB);

    # Attack Claims
    claim_CN4(CN,Nisynch);
    claim_CN5(CN,Niagree);
    claim_CN6(CN,Alive);
    claim_CN7(CN,Weakagree);
    claim_CN8(CN,Reachable);

}

role UEA {

    # Simulate safe connection to same core network
    fresh sidA: SessionKey;
    var CNTK: Nonce;

    # For one-to-one D2D connection
    var hmacAB: SessionKey;
    var TK: Nonce;
    fresh a: Nonce;
    var beta: Ticket;

    # Simulate safe connection to same core network
    send_S1(UEA,CN, {sidA}k(UEA,CN));
    recv_S3(CN,UEA, {CNTK}k(sidA));

    # Receive Indication from CN
    recv_7(CN,UEA, {DISCOVEREE,GUTI-B,hmacAB}k(sidA));

    # DHKE with UEB with TK validation
    send_8(UEA,UEB, GUTI-A,GUTI-B,g1(a),{h(GUTI-A,GUTI-B,g1(a))}k(hmacAB));
    recv_9(UEB,UEA, GUTI-B,GUTI-A,beta,TK,{h(GUTI-B,GUTI-A,beta,TK)}k(hmacAB));

    # Inform CN successful connection
    send_10(UEA,CN, {SUCCESS,GUTI-A,GUTI-B,TK}k(sidA));

    # Secret and DHKE Claims
    claim_A1(UEA,SKR,sidA);
    claim_A2(UEA,SKR,g2(beta,a));
    claim_A3(UEA,Secret,a);
    claim_A4(UEA,SKR,hmacAB);

    # Common Security Claims
    claim_A5(UEA,Nisynch);
    claim_A6(UEA,Niagree);
}

```

```

        claim_A7(UEA,Alive);
        claim_A8(UEA,Weakagree);
        claim_A9(UEA,Reachable);
    }

    role UEB {

        # Simulate safe connection to same core network
        fresh sidB: SessionKey;
        var CNTK: Nonce;

        # For one-to-one D2D connection
        var hmacAB: SessionKey;
        var TK: Nonce;
        fresh b: Nonce;
        var alpha: Ticket;

        # Simulate safe connection to core network
        send_S2(UEB,CN, {sidB}k(UEB,CN));
        recv_S4(CN,UEB, {CNTK}k(sidB));

        # Connect Request to CN (Broadcast Ignored)
        send_5(UEB,CN, {REQUEST,GUTI-B,GUTI-A}k(sidB));
        recv_6(CN,UEB, {ACCEPT,GUTI-A,TK,hmacAB}k(sidB));

        # DHKE with UEA with TK validation
        recv_8(UEA,UEB, GUTI-A,GUTI-B,alpha,{h(GUTI-A,GUTI-B,alpha)}k(hmacAB));
        send_9(UEB,UEA, GUTI-B,GUTI-A,g1(b),TK,{h(GUTI-B,GUTI-
A,g1(b),TK)}k(hmacAB));

        # Secret and DHKE Security Claim
        claim_B1(UEB,SKR,sidB);
        claim_B2(UEB,SKR,g2(alpha,b));
        claim_B3(UEB,Secret,b);
        claim_B4(UEB,SKR,hmacAB);

        # Common Security Claims
        claim_B5(UEB,Nisynch);
        claim_B6(UEB,Niagree);
        claim_B7(UEB,Alive);
        claim_B8(UEB,Weakagree);
        claim_B9(UEB,Reachable);
    }
}

```

Figure 25 - SPDL Model of LT-AKA

Claim	Status	Comments	Patterns
ECCHMAC, CN			
ECCHMAC,CN1 SKR sidA	Ok	No attacks within bounds.	
ECCHMAC,CN2 SKR sidB	Ok	No attacks within bounds.	
ECCHMAC,CN3 SKR hmacAB	Ok	No attacks within bounds.	
ECCHMAC,CN4 Nisynch	Ok	No attacks within bounds.	
ECCHMAC,CN5 Niagree	Ok	No attacks within bounds.	
ECCHMAC,CN6 Alive	Ok	No attacks within bounds.	
ECCHMAC,CN7 Weakagree	Ok	No attacks within bounds.	
ECCHMAC,CN8 Reachable	Ok Verified	At least 1 trace pattern.	1 trace pattern
UEA			
ECCHMAC,A1 SKR sidA	Ok	No attacks within bounds.	
ECCHMAC,A2 SKR g2(beta,a)	Ok	No attacks within bounds.	
ECCHMAC,A3 Secret a	Ok	No attacks within bounds.	
ECCHMAC,A4 SKR hmacAB	Ok	No attacks within bounds.	
ECCHMAC,A5 Nisynch	Ok	No attacks within bounds.	
ECCHMAC,A6 Niagree	Ok	No attacks within bounds.	
ECCHMAC,A7 Alive	Ok	No attacks within bounds.	
ECCHMAC,A8 Weakagree	Ok	No attacks within bounds.	
ECCHMAC,A9 Reachable	Ok Verified	At least 1 trace pattern.	1 trace pattern
UEB			
ECCHMAC,B1 SKR sidB	Ok	No attacks within bounds.	
ECCHMAC,B2 SKR g2(alpha,b)	Ok	No attacks within bounds.	
ECCHMAC,B3 Secret b	Ok	No attacks within bounds.	
ECCHMAC,B4 SKR hmacAB	Ok	No attacks within bounds.	
ECCHMAC,B5 Nisynch	Ok	No attacks within bounds.	
ECCHMAC,B6 Niagree	Ok	No attacks within bounds.	
ECCHMAC,B7 Alive	Ok	No attacks within bounds.	
ECCHMAC,B8 Weakagree	Ok	No attacks within bounds.	
ECCHMAC,B9 Reachable	Ok Verified	At least 1 trace pattern.	1 trace pattern

Done.

Figure 26 - Formal Verification of LT-AKA with Scyther Tool

4.4.3. Security Analysis

We justify that our protocol satisfies the following security requirements:

Mutual authentication: since our protocol requires all D2D UEs to mutually authenticate 5GC using 5G-AKA authentication framework, and all D2D connection requests can only be sent through the secured channel between UE and 5GC, it guarantees that all UEs and 5GC involved during the link setup are legitimate. Also, since the connection request message contains the broadcasted HMAC code of the remote UE, 5GC can also verify the legitimacy of the broadcasting user by calculating the HMAC code using the HMAC key stored in 5GC. Moreover, to allow two D2D UEs to mutually authenticate under the insecure free channel, a randomly generated HMAC key is also distributed from 5GC to the involved UEs. Since this key is never transmitted through insecure channel, and it could only be used once, adversary cannot trick the responder UE by replaying the old messages, nor impersonate the legitimate UEs with another set of DHKE messages.

Secure data transmission: we assume that all user data will be transmitted only when the session is successfully created. Since the session key K is generated by ECDH, the actual session key is never transmitted through insecure free channel. As ECDH relies on Computational Diffie-Hellman (CDH)

problem which finding the $\mathbf{a}$ from $\mathbf{a} \cdot \mathbf{G}$ is difficult, even if the eavesdropper can sniff all the key exchange messages, it is almost impossible for him to know the symmetric key by finding back either $\mathbf{a}$ or $\mathbf{b}$ to get $\mathbf{a} \cdot \mathbf{b} \cdot \mathbf{G}$. Hence, only legitimate users (i.e., UE_A and UE_B) can read the content, even 5GC has no knowledge about the session key.

Session key perfect forward secrecy: our protocol guarantees that even in rare situation that all HMAC keys in AMF are stolen in future, adversary cannot recover the previous D2D message content. It is because the HMAC keys stored in AMF are only for message authenticity verification. All session keys are generated locally in D2D UEs during the creation of DHKE request message and DHKE response message.

Device anonymity: our protocol inherits the device anonymity characteristics of 3GPP 5G network by using 5G-GUTI as the device pseudo identity. Since 5G-GUTI updates periodically and randomly, the eavesdropper could only know that there are some D2D connections over the network, but it is very difficult for adversaries to use these messages to perform location tracking or identification of a specific device.

Traceability and free-riding attack resistant: our protocol requires UE_A to send a success indication message upon successful connection establishment. This message should be a trustworthy indicator of a successful session as it contains the random nonce which can only be found in the DHKE response from UE_B . Then, since the message contains the identity of both UE_A and UE_B , 5GC can use it to track all D2D connections. Besides, 5GC can also use it to record the total number of D2D connections of every UEs. For each connection successful event, 5GC can increase the credibility of UE_A in that connection and increase the connection counters of both UE_A and UE_B . Consequently, if there are some D2D UEs having excessively high number of connection count but low credibility, it is more likely that these UEs are the free riders who always connect other UEs to get data but seldom share data by allowing incoming connection. In this case, 5GC can choose to reject future D2D connection requests from these devices.

Message non-repudiation: in our protocol, every message is sent through the trusted channel, or the insecure channel with either HMAC code or sequence number appended: the broadcast message, DHKE request message and DHKE response message are protected using HMAC code. The D2D connect request, discoveree found message and discoverer continue message are transmitted through secured channel protected by 5G-AKA. Last but not least, the D2D symmetric key based encryption is appended with sequence number before encryption. Hence, it ensures the data sender cannot deny the originality of the message.

4.5. Performance Evaluation

In order to analyze the performance of our proposed protocol, three different D2D schemes, namely 5G-IoT D2D [38], LIKE [39] and UAKA-D2D [83] are included for comparison. The performances of those 3 schemes have been evaluated in terms of computational overhead, transmission overhead, storage usage for UE during protocol execution, average delays incurred by unknown attacks and energy consumption to show that our proposed scheme improves the overall security with the lowest performance overheads.

4.5.1. Computational Overhead

This chapter evaluates the computational time among all schemes. All simulations are conducted on a computer with Intel® Core™ i5-3210M CPU @ 2.5 GHz CPU and 8GB of RAMs. We use Bouncy Castle v1.64 with Java SDK 13.0.1 to write simulation code for all cryptography calculations and time measurement. Bouncy Castle API is a widely used lightweight cryptography API which follows the Java Cryptography Architecture (JCA), and it is also supported in Android environment. Since there are various cryptographic functions used in each scheme, and each function has its own key size requirement to achieve the similar security level, in order to holistically and equally evaluate their performance, we follow the NIST recommendation [118] to use the same 256-bit equivalent key strength throughout all simulations.

Consequently, for all ECC-based cryptographic functions including the Elliptic Curve Integrated Encryption Scheme (ECIES), ECDH and ECDSA, we employ the same 256-bit elliptic curve (secp256k1) and SHA256withECDSA digital signature. For modular exponentiation based cryptographic functions such as DHKE, we employ the equivalent key size of 3072 bits according to the NIST recommendation. Also, for symmetric key cryptography and HMAC, the equivalent key size of 256 bits and HMACSHA256 are chosen. Finally, Table 6 shows the experimental computational time of running the 3 different algorithms with the aforementioned parameters. The results showed that overall, ECC-based key exchange takes much less computational time than modular exponential based key exchanges. Also, it proves that HMAC takes only a small amount of computational time.

Moreover, to find the theoretical computational time, all related schemes are carefully studied. Using LT-AKA as an example, AMF needs to generate HMAC keys for UE_A and UE_B in the initialization response messages (consumes two $T_{HMAC-KeyGen}$) during UE initialization. In the device discovery stage, UE_A needs to broadcast its GUTI with one HMAC generated (consumes one $T_{HMAC-Run}$). After that, in the link setup stage, both UE_A and UE_B need to generate ECDH key pairs (consumes two $T_{ECC-GenKeyPair}$), calculate the ECDH session key (consumes two $T_{ECDH-CommonKey}$), and verify and generate the HMACs in the DHKE request and response messages (2 verifications and 2 generations, totally 4 $T_{HMAC-Run}$). At the same

time, AMF needs to verify the HMAC in the connect request message (consumes one $T_{\text{HMAC-Run}}$), generate one token in the connect request accepted message (consumes one T_{GenToken}), and one common HMAC key for the connect request accepted message, and the discoverer found message (consumes one T_{GenToken}). Using the same way, the theoretical computational time of all related schemes can be derived successfully. Combining the experimental computational time in Table 6, the total experimental time of the related schemes can be derived in Table 7.

According to the table, as our proposed scheme uses symmetric HMAC operations to provide message authenticity, it is much faster than the 5G-IoT D2D [38] and LIKE [39] which employ asymmetric ECDSA digital signature. On the other hand, although UAKA-D2D [83] needs the least computational time in AMF, it takes the longest time among all schemes because of the time-consuming modular exponentiation based key exchanges at the UE side. Overall, it shows that our proposed scheme can achieve the lowest total experimental computational time among all 3 schemes.

Table 6 - Experimental Computational Time for Algorithms

	LT-AKA	5G-IoT D2D	LIKE	UAKA-D2D
$T_{\text{ECDSA-Sign}}$	/	5.05	3.21	/
$T_{\text{ECDSA-Verify}}$	/	4.93	2.30	/
$T_{\text{ECC-GenKeyPair}}$	3.22	1.43	1.20	/
$T_{\text{MODP-GenKeyPair}}$	/	/	/	35.28
$T_{\text{ECDH-CommonKey}}$	5.07	2.41	1.96	/
$T_{\text{DHKE-CommonKey}}$	/	/	/	35.08
T_{GenToken}	0.05	/	/	/
$T_{\text{ECIES-Encrypt}}$	/	/	7.09	/
$T_{\text{ECIES-Decrypt}}$	/	/	2.64	/
T_{SHA1}	/	/	0.02	/
$T_{\text{HMAC-KeyGen}}$	0.06	/	/	0.04
$T_{\text{HMAC-Run}}$	0.28	/	/	0.39

Table 7 - Theoretical and Experimental Computational Time

		LT-AKA	5G-IoT D2D	LIKE	UAKA-D2D
	<i>Entity</i>	<i>Operations</i>			
Initialization	UE	/	/	2 * $T_{\text{ECIES-Enc}}$	/
	AMF	2 * $T_{\text{HMAC-KeyGen}}$	1 * $T_{\text{ECC-GenKeyPair}}$ 2 * $T_{\text{ECDSA-Sign}}$	3 * $T_{\text{ECC-GenKeyPair}}$ 2 * $T_{\text{ECIES-Dec}}$ 2 * T_{SHA1}	/
Device Discovery	UE	1 * $T_{\text{HMAC-Run}}$	/	1 * T_{SHA1}	1 * $T_{\text{HMAC-KeyGen}}$
	AMF	/	/	/	/
Link Setup	UE	2 * $T_{\text{ECDH-CommonKey}}$ 2 * $T_{\text{ECC-GenKeyPair}}$ 4 * $T_{\text{HMAC-Run}}$	2 * $T_{\text{ECDH-CommonKey}}$ 2 * $T_{\text{ECC-GenKeyPair}}$	2 * $T_{\text{ECDH-CommonKey}}$ 2 * $T_{\text{ECDSA-Sign}}$ 2 * $T_{\text{ECDSA-Verify}}$ 1 * T_{SHA1}	2 * $T_{\text{DHKE-CommonKey}}$ 2 * $T_{\text{MODP-GenKeyPair}}$ 4 * $T_{\text{HMAC-Run}}$
	AMF	1 * $T_{\text{HMAC-Run}}$ 1 * T_{GenToken} 1 * $T_{\text{HMAC-KeyGen}}$	2 * $T_{\text{ECDSA-Verify}}$	3 * T_{SHA1}	1 * $T_{\text{HMAC-KeyGen}}$
Experimental Time (ms)		18.46	29.07	38.10	142.36

4.5.2. Transmission Overhead

Bandwidth resources in wireless channel is invaluable, so smaller transmission overhead is always preferred. Our transmission overhead calculation consists of two parts: propagation delays and transmission delays. For propagation delays, it is mainly controlled by the physical distance between two devices (i.e., UE to UE distance and UE to gNB distance), and the total number of signaling messages in the protocol. For transmission delays, it is determined by the length of controlling messages in the protocol, and the data rate of the network. In order to reflect the daily life use case, we adopted the data rate of the city area (urban macro) coverage scenario listed in 3GPP 5G service requirement TS 22.261 [2]. For the propagation delays, inter-site distance (ISD)/2 = 250 meters is chosen to simulate the average UE-UE distance, and the propagation speed of the signals is $3 \times 10^8 \text{ ms}^{-1}$. For transmission delays, the data rate for uplink is 25Mbit/s, and the downlink is 50Mbit/s. Also, for the length of signaling messages in all protocols, all key lengths are calculated based on the NIST 256-bit key strength assumption. For other parameters, the length of 5G-GUTI is 80 bits, 5G-SUCI is 256 bits, message tag is 8 bits, random nonce is 32 bits, timestamp is 32 bits, and the session ID is 64 bits.

Consequently, the lengths of signaling messages of all related protocols are derived in Table 8, and the calculated total overheads combining propagation delays and transmission delays are demonstrated in Figure 27. The simulation result shows that our scheme achieves the best in terms of the lowest transmission overheads, as well as the lowest total overheads among all related schemes. In fact, since the length of symmetric key of 256-bit HMAC is two times shorter than the length of the asymmetric key of SHA256withECDSA, our proposed protocol provides a relatively shorter overall message length than all related schemes.

Table 8 – Total Signaling Message Length in bits

	LT-AKA	5G-IoT-D2D	LIKE	UAKA-D2D
Initialize Broadcast	N/A	N/A	1040	N/A
Initialize Request	8	8	800	N/A
Initialize Response	256	576	792	3080
Msg #1	344	832	1400	336
Msg #2	424	832	1040	160
Msg #3	376	832	1560	400
Msg #4	344	832	688	400
Msg #5	944	8	1264	3528
Msg #6	976	8	736	3528
Msg #7	200	1040	8	N/A
Msg #8	N/A	1040	N/A	N/A
Total	3872	6008	9328	11432

Figure 27 - Total Computational and Transmission Overheads

4.5.3. Storage Overhead for UE

Storage overhead is the theoretical amount of storage space required to initialize the protocol. Since the amount of memory in mobile devices is very limited, smaller storage overhead is always preferred. In fact, the storage overhead involves key parameters and all distributed materials from CN including pseudo-identities, tokens and private keys. For key parameters, we assume an elliptic curve can be specified using a sextuple $T_{EC} = (p, a, b, G, n, h)$ where p is the finite field, a and b are the coefficients of the curve, G is the base point, n is the order and h is the co-factor. The sizes of sextuples follow the recommended elliptic curve domain parameters listed in [109]. For modular exponentiation based key parameters, it can be specified using a 2-tuple $T_{MODP} = (G, P)$ where G is the generator and P is the prime. The sizes of 2-tuples follow the modular exponentiation (MODP) parameters used in the Internet Key Exchange (IKE) standard listed in [123], [124]. Evaluating all related schemes, our scheme involves T_{EC} , a 5G-GUTI pseudo-identity, and a distributed HMAC Key K_A . For 5G-IoT-D2D [38], it involves T_{EC} , a 5G-SUCI pseudo-identity, and a D2D token $D2DTK_{gNBx}$. For LIKE [39], it involves T_{EC} , a public key of security anchor, a pseudo-identity $Sx-ID$, and a public-private key pair. For UAKA-D2D [83], it involves T_{MODP} , a pseudo-identity (assuming 5G-GUTI since it is in the 5G network), and a symmetric key K_{D2D} for communicating with ProSe Function.

Finally, the calculated storage overheads of all related protocols in different NIST equivalent key strength are shown in Figure 28. It shows that our scheme has the lowest storage overhead for UE after protocol initialization in all different key strengths. It is because our proposed scheme employs the shorter ECDH instead of the lengthy modular exponentiation based DHKE. Also, our proposed scheme does not need to store the lengthy ECDSA asymmetric public private key pairs, and the use of 5G-GUTI is also shorter than the 5G-SUCI or the $Sx-ID$ used in other schemes.

Figure 28 - Storage Overhead for UE

4.5.4. Average Delays

Although our scheme can prevent some types of security attacks, there will always be some unknown new attacks in the future. We assume that after initialization, our protocol is facing either known or unknown attacks randomly all the time. For each step of the protocol, there will be a percentage of the chance that it will face an unknown attack. If the incoming attack is a known attack, which we have claimed to be able to be prevented, the protocol continues to the next step without interruption. However, if it is an unknown attack, our protocol will be interrupted immediately, so that the protocol must restart from initialization. Finally, we measure the average time needed to finish the protocol under this model (i.e., the time duration from the first initialization to the success of the protocol). To measure the time efficiently, we have built a simulation model in MATLAB 2019b to simulate the unknown attack assumptions aforementioned. By running it for one million times for each protocol, we can finally get the average time needed for completing all related protocols in Figure 29. It shows that our scheme achieves the lowest average delay among all related schemes. Also, when there are many unknown attacks in the system (i.e. the probability of unknown attacks reaches 80%), the performance improvement of our scheme becomes more significant: it is $(91.48-31.19)/91.48 = 65.9\%$ faster than the 5G-IoT D2D [38], $(138.4-31.19)/138.4 = 77.5\%$ faster than LIKE [39] and $(210.9-31.19)/210.9 = 85.2\%$ faster than UAKA-D2D [83]. As a result, it demonstrates that our protocol works far more efficient than the others in both normal and adverse situations with many unknown attacks.

Figure 29 - Average Delays Incurred due to Unknown Attacks

4.5.5. Energy Consumption for UE

To measure the energy consumption of a protocol holistically, we need to consider both the energy used for transmitting signaling messages and the energy used for running cryptographic functions. Since there is no mobile device running 5G D2D at present, we adopt the LTE data transfer power model carried out by Huang et. al. [120] to estimate the energy used for data transmission, and the energy cost approximation conducted by Potlapally et. al. [121] to estimate the energy used for running cryptographic functions. According to [120], for uplink data transmission, the energy cost for uplink transmission is $E_{ul} = (\alpha_u t_{udr} + \beta) t_{ul}$ mJ and downlink transmission is $E_{dl} = (\alpha_d t_{ddr} + \beta) t_{dl}$ mJ, where $\alpha_u = 438.39$, $\alpha_d = 51.79$, $\beta = 1288.04$, t_{udr} is the uplink throughput, t_{ddr} is the downlink throughput, and t_{ul} and t_{dl} are the time spent for uplink and downlink respectively. We follow the similar uplink and downlink throughput stated in the transmission overhead analysis.

On the other hand, for the energy used for running cryptographic functions, experiments were conducted using a battery-powered Compaq iPAQ H3670 PDA [121]. This PDA is equipped with Intel SA-1110 StrongARM processor clocked at 206MHz, 64MB of RAM, and 16MB of FlashROM. Consequently, for the energy cost for modular exponentiation based DHKE, we estimate the cost of key generation to be 876.96 mJ, and the cost of key exchange is 1046.5 mJ. For ECC based ECDH, the cost of key generation is 276.70 mJ, and the cost of key exchange is 163.5 mJ. For ECDSA, the cost of ECDSA signature is 134.20 mJ, and ECDSA verification is 196.23 mJ. For HMAC, its energy cost is 1.16 mJ, and the symmetric key generation cost is negligible. For SHA1, its energy cost is 0.76 mJ. For ECIES, since it is a hybrid encryption scheme combining both ECC, MAC and key encapsulation mechanism (KEM), we estimate the cost of encryption and decryption to be $E_{ECC} + E_{MAC} + E_{AEAD}$. By assuming SHA1 and AES as the default hashing and AEAD cipher algorithm, its cost would be approximately

$134.2 + 0.76 + 1.21x$ mJ, where x is the number of bytes in the cleartext.

Finally, the average energy consumption for a UE of all related protocols is shown in Figure 30. Since the discoveree UE (UE_A) and discoverer UE (UE_B) should have slightly different energy consumption, this calculation takes the average of them. It shows that our proposed scheme achieves the lowest energy consumption among all schemes. Moreover, to verify the energy consumption of the protocol under unknown attacks, we evaluated the total amount energy consumed by a UE pair under different percentage of unknown attacks in Figure 31. Using the similar statistical model in the calculation of average delays, this also shows that our proposed scheme uses the least energy even in the adverse situation with 80% unknown attacks. It is because our scheme uses HMAC to replace the power-consuming asymmetric ECDSA and use ECDH to replace the power-consuming modular exponentiation based DHKE. Also, thanks to the shorter message length of our proposed scheme, some of the energy used for transmission overhead is also saved.

Figure 30 - Energy Consumption of UE

Figure 31 - Total Energy Consumption of a UE Pair due to Unknown Attacks

4.6. Conclusion of the LT-AKA

We have proposed a novel lightweight traceable authentication and key agreement scheme for 5G D2D network. Based on ECC and HMAC, our scheme is computationally lightweight and therefore suitable for all resource-constrained 5G mobile and IoT devices. Our scheme works under the 3GPP 5G system model, and it enables network operators to trace D2D connection events through success indication messages. Security evaluation verifies that our protocol is resistant to many attacks including replay attack, MITM attack, and free-riding attacks, and the performance evaluation proves that our scheme achieves ideal costs compared with other existing schemes.

Chapter 5. Conclusion

In this chapter, the contributions of this thesis are summarized, and future research works are proposed.

5.1. Contributions

This thesis has explored many security issues in the 3GPP 5G network. It has carried out an extensive literature survey about the system architecture, security architecture, and security requirements of the 3GPP 5G networks. Then, several weaknesses in the 5G-AKA authentication scheme were identified and the current work in the field were studied. Moreover, it has chosen one of the promising technologies, 5G-D2D communication, as a further study of 5G security. Through the extensive literature review, some security issues in 5G-D2D were also found. Afterward, the thesis has improved the security in 5G and 5G-D2D network with the following two major contributions:

- First, a Secure Blockchain-based 5G Authentication and Key Agreement (5GSBA) protocol has been proposed in Chapter 3. The 5GSBA protocol combines some state-of-the-art cryptographic functions to ensure authentication security between gNBs and UEs. The security evaluation has proved that it is resistant to various network attacks, and the performance evaluation shows that it only consumes a tiny amount of extra overhead. Simulation results also show that it has a high resistivity to DoS and DDoS attacks.
- Second, a novel lightweight traceable authentication and key agreement scheme (LT-AKA) for 5G D2D networks has been proposed. The scheme is computationally lightweight and suitable for all resource-constrained 5G mobile and IoT devices. It works under the 3GPP 5G system model, and it enables network operators to trace D2D connection events through success indication messages. Security evaluation verifies that our protocol is resistant to attacks specifically found in D2D networks, and the performance evaluation proves that it achieves the lowest costs compared with other existing schemes.

5.2. Future Research Work

Although this thesis has vastly improved the security in 5G and 5G-D2D communication, network security is the cat-and-mouse game that never ends. In the future, the evolution of computers and algorithms will bring more new challenges. As a result, some future research work is recommended:

Quantum Resistant 5G Authentication: in recent years, Shor's algorithm has proved that all security functions based on modular arithmetic and elliptic curve cryptography are theoretically dangerous because they are hackable with quantum computers in the future. With the rapid development of quantum computer technology, all secure protocols designed for today may become vulnerable. Hence,

the existing 5G authentication protocols should also upgrade with some new cryptographic functions such as lattice-based cryptography to prevent quantum attacks in the future. However, since it may introduce extra computational and storage overheads, further investigations and designs are needed.

Blockchain-based 5G-D2D Authentication: credibility and fairness of all D2D users have always been the issues hindering the adoption of D2D communication. Although our proposed LT-AKA scheme has improved it by tracing the connection establishment of all UEs and only allowing legitimate UEs to join, LT-AKA still relies on the AMF server, which could introduce the single-point-of-failure. Consequently, using blockchain to record all devices in a secure and immutable manner could improve this situation in the future.

Machine Learning-Based Intrusion Detection: the 5GSBA and LT-AKA proposed in this thesis have prevented attacks from outside attackers (i.e., Dolev-Yao adversary model). However, in the future, there could be more malware that compromises the device internally to steal private information from UEs. Since the network intrusion detection systems (NIDS) implemented with deep neural networks or recurrent neural networks are having higher accuracy and low false-positive rate in recent years, it is possible to integrate it into the future cellular network. By doing so, we could monitor suspicious data traffic and alert the users about these events as early as possible.

List of Publication

This following list shows all publications being accepted related to this project:

- [1] M. C. Chow and M. Ma, "A Lightweight D2D Authentication Scheme against Free-riding Attacks in 5G Cellular Network," *Proceedings of 2020 International Electronics Communication Conference, IECC 2020*, 2020.
- [2] M. C. Chow and M. Ma, "Secure D2D in 5G Cellular Network: Architecture, Requirements and Solutions," *Advances in Computing, Informatics, Networking and Cybersecurity*, Springer Nature, 2021 (Accepted).
- [3] M. C. Chow and M. Ma, "A Blockchain-Enabled 5G Authentication Scheme Against DoS Attacks," *Proceedings of 2020 International Conference on Electronics, Communications and Information Technology, CECIT 2020*, 2020.
- [4] M. C. Chow and M. Ma, "A Secure Blockchain-based Authentication and Key Agreement Scheme for 3GPP 5G Networks," *Computer Networks* (Under Review).
- [5] M. C. Chow and M. Ma, "A Lightweight Traceable D2D Authentication and Key Agreement Scheme in 5G Cellular Networks," *Computers and Electrical Engineering* (Under Review).

Bibliography

- [1] N. Panwar, S. Sharma, and A. K. Singh, "A survey on 5G: The next generation of mobile communication," *Phys. Commun.*, vol. 18, pp. 64–84, 2016, doi: 10.1016/j.phycom.2015.10.006.
- [2] 3GPP, "Technical Specification Group Services and System Aspects; Service requirements for the 5G system; Stage 1 (Release 17); TS 22.261." 2019.
- [3] Y. Lu and X. Zheng, "6G: A survey on technologies, scenarios, challenges, and the related issues," *J. Ind. Inf. Integr.*, vol. 19, no. June, p. 100158, 2020, doi: 10.1016/j.jii.2020.100158.
- [4] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016, doi: 10.1109/COMST.2016.2532458.
- [5] 3GPP, "System architecture for the 5G System (5GS) (Release 15.7.0); TS 23.501." p. 353, 2019, [Online]. Available: <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>.
- [6] 3GPP, "Technical Specification Group Services and System Aspects; Proximity-based services (ProSe) (Release 15); TS23.303." Jun. 2018, [Online]. Available: <https://www.3gpp.org/DynaReport/23303.htm>.
- [7] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014, doi: 10.1109/COMST.2014.2319555.
- [8] 3GPP, "Technical Specification Group Services and System Aspects; Feasibility study for Proximity Services (ProSe) (Release 12); TS22.803." 2013, [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=653>.
- [9] R. I. Ansari *et al.*, "5G D2D Networks: Techniques, Challenges, and Future Prospects," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3970–3984, 2018, doi: 10.1109/JSYST.2017.2773633.
- [10] D. Feng, L. Lu, Y. W. Yi, G. Y. Li, G. Feng, and S. Li, "Device-to-device communications underlying cellular networks," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3541–3551, Aug. 2013, doi: 10.1109/TCOMM.2013.071013.120787.
- [11] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, and M. A. Javed, "A survey of device-to-device communications: Research issues and challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 3, pp. 2133–2168, 2018, doi: 10.1109/COMST.2018.2828120.
- [12] A. Zhang and X. Lin, "Security-Aware and Privacy-Preserving D2D Communications in 5G," *IEEE Netw.*, vol. 31, no. 4, pp. 70–77, 2017, doi: 10.1109/MNET.2017.1600290.
- [13] M. Usman, A. A. Gebremariam, U. Raza, and F. Granelli, "A Software-Defined Device-to-Device Communication Architecture for Public Safety Applications in 5G Networks," *IEEE Access*, vol. 3, pp. 1649–1654, 2015, doi: 10.1109/ACCESS.2015.2479855.
- [14] M. Wang and Z. Yan, "A Survey on Security in D2D Communications," *Mob. Networks Appl.*, vol. 22, no. 2, pp. 195–208, Apr. 2017, doi: 10.1007/s11036-016-0741-5.
- [15] O. Nait Hamoud, T. Kenaza, and Y. Challal, "Security in device-to-device communications: a survey," *IET Networks*, vol. 7, no. 1, pp. 14–22, Jan. 2018, doi: 10.1049/iet-net.2017.0119.
- [16] X. Wu *et al.*, "FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1215–1228, 2013, doi: 10.1109/TNET.2013.2264633.
- [17] "Wi-Fi P2P Technical Specification v1.7." Wi-Fi Alliance, Dec. 2016.
- [18] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, "Security for 5G and beyond," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 4, pp. 3682–3722, 2019, doi: 10.1109/COMST.2019.2916180.
- [19] J. Cao, M. Ma, H. Li, Y. Zhang, and Z. Luo, "A survey on security aspects for LTE and LTE-A networks," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 283–302, 2014, doi: 10.1109/SURV.2013.041513.00174.
- [20] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma, and J. Ott, "Security and Privacy in Device-to-Device (D2D) Communication: A Review," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 2, pp. 1054–1079, 2017, doi: 10.1109/COMST.2017.2649687.
- [21] S. Zhang, Y. Wang, and W. Zhou, "Towards secure 5G networks: A Survey," *Comput. Networks*, vol. 162, p. 106871, Oct. 2019, doi: 10.1016/j.comnet.2019.106871.
- [22] M. Wang and Z. Yan, "Security in D2D communications: A review," in *Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015*, Aug. 2015, vol. 1, pp. 1199–1204, doi: 10.1109/Trustcom.2015.505.
- [23] 3GPP, "Security architecture and procedures for 5G System (Release 16.3.0); TS 33.501." 2018, [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/33_series/33.501/33501-g30.zip.
- [24] C. Lai, H. Li, R. Lu, and X. Shen, "SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks," *Comput. Networks*, vol. 57, no. 17, pp. 3492–3510, 2013, doi: 10.1016/j.comnet.2013.08.003.

- [25] J. Cao, M. Ma, and H. Li, "LPPA: Lightweight privacy-preservation access authentication scheme for massive devices in fifth Generation (5G) cellular networks," *Int. J. Commun. Syst.*, vol. 32, no. 3, pp. 1–24, 2019, doi: 10.1002/dac.3860.
- [26] P. Yu, J. Cao, M. Ma, H. Li, B. Niu, and F. Li, "Quantum-Resistance Authentication and Data Transmission Scheme for NB-IoT in 3GPP 5G Networks," *IEEE Wirel. Commun. Netw. Conf. WCNC*, vol. 2019-April, pp. 1–7, 2019, doi: 10.1109/WCNC.2019.8885686.
- [27] A. Braeken, M. Liyanage, P. Kumar, and J. Murphy, "Novel 5G Authentication Protocol to Improve the Resistance Against Active Attacks and Malicious Serving Networks," *IEEE Access*, vol. 7, pp. 64040–64052, 2019, doi: 10.1109/ACCESS.2019.2914941.
- [28] I. Gharsallah, S. Smaoui, and F. Zarai, "A secure efficient and lightweight authentication protocol for 5G cellular networks: SEL-AKA," *2019 15th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2019*, pp. 1311–1316, 2019, doi: 10.1109/IWCMC.2019.8766448.
- [29] X. Hu, C. Liu, S. Liu, and X. Cheng, "A security enhanced 5g authentication scheme for insecure channel," *IEICE Trans. Inf. Syst.*, vol. E103D, no. 3, pp. 711–713, 2020, doi: 10.1587/transinf.2019EDL8190.
- [30] J. Cao, Z. Yan, R. Ma, Y. Zhang, Y. Fu, and H. Li, "LSAA: A lightweight and secure access authentication scheme for both UE and mMTC devices in 5G networks," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5329–5344, 2020, doi: 10.1109/JIOT.2020.2976740.
- [31] Z. Haddad, M. M. Fouda, M. Mahmoud, and M. Abdallah, "Blockchain-based Authentication for 5G Networks," *2020 IEEE Int. Conf. Informatics, IoT, Enabling Technol. ICIOT 2020*, pp. 189–194, 2020, doi: 10.1109/ICIOT48696.2020.9089507.
- [32] M. Khan and V. Niemi, "Privacy Enhanced Fast Mutual Authentication in 5G Network Using Identity Based Encryption," *J. ICT Stand.*, vol. 5, no. 1, pp. 69–90, 2017, doi: 10.13052/jicts2245-800x.514.
- [33] Y. Javed, A. S. Khan, A. Qahar, and J. Abdullah, "EEoP: A lightweight security scheme over PKI in D2D cellular networks," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 3–11, pp. 99–105, 2017, [Online]. Available: <http://journal.utem.edu.my/index.php/jtec/article/view/3191>.
- [34] A. Abro, Z. Deng, and K. A. Memon, "A lightweight elliptic-ElGamal-based authentication scheme for secure device-to-device communication," *Futur. Internet*, vol. 11, no. 5, p. 108, May 2019, doi: 10.3390/fi11050108.
- [35] N. N. Dao, W. Na, Y. Lee, D. N. Vu, and S. Cho, "Prefetched Asymmetric Authentication for Infrastructureless D2D Communications: Feasibility Study and Analysis," in *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, Oct. 2018, pp. 1053–1054, doi: 10.1109/ICTC.2018.8539475.
- [36] R. H. Hsu, J. Lee, T. Q. S. Quek, and J. C. Chen, "GRAAD: Group Anonymous and Accountable D2D Communication in Mobile Networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 2, pp. 449–464, Feb. 2018, doi: 10.1109/TIFS.2017.2756567.
- [37] H. Kwon, D. Kim, C. Hahn, and J. Hur, "Secure authentication using ciphertext policy attribute-based encryption in mobile multi-hop networks," *Multimed. Tools Appl.*, vol. 76, no. 19, pp. 19507–19521, Oct. 2017, doi: 10.1007/s11042-015-3187-z.
- [38] B. Seok, J. C. S. Sicato, T. Erzhen, C. Xuan, Y. Pan, and J. H. Park, "Secure D2D communication for 5G IoT network based on lightweight cryptography," *Appl. Sci.*, vol. 10, no. 1, 2020, doi: 10.3390/app10010217.
- [39] S. B. M. Baskaran and G. Raja, "A Lightweight Incognito Key Exchange Mechanism for LTE-A Assisted D2D Communication," *2017 9th Int. Conf. Adv. Comput. ICoAC 2017*, pp. 301–307, 2018, doi: 10.1109/ICoAC.2017.8441370.
- [40] 3GPP, "Security architecture and procedures for 5G System (Release 16.3.0); TS 33.501." 2020, [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/33_series/33.501/33501-g30.zip.
- [41] J. Cao *et al.*, "A survey on security aspects for 3GPP 5G networks," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 1, pp. 170–195, 2020, doi: 10.1109/COMST.2019.2951818.
- [42] F. Liu, J. Peng, and M. Zuo, "Toward a Secure Access to 5G Network," *Proc. - 17th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. Trust. 2018*, pp. 1121–1128, 2018, doi: 10.1109/TrustCom/BigDataSE.2018.00156.
- [43] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on authentication enhancements in 5G System; (Release 17); 3GPP TR 33.846 V0.7.0." 2020.
- [44] D. Basin, S. Radomirovic, J. Dreier, R. Sasse, L. Hirschi, and V. Stettler, "A formal analysis of 5g authentication," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 1383–1396, 2018, doi: 10.1145/3243734.3243846.
- [45] J. Cao, H. Li, M. Ma, Y. Zhang, and C. Lai, "A simple and robust handover authentication between HeNB and eNB in LTE networks," *Comput. Networks*, vol. 56, no. 8, pp. 2119–2131, 2012, doi: 10.1016/j.comnet.2012.02.012.

- [46] Y. E. H. El Idrissi, N. Zahid, and M. Jedra, "An efficient authentication protocol for 5G heterogeneous networks," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10542 LNCS, pp. 496–508, 2017, doi: 10.1007/978-3-319-68179-5_43.
- [47] K. A. Alezabi, F. Hashim, S. J. Hashim, B. M. Ali, and A. Jamalipour, "Efficient authentication and re-authentication protocols for 4G/5G heterogeneous networks," *Eurasip J. Wirel. Commun. Netw.*, vol. 2020, no. 1, 2020, doi: 10.1186/s13638-020-01702-8.
- [48] A. Braeken, "Symmetric key based 5G AKA authentication protocol satisfying anonymity and unlinkability," *Comput. Networks*, vol. 181, no. April, 2020, doi: 10.1016/j.comnet.2020.107424.
- [49] F. Y. Leu, K. L. Tsai, H. Susanto, C. Y. Gu, and I. You, "A Fault Tolerant Mechanism for UE Authentication in 5G Networks," *Mob. Networks Appl.*, 2020, doi: 10.1007/s11036-019-01502-5.
- [50] C. Lai, H. Li, R. Lu, R. Jiang, and X. Shen, "LGTH: A lightweight group authentication protocol for machine-type communication in LTE networks," *GLOBECOM - IEEE Glob. Telecommun. Conf.*, pp. 832–837, 2013, doi: 10.1109/GLOCOM.2013.6831176.
- [51] J. Choi and S. Jung, "A handover authentication using credentials based on chameleon hashing," *IEEE Commun. Lett.*, vol. 14, no. 1, pp. 54–56, Jan. 2010, doi: 10.1109/LCOMM.2010.01.091607.
- [52] J. Yang, X. Ji, K. Huang, Y. Chen, X. Xu, and M. Yi, "Unified and fast handover authentication based on link signatures in 5G SDN-based HetNet," *IET Commun.*, vol. 13, no. 2, pp. 144–152, 2019, doi: 10.1049/iet-com.2018.5405.
- [53] K. Han, M. Ma, X. Li, Z. Feng, and J. Hao, "An Efficient Handover Authentication Mechanism for 5G Wireless Network," *IEEE Wirel. Commun. Netw. Conf. WCNC*, vol. 2019-April, pp. 0–7, 2019, doi: 10.1109/WCNC.2019.8885915.
- [54] A. Chaer, K. Salah, C. Lima, P. P. Ray, and T. Sheltami, "Blockchain for 5G: Opportunities and challenges," *2019 IEEE Globecom Work. GC Wkshps 2019 - Proc.*, 2019, doi: 10.1109/GCWkshps45667.2019.9024627.
- [55] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2019.
- [56] C. Gorenflo, S. Lee, L. Golab, and S. Keshav, "FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second," *ICBC 2019 - IEEE Int. Conf. Blockchain Cryptocurrency*, pp. 455–463, 2019, doi: 10.1109/BLOC.2019.8751452.
- [57] M. Tahir, M. H. Habaebi, M. Dabbagh, A. Mughees, A. Ahad, and K. I. Ahmed, "A Review on Application of Blockchain in 5G and beyond Networks: Taxonomy, Field-Trials, Challenges and Opportunities," *IEEE Access*, vol. 8, pp. 115876–115904, 2020, doi: 10.1109/ACCESS.2020.3003020.
- [58] Q. Kong, L. Su, and M. Ma, "Achieving Privacy-Preserving and Verifiable Data Sharing in Vehicular Fog With Blockchain," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–10, 2020, doi: 10.1109/tits.2020.2983466.
- [59] S. Zhang and J. H. Lee, "A Group Signature and Authentication Scheme for Blockchain-Based Mobile-Edge Computing," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4557–4565, 2020, doi: 10.1109/JIOT.2019.2960027.
- [60] Y. Zhang, R. Deng, E. Bertino, and D. Zheng, "Robust and Universal Seamless Handover Authentication in 5G HetNets," *IEEE Trans. Dependable Secur. Comput.*, vol. 47907, no. c, pp. 1–1, 2019, doi: 10.1109/tdsc.2019.2927664.
- [61] H. Lee and M. Ma, "Blockchain-based mobility management for 5G," *Futur. Gener. Comput. Syst.*, 2019, doi: 10.1016/j.future.2019.08.008.
- [62] J. Xu, K. Xue, H. Tian, J. Hong, D. S. L. Wei, and P. Hong, "An Identity Management and Authentication Scheme Based on Redactable Blockchain for Mobile Networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6688–6698, 2020, doi: 10.1109/TVT.2020.2986041.
- [63] H. Yang, H. Zheng, J. Zhang, Y. Wu, Y. Lee, and Y. Ji, "Blockchain-based trusted authentication in cloud radio over fiber network for 5G," in *2017 16th International Conference on Optical Communications and Networks (ICOON)*, Aug. 2017, pp. 1–3, doi: 10.1109/ICOON.2017.8121598.
- [64] T. Hewa, A. Bracken, M. Ylianttila, and M. Liyanage, "Blockchain-based Automated Certificate Revocation for 5G IoT," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Jun. 2020, pp. 1–7, doi: 10.1109/ICC40277.2020.9148820.
- [65] Z. Chen, S. Chen, H. Xu, and B. Hu, "A security authentication scheme of 5G ultra-dense network based on block chain," *IEEE Access*, vol. 6, pp. 55372–55379, 2018, doi: 10.1109/ACCESS.2018.2871642.
- [66] Y. Wu, A. Khisti, C. Xiao, G. Caire, K. K. Wong, and X. Gao, "A Survey of Physical Layer Security Techniques for 5G Wireless Networks and Challenges Ahead," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 4, pp. 679–695, Apr. 2018, doi: 10.1109/JSAC.2018.2825560.
- [67] N. Yang, L. Wang, G. Geraci, M. ElKashlan, J. Yuan, and M. Di Renzo, "Safeguarding 5G wireless communication networks using physical layer security," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 20–27, Apr. 2015, doi: 10.1109/MCOM.2015.7081071.
- [68] R. Zhang, X. Cheng, and L. Yang, "Joint power and access control for physical layer security in D2D communications underlying cellular networks," in *2016 IEEE International Conference on*

- Communications, ICC 2016*, May 2016, pp. 1–6, doi: 10.1109/ICC.2016.7511531.
- [69] L. Wang, J. Liu, M. Chen, G. Gui, and H. Sari, “Optimization-Based Access Assignment Scheme for Physical-Layer Security in D2D Communications Underlying a Cellular Network,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 5766–5777, Jul. 2018, doi: 10.1109/TVT.2017.2789022.
 - [70] D. Ying and F. Ye, “D2D-Assisted Physical-Layer Security in Next-Generation Mobile Network,” in *2019 International Conference on Computing, Networking and Communications, ICNC 2019*, Feb. 2019, pp. 324–328, doi: 10.1109/ICCNC.2019.8685362.
 - [71] H. He and P. Ren, “Secure Communications in Cooperative D2D Networks by Jointing Wyner’s Code and Network Coding,” *IEEE Access*, vol. 7, pp. 34533–34540, 2019, doi: 10.1109/ACCESS.2019.2903253.
 - [72] X. Chen, Y. Zhao, Y. Li, X. Chen, N. Ge, and S. Chen, “Social trust aided D2D communications: Performance bound and implementation mechanism,” *IEEE J. Sel. Areas Commun.*, vol. 36, no. 7, pp. 1593–1608, Jul. 2018, doi: 10.1109/JSAC.2018.2825658.
 - [73] S. Yan, Y. Shang, and Y. Huang, “Social-aware jamming allocation for physical-layer security in D2D multicast network,” in *2018 IEEE Conference on Communications and Network Security, CNS 2018*, May 2018, pp. 1–7, doi: 10.1109/CNS.2018.8433158.
 - [74] L. Li, Y. Hu, H. Zhang, W. Liang, and A. Gao, “Deep learning based physical layer security of D2D underlay cellular network,” *China Commun.*, vol. 17, no. 2, pp. 93–106, 2020, doi: 10.23919/JCC.2020.02.008.
 - [75] E. Abd-Elrahman, H. Ibn-Khedher, H. Afifi, and T. Toukabri, “Fast group discovery and non-repudiation in D2D communications using IBE,” in *IWCMC 2015 - 11th International Wireless Communications and Mobile Computing Conference*, Aug. 2015, pp. 616–621, doi: 10.1109/IWCMC.2015.7289154.
 - [76] E. Abd-Elrahman, H. Ibn-Khedher, and H. Afifi, “D2D group communications security,” in *International Conference on Protocol Engineering, ICPE 2015 and International Conference on New Technologies of Distributed Systems, NTDS 2015 - Proceedings*, Jul. 2015, pp. 1–6, doi: 10.1109/NOTERE.2015.7293504.
 - [77] Y. Sun, J. Cao, M. Ma, H. Li, B. Niu, and F. Li, “Privacy-Preserving Device Discovery and Authentication Scheme for D2D Communication in 3GPP 5G HetNet,” in *2019 International Conference on Computing, Networking and Communications, ICNC 2019*, Feb. 2019, pp. 425–431, doi: 10.1109/ICCNC.2019.8685499.
 - [78] R. Sedidi and A. Kumar, “Key exchange protocols for secure Device-to-Device (D2D) communication in 5G,” in *IFIP Wireless Days*, Mar. 2016, vol. 2016-April, pp. 1–6, doi: 10.1109/WD.2016.7461477.
 - [79] M. Schmittner, A. Asadi, and M. Hollick, “SEMUD: Secure multi-hop device-to-device communication for 5G public safety networks,” in *2017 IFIP Networking Conference, IFIP Networking 2017 and Workshops*, Jun. 2017, vol. 2018-Janua, pp. 1–9, doi: 10.23919/IFIPNetworking.2017.8264846.
 - [80] C. Suraci, S. Pizzi, A. Iera, and G. Araniti, “Enhance the protection of transmitted data in 5G D2D communications through the Social Internet of Things,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, Sep. 2018, vol. 2018-Septe, pp. 376–380, doi: 10.1109/PIMRC.2018.8580860.
 - [81] W. Shen, W. Hong, X. Cao, B. Yin, D. M. Shila, and Y. Cheng, “Secure key establishment for device-to-device communications,” in *2014 IEEE Global Communications Conference, GLOBECOM 2014*, Dec. 2014, pp. 336–340, doi: 10.1109/GLOCOM.2014.7036830.
 - [82] W. Shen, B. Yin, X. Cao, L. X. Cai, and Y. Cheng, “Secure device-to-device communications over WiFi direct,” *IEEE Netw.*, vol. 30, no. 5, pp. 4–9, Sep. 2016, doi: 10.1109/MNET.2016.7579020.
 - [83] M. Wang, Z. Yan, and V. Niemi, “UAKA-D2D: Universal Authentication and Key Agreement Protocol in D2D Communications,” *Mob. Networks Appl.*, vol. 22, no. 3, pp. 510–525, Jun. 2017, doi: 10.1007/s11036-017-0870-5.
 - [84] T. Balan, A. Balan, and F. Sandu, “SDR Implementation of a D2D security cryptographic mechanism,” *IEEE Access*, vol. 7, pp. 38847–38855, 2019, doi: 10.1109/ACCESS.2019.2904909.
 - [85] A. Zhang, J. Chen, R. Q. Hu, and Y. Qian, “SeDS: Secure data sharing strategy for d2d communication in LTE-advanced networks,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2659–2672, Apr. 2016, doi: 10.1109/TVT.2015.2416002.
 - [86] R. Melki, H. N. Noura, and A. Chehab, “Lightweight and Secure D2D authentication & key management based on PLS,” in *IEEE Vehicular Technology Conference*, Sep. 2019, vol. 2019-Septe, pp. 1–7, doi: 10.1109/VTCFall.2019.8891531.
 - [87] W. Boubakri, W. Abdallah, and N. Boudriga, “Access control in 5G communication networks using simple PKI certificates,” in *2017 13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017*, Jun. 2017, pp. 2092–2097, doi: 10.1109/IWCMC.2017.7986606.
 - [88] O. N. Hamoud, T. Kenaza, and Y. Challal, “Towards using multiple KGC for CL-PKC to secure D2D communications,” in *2018 International Conference on Smart Communications in Network Technologies*,

- SaCoNeT 2018, Oct. 2018, pp. 283–287, doi: 10.1109/SaCoNeT.2018.8585671.
- [89] O. N. Hamoud, T. Kenaza, and Y. Challal, “A new certificateless system construction for multiple key generator centers to secure device-to-device communications,” in *ICETE 2019 - Proceedings of the 16th International Joint Conference on e-Business and Telecommunications*, 2019, vol. 2, pp. 84–95, doi: 10.5220/0007841500840095.
 - [90] Z. Yan, H. Xie, P. Zhang, and B. B. Gupta, “Flexible data access control in D2D communications,” *Futur. Gener. Comput. Syst.*, vol. 82, pp. 738–751, May 2018, doi: 10.1016/j.future.2017.08.052.
 - [91] M. Wang, Z. Yan, B. Song, and M. Atiquzzaman, “AAKA-D2D : Anonymous Authentication and Key Agreement Protocol in D2D Communications,” *2019 IEEE SmartWorld, Ubiquitous Intell. Comput. Adv. Trust. Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart City Innov.*, 2019, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00248.
 - [92] R. H. Hsu and J. Lee, “Group anonymous D2D communication with end-to-end security in LTE-A,” in *2015 IEEE Conference on Communications and Network Security, CNS 2015*, Sep. 2015, pp. 451–459, doi: 10.1109/CNS.2015.7346857.
 - [93] L. Wang, Y. Tian, D. Zhang, and Y. Lu, “Constant-round authenticated and dynamic group key agreement protocol for D2D group communications,” *Inf. Sci. (Ny)*, vol. 503, pp. 61–71, Nov. 2019, doi: 10.1016/j.ins.2019.06.067.
 - [94] Z. Shang, M. Ma, and X. Li, “A certificateless authentication protocol for D2D group communications in 5G cellular networks,” *2019 IEEE Glob. Commun. Conf. GLOBECOM 2019 - Proc.*, pp. 1–7, 2019, doi: 10.1109/GLOBECOM38437.2019.9014047.
 - [95] M. Wang and Z. Yan, “Privacy-preserving authentication and key agreement protocols for D2D group communications,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 8, pp. 3637–3647, Aug. 2018, doi: 10.1109/TII.2017.2778090.
 - [96] Y. Sun, J. Cao, M. Ma, Y. Zhang, H. Li, and B. NIU, “EAP-DDBA: Efficient Anonymity Proximity Device Discovery and Batch Authentication Mechanism for Massive D2D Communication Devices in 3GPP 5G HetNet,” *IEEE Trans. Dependable Secur. Comput.*, vol. 14, no. 8, pp. 1–1, 2020, doi: 10.1109/tdsc.2020.2989784.
 - [97] I. Abualhaol and S. Muegge, “Securing D2D wireless links by continuous authenticity with legitimacy patterns,” in *Proceedings of the Annual Hawaii International Conference on System Sciences*, Jan. 2016, vol. 2016-March, pp. 5763–5771, doi: 10.1109/HICSS.2016.713.
 - [98] H. Tan, Y. Song, S. Xuan, S. Pan, and I. Chung, “Secure D2D group authentication employing smartphone sensor behavior analysis,” *Symmetry (Basel)*, vol. 11, no. 8, p. 969, Aug. 2019, doi: 10.3390/sym11080969.
 - [99] A. Zhang, L. Wang, X. Ye, and X. Lin, “Light-Weight and Robust Security-Aware D2D-Assist Data Transmission Protocol for Mobile-Health Systems,” *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 3, pp. 662–675, Mar. 2017, doi: 10.1109/TIFS.2016.2631950.
 - [100] C. Zhou, “An improved lightweight certificateless generalized signcryption scheme for mobile-health system,” *Int. J. Distrib. Sens. Networks*, vol. 15, no. 1, p. 1550147718824465, Jan. 2019, doi: 10.1177/1550147718824465.
 - [101] L. Wang, Z. Li, M. Chen, A. Zhang, J. Cui, and B. Zheng, “Secure content sharing protocol for D2D users based on profile matching in social networks,” in *2017 9th International Conference on Wireless Communications and Signal Processing, WCSP 2017 - Proceedings*, Oct. 2017, vol. 2017-Janua, pp. 1–5, doi: 10.1109/WCSP.2017.8171117.
 - [102] B. Jin, D. Jiang, J. Xiong, L. Chen, and Q. Li, “D2D Data Privacy Protection Mechanism Based on Reliability and Homomorphic Encryption,” *IEEE Access*, vol. 6, pp. 51140–51150, 2018, doi: 10.1109/ACCESS.2018.2869575.
 - [103] D. Evans, V. Kolesnikov, and M. Rosulek, “A Pragmatic Introduction to Secure Multi-Party Computation,” *Found. Trends® Priv. Secur.*, vol. 2, no. 2–3, pp. 70–246, 2018, doi: 10.1561/33000000019.
 - [104] M. Li, N. Cao, S. Yu, and W. Lou, “FindU: Privacy-preserving personal profile matching in mobile social networks,” in *Proceedings - IEEE INFOCOM*, Apr. 2011, pp. 2435–2443, doi: 10.1109/INFCOM.2011.5935065.
 - [105] M. Li, S. Yu, N. Cao, and W. Lou, “Privacy-preserving distributed profile matching in proximity-based mobile social networks,” *IEEE Trans. Wirel. Commun.*, vol. 12, no. 5, pp. 2024–2033, May 2013, doi: 10.1109/TWC.2013.032513.120149.
 - [106] R. Zhang, J. Zhang, Y. Zhang, J. Sun, and G. Yan, “Privacy-preserving profile matching for proximity-based mobile social networking,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 656–668, Sep. 2013, doi: 10.1109/JSAC.2013.SUP.0513057.
 - [107] Q. Li, L. Huang, R. Mo, H. Huang, and H. Zhu, “Robust and scalable data access control in D2D communications,” *IEEE Access*, vol. 6, pp. 58858–58867, 2018, doi: 10.1109/ACCESS.2018.2874066.
 - [108] H. Wang, Y. Li, Y. Jiao, and Z. Jin, “An efficient outsourced attribute-based encryption scheme in the

- device-to-device mobile network,” *Int. J. Distrib. Sens. Networks*, vol. 15, no. 7, p. 155014771986550, Jul. 2019, doi: 10.1177/1550147719865507.
- [109] Certicom Research, “SEC2: Recommended Elliptic Curve Domain Parameters (Version 1.0),” 2000.
 - [110] E. Lombrozo, J. Lau, and P. Wuille, “BIP 0141 Segregated Witness (Consensus layer),” 2015. https://github.com/bitcoin/bips/blob/df3bf7bd1b641893991a69114e20087307647326/bip-0141.mediawiki#Block_size.
 - [111] M. Castro, “Practical Byzantine Fault Tolerance,” *Proc. Third Symp. Oper. Syst. Des. OSDI '99*, no. February, pp. 1–172, 2001, [Online]. Available: <http://pmg.csail.mit.edu/papers/osdi99.pdf>.
 - [112] R. M. Needham and M. D. Schroeder, “Using Encryption for Authentication in Large Networks of Computers,” *Commun. ACM*, vol. 21, no. 12, pp. 993–999, 1978, doi: 10.1145/359657.359659.
 - [113] G. Lowe, “An attack on the Needham-Schroeder public-key authentication protocol,” *Inf. Process. Lett.*, vol. 56, no. 3, pp. 131–133, 1995, doi: 10.1016/0020-0190(95)00144-2.
 - [114] M. Burrows, M. Abadi, and R. Needham, “A logic of Authentication,” *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990, doi: 10.1145/77648.77649.
 - [115] C. Cremers, “The Scyther Tool,” *CISPA Helmholtz Center for Information Security*, 2020. <https://people.cispa.io/cas.cremers/scyther/> (accessed Mar. 18, 2020).
 - [116] J. Mirkovic and P. Reiher, “A taxonomy of DDoS attack and DDoS defense mechanisms,” *Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, 2004, doi: 10.1145/997150.997156.
 - [117] J. A. Akinyele *et al.*, “Charm: a framework for rapidly prototyping cryptosystems,” *J. Cryptogr. Eng.*, vol. 3, no. 2, pp. 111–128, 2013, doi: 10.1007/s13389-013-0057-3.
 - [118] E. Barker, “Recommendation for Key Management – Part 1: General,” *NIST Spec. Publ. 800-57*, pp. 1–142, 2016, doi: 10.6028/NIST.SP.800-57pt3r1.
 - [119] F. D. Gunstone, “Global Mobile Market Report 2019,” 2019. [Online]. Available: <https://newzoo.com/insights/trend-reports/newzoo-global-mobile-market-report-2019-light-version/>.
 - [120] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, “A close examination of performance and power characteristics of 4G LTE networks,” *MobiSys'12 - Proc. 10th Int. Conf. Mob. Syst. Appl. Serv.*, pp. 225–238, 2012, doi: 10.1145/2307636.2307658.
 - [121] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha, “A study of the energy consumption characteristics of cryptographic algorithms and security protocols,” *IEEE Trans. Mob. Comput.*, vol. 5, no. 2, pp. 128–143, 2006, doi: 10.1109/TMC.2006.16.
 - [122] 3GPP, “Universal Mobile Telecommunications System (UMTS); LTE; Proximity-based Services (ProSe); Security aspects (3GPP TS 33.303 version 15.0.0 Release 15),” 2018.
 - [123] D. Harkins D, Carrel, “RFC 2409 - The Internet Key Exchange (IKE),” 1998. doi: 10.17487/rfc2409.
 - [124] T. Kivinen and M. Kojo, “RFC 3526 - More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE),” 2013.