



# **TRABALHO DE GRADUAÇÃO**

## **Group Authentication Protocols for Internet of Things (IoT) – QoS and Security Properties Evaluation**

**Ana Paula Golembiouski Lopes  
Lucas de Oliveira Hilgert**

**Brasília, dezembro de 2016**

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA**

## TRABALHO DE GRADUAÇÃO

# Group Authentication Protocols for Internet of Things (IoT) – QoS and Security Properties Evaluation

Ana Paula Golembiouski Lopes  
Lucas de Oliveira Hilgert

Relatório submetido como requisito parcial para obtenção  
do grau de Engenheiro de Redes de Comunicação

### Banca Examinadora

Prof. Paulo Roberto Gondim, UnB/ ENE (Orientador) \_\_\_\_\_

Profa. Cláudia Jacy Barenco Abbas, UnB/ENE \_\_\_\_\_

Prof. Leonardo R.A.X. Menezes, UnB/ ENE \_\_\_\_\_

## **Dedicatórias**

*Dedico este trabalho aos meus pais e à minha irmã, que me prestaram apoio fundamental nesta jornada acadêmica.*

*Ana Paula Golembiouski Lopes*

*Dedico este trabalho à minha família, amigos e a todos aqueles que me apoiaram e acreditaram no meu esforço.*

*Lucas de Oliveira Hilgert*

## **Agradecimentos**

*Agradeço a Deus por ter concedido esta e outras boas oportunidades em minha vida. Aos meus pais e a minha irmã, que me incentivam desde muito pequena a ser dedicada aos meus estudos e que sempre me guiaram da melhor forma possível para que eu pudesse alcançar meus sonhos e objetivos. Às minhas colegas de curso e amigas Aline Santos e Raquel Nascimento, pelo companheirismo e resiliência, aos desafios enfrentados da vida acadêmica na engenharia, desde o primeiro semestre de curso. À minha querida amiga Luciana Brito, por todos os anos de amizade e por ter compartilhado da minha jornada desde o início, sempre presente quando precisei. Às minhas amigas do Lewis 4th, pela cumplicidade e por estarem sempre disponíveis para me aconselhar e dar apoio. Por fim, a todos os meus professores, em especial ao meu orientador Paulo Gondim, pelos ensinamentos que possibilitaram a execução deste trabalho.*

*Ana Paula Golembiouski Lopes*

*Agradeço primeiramente a Deus por todas as bençãos que Ele tem me proporcionado e por ter colocado tantas pessoas boas no meu caminho. Agradeço a minha mãe que sempre me apoiou em tudo, me deu todo o suporte para que eu tivesse uma ótima educação e mesmo nos momentos mais difíceis que passamos ela nunca desistiu de mim e sempre esteve presente em minhas conquistas. Agradeço também a minha namorada em que além de ter me ajudado bastante na produção deste trabalho, ela também sempre me aguentou nos meus momentos de stress, sempre me apoiou nos momentos em que tudo dava errado e que comemorou sempre que algo de bom me acontecia. Agradeço também a minha colega de trabalho Ana Paula e meu orientador Paulo Gondim em que durante esse 1 ano tivemos um trabalho árduo, porém gratificante e com muito companheirismo e amizade. Por fim, agradeço a todos os meus familiares, amigos e colegas que me apoiaram ou que tiveram uma parcela neste trabalho. Obrigado a todos.*

*Lucas de Oliveira Hilgert*

## 1. INTRODUÇÃO

A Internet das Coisas (IoT, do inglês *Internet of Things*) tem sido caracterizada como uma forte tendência nos próximos anos, trazendo novas aplicações para quase todas as tarefas de nossa rotina diária. Ela irá revolucionar consideravelmente a maneira em que vivemos e interagimos com as coisas, através da conexão de praticamente tudo o que possa ser imaginado com a Internet. Algumas aplicações importantes são as redes elétricas inteligentes (Smart grid), que irão fornecer medição e monitoramento inteligente da energia, as cidades inteligentes, onde se prevê a integração e conexão de residências, serviços e transporte dentro dos centros urbanos, e o *Mobile healthcare (M-health)*, que proporcionará a coleta de dados da saúde dos pacientes através de sensores, encaminhando essas informações para os centros de tratamento por meio de uma ou mais redes de comunicação, com emprego de terminais móveis.

De forma integrada à IoT, uma outra forte tendência envolve a Comunicação do Tipo Máquina (MTC, do inglês *Machine-Type Communication*), para qual a rede LTE/LTE-A (Long Term Evolution/Long Term Evolution-Advanced) do 3GPP (3rd Generation Partnership Project) é forte candidata a receber o tráfego da MTC. De acordo com o 3GPP [1], sua arquitetura suporta cenários de roaming, onde os dispositivos possuem conexão apenas com o HSS (Home Subscriber Server) de sua rede de origem e com o MME (Mobility Management Entity) da rede servidora.

O protocolo de autenticação e acordo de chaves proposto pelo 3GPP para LTE é o EPS-AKA (Evolved Packet System – Authentication and Key Agreement), que autentica individualmente cada dispositivo que chega na rede visitada. Em um cenário com bilhões de dispositivos, um processo de autenticação que é executado completamente para cada dispositivo pode causar diversos problemas, como congestionamento causado pela alta no tráfego de sinalização e vulnerabilidades de segurança.

O problema da autenticação de grupos surge considerando que as aplicações MTC preveem a geração de grandes quantidades de dados, derivada de bilhões de dispositivos nos próximos anos. Os atuais padrões do 3GPP não estão adaptados para grandes grupos de dispositivos e seu uso para aplicações MTC se provou causador de falhas de segurança como perda de integridade, disponibilidade, confidencialidade e vulnerabilidade a diversos ataques.

Além disso, é importante ressaltar que a maioria destes dispositivos possuem recursos escassos, o que gera a necessidade de protocolos leves, com consumo reduzido de recursos computacionais e de comunicação. Consequentemente, o crescimento exponencial de dispositivos conectados requer métodos especiais de autenticação, para que ocorra prevenção de falhas de segurança e melhora na performance do sistema, providenciando todos os requisitos da MTC.

Baseado no problema descrito, o objetivo deste trabalho é o de propor dois protocolos leves para a autenticação de grupos de dispositivos para a IoT. O primeiro será baseado em ECDH (Elliptic Curve Diffie-Hellman) e em emparelhamento bilinear e o segundo, baseado no segredo de Shamir. Ambos capazes de executar a autenticação entre um grupo de dispositivos e um MME de maneira segura e eficiente, buscando reduzir custos computacionais e de comunicação. Consequentemente, buscando consumir menos recursos e otimizando suas performances e objetivos de segurança.

A organização do restante do trabalho é a seguinte: a seção 2 apresenta a fundamentação teórica, importante para o entendimento dos conceitos abordados no trabalho; A seção 3 apresenta o primeiro protocolo proposto, baseado em criptografia assimétrica e suas análises; A seção 4 apresenta o segundo protocolo proposto, baseado em criptografia simétrica e suas respectivas análises; por fim, a seção 5 apresenta as conclusões e indica possíveis trabalhos futuros.

## **2. FUNDAMENTAÇÃO TEÓRICA**

Este capítulo tem como objetivo introduzir os principais conceitos necessários para o entendimento do trabalho. Primeiro, são apresentadas definições de importantes propriedades de segurança, boa parte baseada em conceitos abordados por [2], como integridade, confidencialidade e disponibilidade, em seguida, são definidos os principais tipos de ataques atualmente executados no cenário da autenticação de grupos, como o ataque de repetição, man-in-the-middle e personificação, e suas respectivas formas de defesa.

Em seguida, são descritos os mecanismos de segurança utilizados nos dois protocolos de autenticação propostos, ECDH, emparelhamento bilinear, segredo de Shamir e interpolação de Lagrange. Por fim, são apresentadas uma descrição das redes 3GPP LTE/LTE-A, seus principais componentes e ainda uma breve introdução ao AVISPA, a ferramenta utilizada na simulação da segurança dos protocolos propostos.

## **3. A LOW COST GROUP AUTHENTICATION PROTOCOL FOR THE INTERNET OF THINGS**

Este capítulo é destinado a apresentação do primeiro protocolo proposto, baseado em criptografia assimétrica e que tem o objetivo de autenticar grupos de dispositivos MTC na rede 3GPP LTE de maneira segura e eficiente, cumprindo com os requisitos da MTC e apresentando resultados superiores aos de outros protocolos já propostos na literatura. Para isto, primeiro são apresentados os protocolos de referência que mais influenciaram na concepção desta proposta, [9], [12] e [14]. Em seguida, o protocolo proposto é apresentado, baseado em ECDH, e em emparelhamento bilinear, com gerenciamento de grupo realizado através de árvore binária. Depois, são feitas análises de segurança e de performance e as respectivas comparações com os protocolos descritos no início do capítulo. Por fim, o protocolo é validado formalmente utilizando a ferramenta AVISPA.

## **4. AUTHENTICATION AND KEY AGREEMENT PROTOCOL BASED ON SECRET SHARING FOR MACHINE TYPE COMMUNICATIONS**

Este capítulo é destinado a apresentação do segundo protocolo proposto, baseado em criptografia simétrica, com os mesmos objetivos do protocolo proposto no capítulo 3, porém com custos muito mais reduzidos, fato ocasionado pelo uso de criptografia simétrica. Primeiro são apresentados alguns protocolos de referência que influenciaram fortemente a composição da proposta, [11], [13],[15] e [16]. Em seguida, o protocolo proposto é apresentado, baseado no segredo de Shamir e no protocolo de autenticação para grupos proposto por [16]. Assim como no capítulo 3, também são feitas análises de segurança e de performance, além de comparações com [11], [13] e [15] e da verificação formal feita na ferramenta AVISPA.

## **5. CONCLUSÃO**

É muito elevada a importância do presente trabalho, pois seu caráter interdisciplinar aborda redes sem fio, segurança da informação, Internet das coisas (IoT), qualidade de serviço (QoS) e validação formal de protocolos. Além disso, aborda um aspecto essencial na concepção da IoT nos próximos anos, através do estabelecimento da MTC sobre a rede LTE/LTE-A. A autenticação de grupos permitirá o acesso à rede para bilhões de dispositivos, de maneira altamente segura e eficiente, revolucionando o cenário tecnológico como nunca visto antes.

Os dois protocolos de autenticação de grupos desenvolvidos se apresentaram bastante eficientes e seguros, inclusive quando comparados com outros protocolos presentes na literatura. O primeiro protocolo se baseou no uso de criptografia assimétrica, em ECDH e em emparelhamento bilinear. Enquanto o segundo protocolo foi baseado em criptografia simétrica, no segredo de Shamir e no protocolo proposto por [16].

Ambos os protocolos propostos cumprem com os requisitos de segurança da MTC e apresentam todas as propriedades de segurança e resistência a ataques descritas nas referências adotadas. Além disso, ambos obtiveram resultados satisfatórios em suas performances computacional e de comunicação, mesmo que apresentando uma certa desvantagem em alguns aspectos em relação a alguns dos outros protocolos analisados.

***Palavras-chave – Chave de Sessão, Autenticação e Acordo de chaves (AKA), Comunicações tipo máquina (MTC), AVISPA.***

---

## ABSTRACT

The objective of this work is to provide an overview on group authentication protocols for Internet of Things (IoT) and to propose two new group protocols. Both protocols perform authentication and key agreement among a group of devices and a Mobility Management Entity (MME) and aim performance improvements, ensuring a robust security and anonymity protection. One scheme is based on both Elliptical Curves Diffie-Hellman protocol and bilinear pairing and the other is a lightweight symmetric protocol based on Shamir's secret. Additionally, both protocols have their performance and security objectives accomplishment analyzed and compared with other works already proposed in the literature. The performance analysis and comparison comprises communication, computational, verification and storage costs. Some of the security features analyzed are forward/backward secrecy (FS/BS), anonymity and resistance to several attacks. Finally, the protocols were formally validated by AVISPA tool.

***Keywords – Session Key, Authentication and Key Agreement (AKA), Machine Type Communication (MTC), AVISPA.***



# TABLE OF CONTENTS

<b>Chapter 1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	CONTEXT AND MOTIVATIONS .....	1
1.2	GROUP AUTHENTICATION PROBLEM .....	2
1.3	OBJECTIVES .....	2
1.4	WORK ORGANIZATION .....	2
<b>Chapter 2</b>	<b>THEORETICAL BACKGROUND.....</b>	<b>3</b>
2.1	SECURITY PROPERTIES .....	3
2.2	SECURITY ATTACKS AND DEFENSES .....	4
2.3	ELLIPTIC CURVE DIFFIE-HELLMAN PROTOCOL (ECDH) .....	6
2.4	BILINEAR PAIRING.....	6
2.5	SHAMIR'S SECRET .....	7
2.6	LAGRANGE INTERPOLATION FORMULA.....	8
2.7	LTE/LTE-A NETWORK.....	8
2.8	AVISPA TOOL.....	10
<b>Chapter 3</b>	<b>A LOW COST GROUP AUTHENTICATION PROTOCOL FOR THE INTERNET OF THINGS.....</b>	<b>11</b>
3.1	INTRODUCTION.....	11
3.2	RELATED WORK.....	12
3.2.1	GROUP AUTHENTICATION PROTOCOL DEVELOPED BY CAO ET AL. [9].....	12
3.2.2	GROUP AUTHENTICATION PROTOCOL DEVELOPED BY FU ET AL. [12] .....	16
3.2.3	OTHER GROUP AUTHENTICATION PROTOCOLS .....	18
3.2.4	COMPARATIVE EVALUATION .....	18
3.3	PROPOSED PROTOCOL .....	19
3.3.1	INITIALIZATION PHASE.....	21
3.3.2	MUTUAL AUTHENTICATION AND KEY AGREEMENT PHASE.....	22
3.3.3	GROUP KEY UPDATE.....	25
3.4	SECURITY ANALYSIS OF THE PROPOSED PROTOCOL AUTHENTICATION PROCEDURE .....	25
3.5	PERFORMANCE EVALUATION AND COMPARISONS.....	27
3.5.1	COMPUTATIONAL COST.....	27
3.5.2	COMMUNICATION COST .....	29
3.5.3	VERIFICATION COST .....	31
3.5.4	STORAGE COST.....	32
3.6	PROPOSED PROTOCOL FORMAL VERIFICATION.....	33
3.6.1	PROTOCOL SIMULATION .....	33
3.6.2	SECURITY VERIFICATION RESULTS .....	36
3.7	CONCLUSION.....	38
<b>Chapter 4.....</b>	<b>AUTHENTICATION AND KEY AGREEMENT PROTOCOL BASED ON SECRET SHARING FOR MACHINE TYPE COMMUNICATIONS.....</b>	<b>39</b>
4.1	INTRODUCTION .....	39
4.2	RELATED WORK.....	40
4.2.1	SHAMIR'S SECRET BASED SCHEMES PROPOSED BY HARN [16].....	40
4.2.2	GROUP AUTHENTICATION PROTOCOL DEVELOPED BY LAI ET AL. [13].....	41
4.2.3	GROUP AUTHENTICATION PROTOCOL DEVELOPED BY LI ET AL. [15].....	43
4.2.4	OTHER GROUP AUTHENTICATION PROTOCOLS .....	46
4.2.5	COMPARATIVE EVALUATION .....	46
4.3	PROPOSED PROTOCOL .....	46
4.3.1	REGISTRATION PHASE.....	48
4.3.2	AUTHENTICATION AND KEY AGREEMENT PHASE.....	49
4.3.3	GROUP KEY UPDATE.....	53
4.3.4	GROUP SECRET UPDATE .....	53
4.4	SECURITY ANALYSIS .....	54
4.5	PERFORMANCE EVALUATION AND COMPARISONS .....	57
4.5.1	COMPUTATIONAL COST.....	57
4.5.2	COMMUNICATION COST .....	61
4.5.3	STORAGE COST.....	63
4.6	PROPOSED PROTOCOL FORMAL VERIFICATION.....	64
4.6.1	PROTOCOL SIMULATION .....	64

4.6.2 SECURITY VERIFICATION RESULTS .....	69
4.7 CONCLUSION.....	71
<b>Chapter 5 CONCLUSION .....</b>	<b>73</b>
<b>REFERENCES .....</b>	<b>75</b>



# LIST OF FIGURES

Figure 1 – 3GPP MTC adapted network architecture. ....	1
Figure 2 – Passive attack (source: [2]). ....	4
Figure 3 - Active attack (source: [2]). ....	4
Figure 4 -Example of MitM attack in shared key generation (source: [2]). ....	5
Figure 5 - Elliptic Curve Diffie Hellman protocol. ....	6
Figure 6 – Architecture SAE of LTE/LTE-A network (source: [25]). ....	9
Figure 7 –Register phase in Cao et al. [9] protocol.....	14
Figure 8– Group-based Access Authentication phase in Cao et al. [9] protocol.....	15
Figure 9 – Mutual authentication phase in Fu et al. protocol (source: [12]). ....	18
Figure 10 – Network architecture of the proposed protocol.....	21
Figure 11 – Binary tree presented for group organization (source: [11]). ....	21
Figure 12 – The authentication phase of the proposed protocol.....	24
Figure 13 – The binary tree after the entrance of MME (source [11]). ....	24
Figure 14 – Comparison of computational cost. ....	28
Figure 15 – Comparison of communication cost. ....	30
Figure 16 – Comparison of improvement rate for communication cost.....	31
Figure 17 – Comparison of verification cost.....	32
Figure 18 – Role of each MTCN in HLSPL.....	33
Figure 19 – Role of MTCN <sub>leader</sub> in HLSPL. ....	34
Figure 20 – Role of the MME in HLSPL.....	34
Figure 21 – Role of the HSS in HLSPL .....	35
Figure 22 – Role specification for the session and environment in HLSP.....	35
Figure 23 – Security goals established in HLSP. ....	36
Figure 24 – Security simulation results for OFMC.....	36
Figure 25 – Security simulation results for CL-AtSe.....	37
Figure 26 – Protocol’s message exchange in SPAN.....	37
Figure 27 – Intruder’s simulation in SPAN.....	38
Figure 28 –Shamir’s Secret in Harn scheme (source: [16]). ....	40
Figure 29 – (t,m,n) GAS in Harn scheme (source: [16]). ....	41
Figure 30 – Mutual authentication phase in Lai et al (source: [13]). ....	43
Figure 31 – Group-based Authentication and Agreement phase in Li et al. (source: [15]). ....	45
Figure 32 – The network architecture adopted.....	47
Figure 33 – Binary tree presented for group organization (source: [11]). ....	49
Figure 34 – The authentication and key agreement phase of the proposed protocol. ....	52
Figure 35 – The binary tree after the entrance of MME (source [11]). ....	52
Figure 36 – Comparison of computational cost based on first analysis. ....	59
Figure 37 – Comparison of computational cost based on second analysis. ....	60
Figure 38 – Comparison of communication cost. ....	62
Figure 39 – Comparison of improvement rate for communication cost.....	63
Figure 40 - Role of each MTCN in HLSPL. ....	64
Figure 41 - Role of MTCN <sub>leader</sub> in HLSPL.....	65
Figure 42 – Role of the MME in HLSPL .....	66
Figure 43 – Role of the HSS in HLSPL .....	67
Figure 44 – Role specification for the session and environment in HLSP.....	68
Figure 45 – Security goals intended in HLSP. ....	69
Figure 46 - Security simulation results for OFMC.....	69
Figure 47 - Security simulation results for CL-AtSe.....	70
Figure 48 – Protocol’s message exchange in SPAN.....	70
Figure 49 - Intruder’s simulation in SPAN.....	71

# LIST OF TABLES

Table 1 - Main entities involved in the architecture of Cao et al. [9] protocol .....	12
Table 2 — Notations used by Cao et al. [9].....	13
Table 3 – Main entities involved in the architecture of Fu et al. [12] protocol.....	16
Table 4 - Notations used by Fu et al. [12].....	16
Table 5 - Described protocols’ comparative table.....	19
Table 6 - Main entities involved in the architecture of the proposed protocol.....	19
Table 7 - Notations used in the proposed protocol.....	20
Table 8 - Comparison of security objectives between protocols.....	26
Table 9 - Time cost of each operation considered in <i>ms</i> . .....	27
Table 10 – Computational cost comparison between protocols in authentication phase .....	28
Table 11 – Communication cost of each parameter transmitted. ....	29
Table 12 - Communication cost in bits by message and in total. ....	29
Table 13 – Verification cost in bits by message and in total.....	32
Table 14 - Storage cost in bits by entity. ....	32
Table 15 – Main entities involved in the architecture of Lai et al. [13] protocol.....	41
Table 16 - Notations used by Lai et al. [13].....	42
Table 17 - Main entities involved in the architecture of Li et al. [15] protocol .....	44
Table 18 - Notations used by Li et al. [15].....	44
Table 19 – Comparative evaluation of the described protocol. ....	46
Table 20 - The main entities involved in the proposed protocol architecture .....	47
Table 21 - Notations used in the proposed protocol.....	48
Table 22 – Comparison of security objectives between protocols .....	57
Table 23 – Time cost of each operation considered in <i>ms</i> .....	57
Table 24 – Computation cost comparison between protocols (First method).....	58
Table 25 – Computation cost comparison between protocols (Second method).....	60
Table 26 – Communication cost of each parameter transmitted .....	61
Table 27 – Communication cost in bits by message and in total. ....	61
Table 28 – Storage cost in bits by entity. ....	63

# SYMBOLS AND ACRONYMS

$p$	Large Prime Number
$Z_p$	A Finite Field of Size $p$
$G$	Elliptic Curve Group
$P$	Generator of Group $G$
$e(-,-)$	Bilinear Pairing Function
$h(.)$	Hash Function
$\parallel$	Concatenation Operation
$\oplus$	XOR Operation
$*$	Point Multiplicative Operation
	Secure Channel
	Insecure Channel

## Abbreviations

3GPP	3 <sup>o</sup> Generation Partnership Project
AKA	Authentication and Key Agreement
AVISPA	Automated Validation of Internet Security Protocols and Applications
BS	Backward Secrecy
CL-AtSe	Constraint Logic-based ATtack Searcher
DoS	Denial of Service
GKi	Group Key of $i$
HLPSL	High Level Protocol Specification Language
HSS	Home Subscriber Server
IoT	Internet of Things
IR	Improvement Rate
KGC	Key Generation Center
LC	Lagrange Component
LTE	Long Term Evolution
MAC	Message Authentication Code
MitM	Man-in-the-Middle
MME	Mobility Management Entity
MTC	Machine Type Communication
MTCDi-j	Mobile Terminal Communication Device
OFMC	Open-source Fixed-point Model Checker
FS	Forward Secrecy
SPAN	Security Protocol Animator for AVISPA
XOR	Exclusive or operation
XRES	Expected Authentication Response

*This chapter presents the main context addressed in this work, introducing a global view of group authentication problem and presenting the motivation of this work. It also provides a brief description of the paper's organization.*

## 1.1 CONTEXT AND MOTIVATIONS

The Internet of Things (IoT) is expected to trend in the next following years, bringing new applications to almost every task of our daily routine. It will considerably revolutionize the way we live and interact with things, through the connection of everything it can be imagined to the Internet. Some important applications are Smart grid that will provide intelligent energy metering and monitoring, intelligent cities that predict the integration and connection of residences, services and transportation inside urban centers and the Mobile healthcare (M-health) that expect to provide health information collected by sensors to treatment centers.

All these technologies can be classified as Machine Type Communication (MTC). The 3rd Generation Partnership Project (3GPP) has a very well implemented system architecture that is a strong candidate to receive the MTC traffic. The 3GPP MTC simplified adapted architecture is presented on Figure 1. It consists of the following: The Evolved Universal Terrestrial Access Network (E-UTRAN), which is composed by groups of Mobile Terminal Communication Device (MTCD) and groups of EnodeB (eNB); The Evolved Packet Circuit (EPC), which is composed of a Mobility Management Entity (MME) and a Home Subscriber Server (HSS); The MTC-Server, responsible to manage the MTCDs activities and information; the MTC-Users, that can access the devices collected information and data.

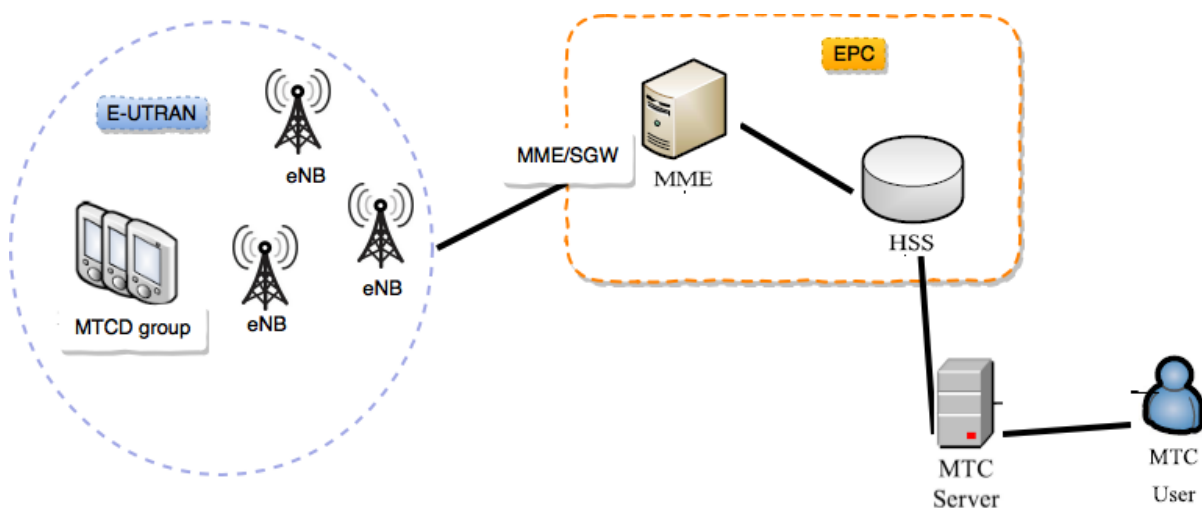


Figure 1 – 3GPP MTC adapted network architecture.

According with the 3GPP [1], its architecture supports roaming scenarios, where the MTCD must only have a connection with the HSS of the home network and with the MME of the server network. The authentication and key agreement protocol proposed by 3GPP to Long Term Evolution (LTE) is the Evolved Packet System – Authentication and Key Agreement (EPS-AKA) and it authenticates individually each device arriving in a network. In a scenario with billions of devices, a single device based authentication procedure may cause several problems, as signaling congestion and security vulnerabilities.

## **1.2 GROUP AUTHENTICATION PROBLEM**

Machine type communication (MTC) applications predict the generation of large amounts of data, derived from billions of devices in the next years. The standard protocols currently adopted by 3GPP are not adapted to huge groups of devices and their use to these types of applications have been proven to cause security losses as lack of integrity, availability, confidentiality and vulnerability to several attacks. Additionally, it is important to emphasize that these devices are mainly resource constrained, what generates the necessity of lightweight protocols, with reduced consumption of computational and communication resources. Consequently, the exponential growth of connected devices and data exchange require special authentication methods, in order to prevent security failures, improve the system performance and provide all the MTC requisites.

## **1.3 OBJECTIVES**

The objective of this work is to propose two different group-based authentication and key agreement protocols for IoT, one based on ECDH protocol and bilinear pairing and the other based on Shamir's secret respectively. Both protocols are expected to securely and efficiently perform authentication between a group of devices and MME, reducing computational and communication costs. Consequently, consuming less resources and improving their performances and security goals.

## **1.4 WORK ORGANIZATION**

This section describes the organization of the work in the following chapters. Chapter 2 presents the theoretical background used to build this work's main ideas and the information necessary to understand it. Security concepts and properties are described, followed by ECDH protocol, bilinear pairing, Shamir's secret, Lagrange's interpolation and LTE network descriptions. The chapter is finished with a brief description of the AVISPA tool. Chapter 3 presents related protocols for group authentication that use ECDH and a brief description of some of them, followed by the first proposed protocol, which is based on ECDH and bilinear pairing. Chapter 4 presents related protocols that use symmetric cryptography and a brief description of some of them, followed by the second proposed protocol, using symmetric cryptography and Shamir's secret. Finally, chapter 5 presents the conclusion.

# Chapter 2

# THEORETICAL BACKGROUND

*This chapter provides all the important security concepts and properties background necessary to understand the protocols proposed in this work.*

## 2.1 SECURITY PROPERTIES

As presented in Stallings [2], confidentiality, integrity and availability are the three key objectives considered the foundation of a system's security. Together, they are referred the CIA triad.

**Confidentiality:** Is the control of which are the information disclosed and available to authorized individuals.

**Integrity:** "Guarding against improper information modification or destruction, including ensuring information nonrepudiation and authenticity. A loss of integrity is the unauthorized modification or destruction of information".

**Availability:** Guarantees that the system is operating to all authorized users when necessary.

Other three important concepts in the system's security scenario and they are named the AAA:

**Authentication:** Assures a communication is authentic by guaranteeing to a receiver that a message is from it claims to be from.

**Authorization:** "Granting access to specific services and/or resources based on the authentication."

**Accounting:** Assures the control and register of the actions executed in the system by authorized individuals.

In addition, there are other important security concepts and they are described below:

**Nonrepudiation:** Assures that an entity involved in a communication cannot deny its participation in the process.

**Privacy (Anonymity):** Assures that an entity's real identity is not a public information, guaranteeing it is untraceable to bad intentioned individuals.

**Backward and Forward Secrecy (BS/FS):** As described by Cremers [23], it is the secrecy of previous and subsequent information. In cryptography, this information is a secret key, where the information is protected by the use of these keys. Anyone who has the right key can access the information. Then, the backward and forward secrecy are the security properties that guarantee the protection of previous and subsequent keys, even if the current key is discovered. BS/FS have two security levels:

*Weak Backward and Forward Secrecy (WBS/WFS):* Guarantees BS/FS to the case of an attacker discovering a key, but did not participate actively of the key agreement process, meaning that it do not know how to generate the respective key. However, if the attacker was directly involved in the key agreement processes, the BS/FS is not guaranteed.

*Strong Backward and Forward Secrecy:* The highest level of backward and forward secrecy guarantees that even though an attacker participates actively in the key agreement procedure and can generate the current keys, it is not capable to discover previous or subsequent keys.



## 2.2 SECURITY ATTACKS AND DEFENSES

Stallings [2] classifies the security attacks in passive and active. A passive attack is used to learn transmitted information and to create parameters of the system by eavesdropping the communication channel. This type of attack cannot cause damages to the system because it does not modify the system's information. In contrast, an active attack is capable to affect resources and the operation of the system, because it executes modifications on the information exchanged. Passive and active attacks are seen on Figure 2 and 3 respectively.

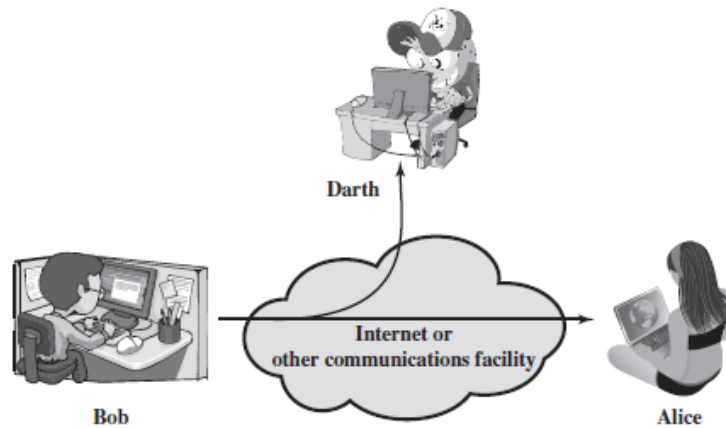


Figure 2 – Passive attack (source: [2]).

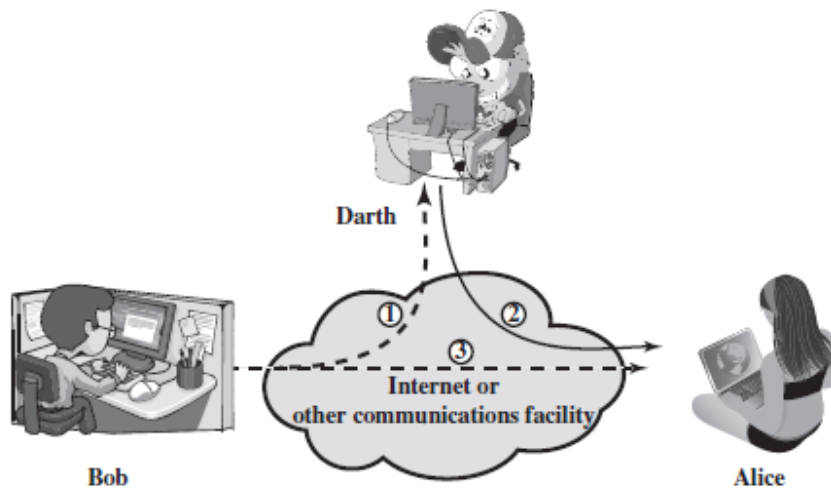


Figure 3 - Active attack (source: [2]).

The MTC most common attacks are man-in-the-middle, replay, denial of service, redirecting and impersonation. A brief description and an explanation of how to avoid them are presented below:

**Man-in-the-Middle (MitM) Attack:** Occur when an intruder joins the communication channel as a third entity, so it can eavesdrop the messages to obtain all the information passing through it. In the example using Alice and Bob, an intruder must trick Alice to believe it is Bob and trick Bob to believe it is Alice. Then, all the messages that Alice sends to Bob passes through the intruder. The same happens to the messages that Bob sends to Alice. The intruder forward the messages to the original destination, making this attack very difficult to detect, and has access to all the information exchanged. Figure 4 presents an example of MitM attack in a shared key agreement.

*Solution:* This attack can be avoided in the authentication procedure by using pre-shared keys that are not transmitted in plaintext through the channel in the confection of the authentication parameters and shared key, so the attacker cannot forge them. Another solution is the use of timestamps and sequence numbers that could only be generated by the destination entity, so the intruder would fail forging them.

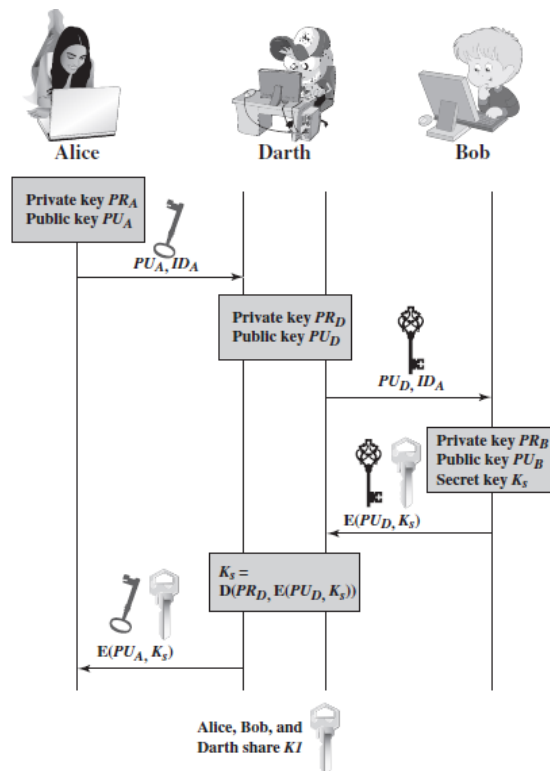


Figure 4 -Example of MitM attack in shared key generation (source: [2]).

**Replay Attack:** Occur when an intruder eavesdrops the communication channel to obtain important parameters and use them to impersonate one of the entities involved in the subsequent process executions.

*Solution:* Produce different and fresh parameters to each process execution that will expire after it ends. If an intruder obtains a system parameter, the attack fails because the parameter will already be expired in the next process execution.

**Denial of Service Attack (DoS):** Occur when an authorized individual cannot access a service due its unavailability. An attacker can induce an entity to suppress messages destined to a particular destination or interrupt the entire service by disabling or overloading a network with a large quantity of messages. According to Khan et al. [4], in mobile devices, this attack is related to their limited resources capabilities, limited hardware resources for example, and maybe one attacker would be enough to successful execute it. In the authentication scenario, an attacker can cause the DoS attack by sending large amounts of invalid authentication parameters.

*Solution:* The solution to this attack is simple and involves the inclusion of a verification parameter in the message that precedes the authentication procedure. This verification parameter can be a timestamp or a sequence number and its validity is verified before the authentication starts. Then, if an attacker uses an invalid timestamp or sequence number, the entire procedure is interrupted in time to prevent the DoS attack.

**Redirecting Attack:** According to Zhang et al. [5] it occurs when an attacker in possession of a false base station (BS) can use it to impersonate a genuine BS and receive all the traffic destined to it.

*Solution:* The introduction of a BS verification parameter to provide the chance of confirming its origin. This parameter can be the Location Area Identification (LAI) that is unique to each BS.

**Impersonation Attack:** Occur when a false device succeeds pretending it is genuine and receive the messages destined to this genuine device.

*Solution:* Use of pre-shared secrets in the creation of the authentication parameters used to verification that are not transmitted in plaintext through the channel. Thus, only legit device would be able to generate valid authentication parameters.

## 2.3 ELLIPTIC CURVE DIFFIE-HELLMAN PROTOCOL (ECDH)

Stallings [2] affirms that “elliptic curve cryptography, compared to the RSA, offers equal security for a far smaller key size, thereby reducing processing overhead”. When combined with the Diffie-Hellman problem, it trusts its success in the difficulty of discrete logarithm problem. The Elliptic Curve Diffie-Hellman protocol (ECDH) is characterized as the generation of a secret shared among two or more individuals, which can be used as a shared key, based on information about each of them. Lai et al. [14] describes the ECDH using an example, as follows:

- First, some system parameters are set: a large finite prime number  $p$ , an elliptic curve  $E$  over a large finite field  $F_p$ , and a point  $P$  on that curve, which is a public value.
- The example considers two entities wanting to communicate, named Alice and Bob. Next, both entities choose a random number,  $R_a$  to Alice and  $R_b$  to Bob, and execute a multiplication over the elliptic curve  $R_aP$  and  $R_bP$ . Then, they proceed as presented in Figure 5:
  1. Alice sends  $R_aP$  to Bob and Bob sends  $R_bP$  to Alice. The secrecy of  $R_a$  and  $R_b$  is trusted on the discrete logarithm problem that consists on the difficulty of an attacker discovering  $R_a$  or  $R_b$  if it knows  $R_aP$ ,  $R_bP$  and  $P$ .
  2. Then, each of them calculates  $R_aR_bP$ , which is the secret shared among them. Finally, they can send data to each other, using the secret to encrypt the messages.

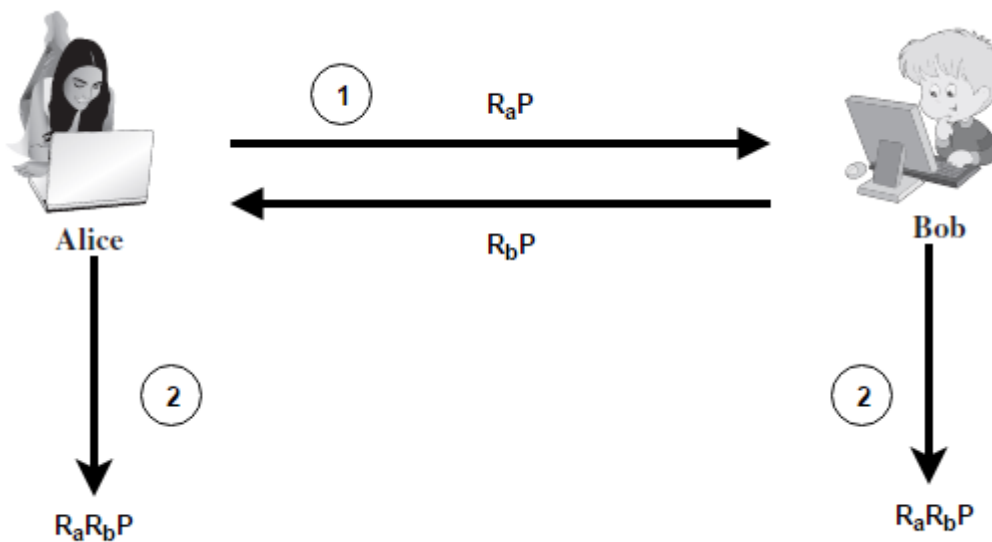


Figure 5 - Elliptic Curve Diffie Hellman protocol.

Besides the protocol effectiveness, it is vulnerable to Man-in-the-Middle attack because an intruder may be positioned between Alice and Bob. In this work, the ECDH was chosen because of the security it can provide, based on the discrete logarithm problem. It is used in the session key generation, as part of the secret shared among the devices and the MME. However, the session key is not vulnerable to Man-in-the-Middle attack because it is also composed of other parameters, not exchanged in the communication channel.

## 2.4 BILINEAR PAIRING

The bilinear pairing operation is widely employed in cryptographic applications that require stronger security mechanisms. According to Dutta et al. [29], it is also applied to signatures verification, key agreement, signcryption, threshold decryption, key sharing, identification and many other applications. It is used in this work because it can provide efficient and secure verification. The verification is

performed by the server network in the authentication procedure of the protocol proposed in chapter 3, when the devices' aggregated information is verified, authenticating them simultaneously.

Menezes [3] presents in its work an introduction to pairing-based cryptography that is described as follows. Considering a prime number  $p$ ,  $G_1$  an additive group and  $G_T$  a multiplicative group of order  $p$ . "A bilinear pairing on  $(G_1, G_T)$  is a map:

$$\hat{e}: G_1 \times G_1 \rightarrow G_T \quad (2.1)$$

That satisfies the following conditions:

1. **Bilinearity:** For all  $R, S, T \in G_1$ ,  $\hat{e}(R+S, T) = \hat{e}(R, T)\hat{e}(S, T)$  and  $\hat{e}(R, S+T) = \hat{e}(R, S)\hat{e}(R, T)$ .
2. **Non-degeneracy:**  $\hat{e}(P, P) \neq 1$ .
3. **Computability:**  $\hat{e}$  can be efficiently computed."

Menezes [3] also describes some bilinear pairing properties, as presented below:

$$1. \hat{e}(S, \infty) = 1 \text{ and } \hat{e}(\infty, S) = 1. \quad (2.2)$$

$$2. \hat{e}(S, -T) = \hat{e}(-S, T) = \hat{e}(S, T)^{-1}. \quad (2.3)$$

$$3. \hat{e}(aS, bT) = \hat{e}(S, T)^{ab} \text{ for all } a, b \in \mathbb{Z}. \quad (2.4)$$

$$4. \hat{e}(S, T) = \hat{e}(T, S). \quad (2.5)$$

$$5. \text{ If } \hat{e}(S, R) = 1 \text{ for all } R \in G_1, \text{ then } S = \infty. \quad (2.6)$$

Dutta et al. [29] is a survey that presents bilinear pairing applications, with several detailed examples. Two of them are described below due their relevance to this work. They are the Computational Diffie-Hellman Problem and the Aggregate Signature scheme.

1. The Computational Diffie-Hellman (CDH) problem combines ECDH with the implementation of bilinear pairing. Considering the bilinear pairing properties described previously, the same environment and entities from the example presented in section 2.3 (Alice and Bob), their respective random values  $RaP$  and  $RbP$ , and a given  $P$  it is possible to compute:

$$\hat{e}(S, T)^{RaRbP} \quad (2.7)$$

Resulting in the output  $RaRbP$ , a secret shared among Alice and Bob.

2. The Aggregating Signature scheme considers  $n$  signatures on  $n$  messages  $mi$ , where  $1 \leq i \leq n$ , that are aggregated in a single short signature. The single signature is verified to confirm that each of the users signed its respective message  $mi$ .

Considering the bilinear pairing properties described previously, a public key  $vi = g^{ri}$ , the messages  $mi$ , an aggregated signature  $\sigma = H(mi)^{ri}$ , where  $1 \leq i \leq n$ , and  $H$ , a map to point hash function, it is possible to verify if:

$$e(g, \sigma) = \prod e(vi, H(mi)) \quad (2.8)$$

## 2.5 SHAMIR'S SECRET

In 1979, Adi Shamir proposed a scheme, Shamir [22], for cryptography systems based on sharing secret, which enables the reconstruction of a parameter from a set of share secrets. Shamir defined a  $(k, n)$  *threshold scheme* [22], where a secret  $D$  is divided into  $n$  pieces  $D_1, D_2, \dots, D_n$ , and only with at least  $k$  pieces the secret  $D$  can be rebuild. The Shamir's scheme is presented with more details below:

**(k,n) Threshold Scheme** [22]: First, a polynomial function  $f(x)$  with degree  $k-1$  is defined:

$$f(x) = a_0 + a_1x + \dots + a_{k-1}x^{k-1} \quad (2.9)$$

Where the term  $a_0$  is defined as the secret  $D$ . Next, the  $n$  pieces are defined as  $Di = f(i)$ , where  $i = 1, \dots, n$ . Then, the scheme guarantees that with any subset of  $k$  pieces  $Di$  it is possible to recover the secret  $D$ ,

through a polynomial interpolation. Shamir also say that his scheme is safe until  $k-1$  pieces are revealed. In other words, it is only unsafe if all the pieces are revealed.

With this scheme, Shamir brought a way to share a secret, divided into several pieces and only possible to be recovered if all the reunited pieces are legitimate. The Shamir's secret is a dynamic, flexible and secure form to verify if a set of shares are legitimate. It is used in many areas nowadays, such image compression, cryptography algorithms and authentication protocols, which is the main subject of this work, enabling a scheme where a set of credentials is shared among the members of a group of devices and providing them authentication based on the knowledge of this credential.

The Shamir's secret is a good way to obtain group authentication and to group members authenticate themselves. It is useful because if all members authenticate themselves, then, the group is considered authentic and all members are trustful by just verifying if the secret generated is the same as the original.

The scheme of Shamir has the advantage of being a way of fast and dynamic authentication, because every member checks the secret and sends the information necessary to secret generation to other members. Then, in one verification, it is possible to authenticate all the group members. The disadvantage of this scheme is the computational costs that is not so high, but have other operations with lower cost. Other disadvantage is the impossibility of discovering which member is an attacker, in case of authentication failure, because all members are authenticated at the same time, using a specific defined secret. Therefore, it is impossible to divide the devices in subgroups and to verify which is the attacker.

## 2.6 LAGRANGE INTERPOLATION FORMULA

As described by Jeffreys et al. [21], in 1795, Joseph Louis Lagrange published the polynomial interpolation or "Lagrange interpolation polynomial", which is a formula that makes possible to rebuild an approximate polynomial function, through a set of points belonging to this function, where the many are the points applied to the formula, the closer to the recovery of the polynomial.

Considering  $n$  points  $(x_1, f(x_1)), \dots, (x_n, f(x_n))$  of a polynomial function  $P(x)$  with degree  $n-1$ , it is possible to approximate the function  $P(x)$  using  $n$  points as follows:

$$P'(x) = \sum_{j=1}^n f(x_j) \prod_{k=1; k \neq j}^n \frac{x - x_k}{x_j - x_k} \quad (2.10)$$

The larger  $n$  is, the more approximated  $P'(x)$  is to  $P(x)$ . However, large  $n$  gets the calculation more complex. The Lagrange interpolation formula is very mathematical, which makes possible to recover functions approximated to some points of this function. In addition, this characteristic of rebuilding brings some applications to this formula. It is largely used to rebuild compressed files, where a file can be compressed until the minimum possible size and recovered with Lagrange component. Other usual application is to recover damaged files, where with some samples is possible to recover an approximated version of the original file.

## 2.7 LTE/LTE-A NETWORK

Holma et al. [25] affirms that LTE/LTE-A is a mobile communication standard created by 3GPP to support the fourth generation of wireless mobile communication, named 4G. LTE is considered the evolution of GSM, HSPA, CDMA and WiMAX systems with the proposal to improve performance and offer higher data rates, lower latency, higher support to mobility, better handover performance, higher spectral efficiency and other improvements. The LTE-A network was launched in 2011.

LTE brought an evolution in the architecture to mobile communication networks, which is named SAE (System Architecture Evolution). Holma et al [25] says that SAE decreases the number of nodes between user and the core network, improving the performance and reducing the costs of network. It also enables an integration with other technologies or architectures used in mobile communication. SAE is divided

into two parts: the user plan, which is the Radio Access Network (RAN) and the core plan, which is the Evolved Packet Core (EPC). Figure 6 shows LTE network's architecture:

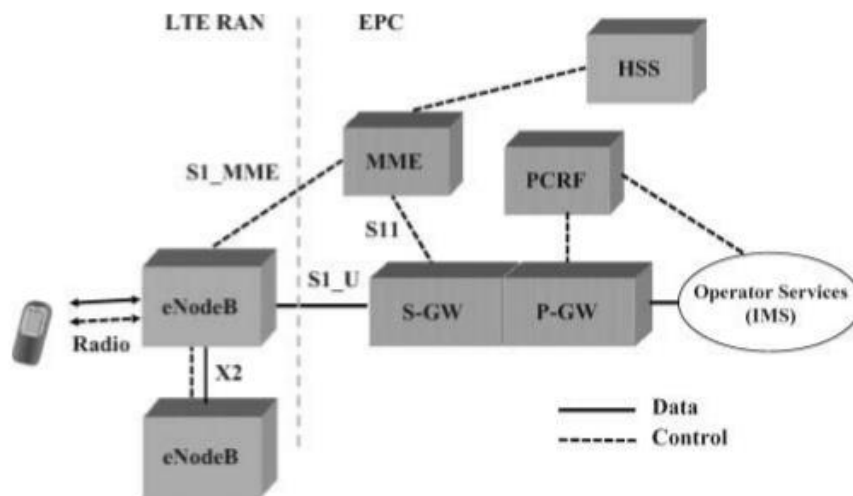


Figure 6 – Architecture SAE of LTE/LTE-A network (source: [25]).

According to [25], LTE RAN or E-UTRAN is responsible of the control plane of users and the management of radio resources, the E-UTRAN is simplified in a single element that is the eNodeB (evolved NodeB). The eNodeB manages the user plane and establishes the connection between user and network through the S-GW (SAE Gateway).

EPC is the core of LTE network and is the responsible of the connection with external networks, storage and controlling the subscriber information, signaling control, management and other functions. The EPC is compound of several entities with specific functionalities. The main entities are showed in Figure 6, as presented in [25], and described below:

- MME (Mobility Management Entity): Responsible to authenticate the user equipment (UE) that is trying to connect with the network through an eNodeB. The MME gets the subscriber information from home network, adds it to the information that UE sent and authenticates the user, determining the parameters of connection. In addition, MME is also responsible to the mobility management of users, verifying the coverage of each user and providing handover information to eNodeBs.
- S-GW (Serving Gateway): It is the gateway between the eNodeB and the core network, and acts according with the instructions of MME.
- P-GW (Packet Data Network Gateway): It is the gateway between the EPC and external networks, so when users tries to connect with external networks the P-GW performs the routing and filtering functions of these connections.
- PCRF (Policy and Charge Resource Function): Responsible for taxing users and QoS issues.
- HSS (Home Subscriber Server): It is the database of LTE network with the user and subscriber information, containing eNodeB's information. It provides information to MME in relation to UE authentication and authorization. It also has an Authentication Center (AuC).

Interfaces: All the entities in LTE architecture are connected through certain types of interfaces. Some of them are:

- LTE radio: It is an air interface between UE and eNodeB.
- S1 interface: It is responsible to connect the eNodeBs to EPC and it is divided into two types: S1\_U interface, which transmit issues related with user plane and connects the eNodeB with S-GW, and S1\_MME interface, which connects the eNodeB with MME and is related with control information.

- X2 interface: It connects the eNodeBs and used for control plane information between eNodeBs and to mobility information, like handover between eNodeBs.
- S11 interface: It is an interface between S-GW and MME.
- S6a interface: It is an interface between MME and HSS and it transfers authentication and subscription data for authenticating or authorizing user access.
- S5 interface: It is an interface between S-GW and P-GW and it serves like a tunnel to external connections.

Holma et al. [25] also says that:

- The LTE/LTE-A is based on OFDMA, in the downlink and on SC-FDMA in the uplink. It enables a spectrum flexibility and use of multiple subcarriers, resulting in high data rate.
- LTE uses MIMO to enhance the data rate taking advantage in propagation in multiple paths.
- The system enables 100Mbps peak data rate for high mobility and 1Gbps peak data rate for low mobility.
- Have a peak spectral efficiency of 15 bits/s/Hz.
- Have a bandwidth scalability to 40 MHz.
- Guarantee a latency to user plane of 10ms.
- Handover interruption in no more than 60ms (worst case).
- Supports other radio access systems and other radio frequency (RF) technologies.

## 2.8 AVISPA TOOL

Automated Validation of Internet Security Protocols and Applications (AVISPA) [6] is a formal verification tool that provides the simulation of Internet security-sensitive protocols. The High-level Protocol Specification Language (HLPSL) is the language used by AVISPA. According to Armando et al. [7] “it is an expressive, modular, role-based, formal language that allows for the specification of control flow patterns, data-structures, alternative intruder models, complex security properties, as well as different cryptographic primitives and their algebraic properties”. The HLPSL code of a protocol is divided into roles, one to each entity involved in the process and session and environment roles.

Each entity’s role describes its behavior during the protocol execution. All the parameters necessary to build the messages are generated and monitoring is added to the parameters that need to have their secrecy preserved. Then, the messages are sent to the recipient, which perform the generation of new parameters and follows with the message exchange.

The session role determines how the entities are related and how the sessions are established. One possible session has the participation of all the entities or may include an additional entity, the intruder. The environment role describes the environment of the protocol, presenting sessions’ composition and the intruder’s knowledge.

The tool used in this work has four back-ends, two of them were used to validate our proposed protocols, the **On-the-fly-Model-Checker (OFMC)** and the **Constraint-Logic-based Attack Searcher (CL-AtSe)**. The other two back-ends could not be used because they did not support some of the operations used in our protocols, as point multiplicative and XOR operations. The back-ends used return “SAFE” if no problems were detected, otherwise it returns “UNSAFE”, meaning that security properties were violated and the protocol is vulnerable to specified attacks.

# Chapter 3 A LOW COST GROUP AUTHENTICATION PROTOCOL FOR THE INTERNET OF THINGS

**Resumo:** Este capítulo apresentará uma breve descrição de alguns protocolos propostos para a autenticação de grupos de dispositivos MTC em LTE/LTE-A e a proposta de um novo protocolo de autenticação de grupos baseado no protocolo ECDH e em emparelhamento bilinear. Além disso, são apresentadas uma análise de segurança e comparações entre o protocolo proposto, o protocolo de referência 3GPP EPS-AKA e outros três protocolos estudados. Serão comparadas as propriedades de segurança e os custos computacionais, comunicação, armazenamento e verificação. Finalmente, é apresentada a validação formal do protocolo proposto utilizando o AVISPA, uma ferramenta que permite avaliar o atendimento a propriedades de segurança e objetivos de protocolos de autenticação. A avaliação da performance permite concluir que o protocolo proposto é mais vantajoso que as referências comparativas.

**Abstract:** This chapter briefly describes some protocols proposed for the authentication of MTC groups of devices in LTE/LTE-A and presents a new group authentication protocol based on both ECDH protocol and bilinear pairing. It also provides a security analysis and comparisons among the proposed protocol, the 3GPP EPS-AKA reference and other three protocols studied. The comparisons regard the security properties and the computation, communication, storage and verification costs. Finally, a formal validation of the protocol by AVISPA, a tool that evaluates the fulfillment of the security objectives of authentication protocols is presented. The performance evaluation allows to conclude that the proposed protocol is more advantageous than the comparative references.

## 3.1 INTRODUCTION

The Internet of Things (IoT) aims at the connection of billions of devices worldwide. It is directly linked to several applications, as Smart Grid, Vehicular Networks and Mobile Health (*m-Health*), expected to efficiently connect an assortment of types of device and concomitantly perform any type of application. One of the main challenges for the accomplishment of such a mass of connections is the secure and efficient authentication of the large number of devices involved the process.

A good solution is the aggregation of the devices into groups and their simultaneous authentication with the server network (*SN*). The current 3GPP standard EPS-AKA [1] authentication protocol is not suitable to large groups of devices, because it authenticates each device individually. In the IoT case, it can cause problems, such rise of computational and communication costs, and security vulnerabilities that might compromise both the communication and operation of the devices, mainly because many of them are resource-constrained.

Several group authentication protocols have been developed to decrease the communication and computational costs and provide the appropriate safety that MTC requires to flawlessly work. Some of them use the Elliptic Curves Diffie Hellman (*ECDH*)-based protocol to hold on Forward Secrecy and Backward Secrecy (*FS/BS*). Despite security advantages, ECDH-based protocols require higher computational costs in comparison with protocols based on symmetric cryptography. Cao et al. [9] use bilinear pairing in the authentication process, which can provide fast authentication to the devices, as all of them are authenticated with a single operation and reduce of the communication costs, mainly when associated with identity-based signatures.

The remainder of the chapter is organized as follows: Section 3.2 addresses some related work and the description of the work by Cao et al. [9] and Fu et al. [12]; Section 3.3 presents the proposed protocol with an initialization phase and mutual authentication and key agreement; Sections 3.4 and 3.5 report on the security analysis and the performance evaluation of the protocol, respectively; Section 3.6 presents the formal verification of the proposed protocol; Finally, Section 3.7 provides the conclusions.



## 3.2 RELATED WORK

Several protocols for group authentication have been proposed for MTC in LTE/LTE-A, since the increase of connected devices caused by IoT has been recognized. Among these protocols, we selected the following ones:

- Cao et al. [9] - GBAAM: Group-based Access Authentication for MTC in LTE Networks;
- Fu et al. [12] - A privacy-preserving group authentication protocol for machine-type communication in LTE/LTE-A networks;
- Lai et al. [14] - SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks.

The criteria adopted for papers' selection were the employment of asymmetric cryptography, challenge-response and ECDH, which is a robust way to preserve FS/BS. As well as good performance and the publication in periodicals or events of good quality. Additionally, [9] was selected for a more detailed analysis because it uses bilinear pairing to verify the devices and [12] and [14] were selected because they use pseudo identities to protect the permanent identity of the devices.

### 3.2.1 GROUP AUTHENTICATION PROTOCOL DEVELOPED BY CAO ET AL. [9]



Cao et al. [9] developed a group authentication protocol based on aggregated signatures. Its main objective is to perform a secure and efficient mutual authentication and a key agreement among groups of devices and a server MME. The protocol comprises two phases, namely register phase and group-based access authentication phase.

In the register phase all devices must prove authentic to the KGC to receive their private keys. A leader is elected to aggregate all the devices signatures. In the authentication phase the MME employs bilinear pairings to verify the aggregated signature of the MTCDS. In case of a verification failure, the MME divides the signature group into subgroups and remakes the bilinear pairing in each of them, until it detects the subgroup with an invalid signature. It performs successive divisions and repeats process until it finds the invalid signature. Despite its security robustness and simple verification process, the use of ECDH combined with the bilinear pairing increases computational costs, which may be a problem to resource-constrained devices. Table 1 presents the main entities involved in the authentication procedure and Table 2 provides all the notations described in Cao's paper.

Table 1 - Main entities involved in the architecture of Cao et al. [9] protocol

Abbreviation	Entity
<b>MTCDi</b>	Mobile Terminal Communication Device $j$ of group $i$
<b>MTCD<sub>leader</sub></b>	Mobile Terminal Communication Devices' group leader
<b>HSS</b>	Home Subscriber Server
<b>MME</b>	Mobile Management Entity

Table 2 — Notations used by Cao et al. [9]

Notation	Definition
$p$	A $k$ -bit prime
$Z_p$	A prime finite field
$G_1, G_2$	Two elliptic curve groups
$P$	Generator for group $G_1$
$e(-,-)$	Bilinear pairing function $e: G_1 \times G_1 \longrightarrow G_2$
$h_1(\cdot)$	A hash function $h_1: \{0,1\}^* \longrightarrow G_1$
$h_2(\cdot)$	A hash function $h_2: \{0,1\}^* \longrightarrow Z_p$
$h_3(\cdot)$	A hash function $h_3: \{0,1\}^* \times G_1 \longrightarrow Z_p$
$T_{exp_i}/ T_i$	$i$ 's expiration time/ current time
$ID_i$	Identity of node $i$
$x/PK$	Private/Public key of KGC ( $x \in Z_p^*$ ) ( $PK = xP$ )
$GID$	MTC group's identity
$SID_i$	$i$ 's private long-term key generated by KGC
$(S_{MME}, R_{MME})$	MME's private long-term key
$SK_{MMEi}$	Session key between MME and MTCD $i$
$\parallel$	Concatenation operation
$*$	Elliptic curve scalar multiplication operation
	Secure channel
	Insecure channel

Its architecture is like the 3GPP LTE, with the addition of an MTC Server that can be located inside or outside the LTE architecture. KGC can be integrated with the HSS. The channel between MME and HSS is secure. The two phases of the protocol are described below.

**Register phase:** According to Cao et al., [9] this phase is executed only once. Its main objective is to share secret parameters and keys among the entities to be used in the authentication phase. A group of MTCDs is created based on some common characteristics, as location. The group receives its GID and elects an MTCD leader according to communication capability, communication link quality, and storage and battery status. Each MTCD and MME contact the KGC through a adapted version of EPS-AKA protocol to obtain their private key. The KGC generates and publishes the system parameters  $\{p, G_1, G_2, e, P, PK, h_1, h_2, h_3\}$  and keeps the master key  $x$  secret. The register phase is detailed in Figure 7.

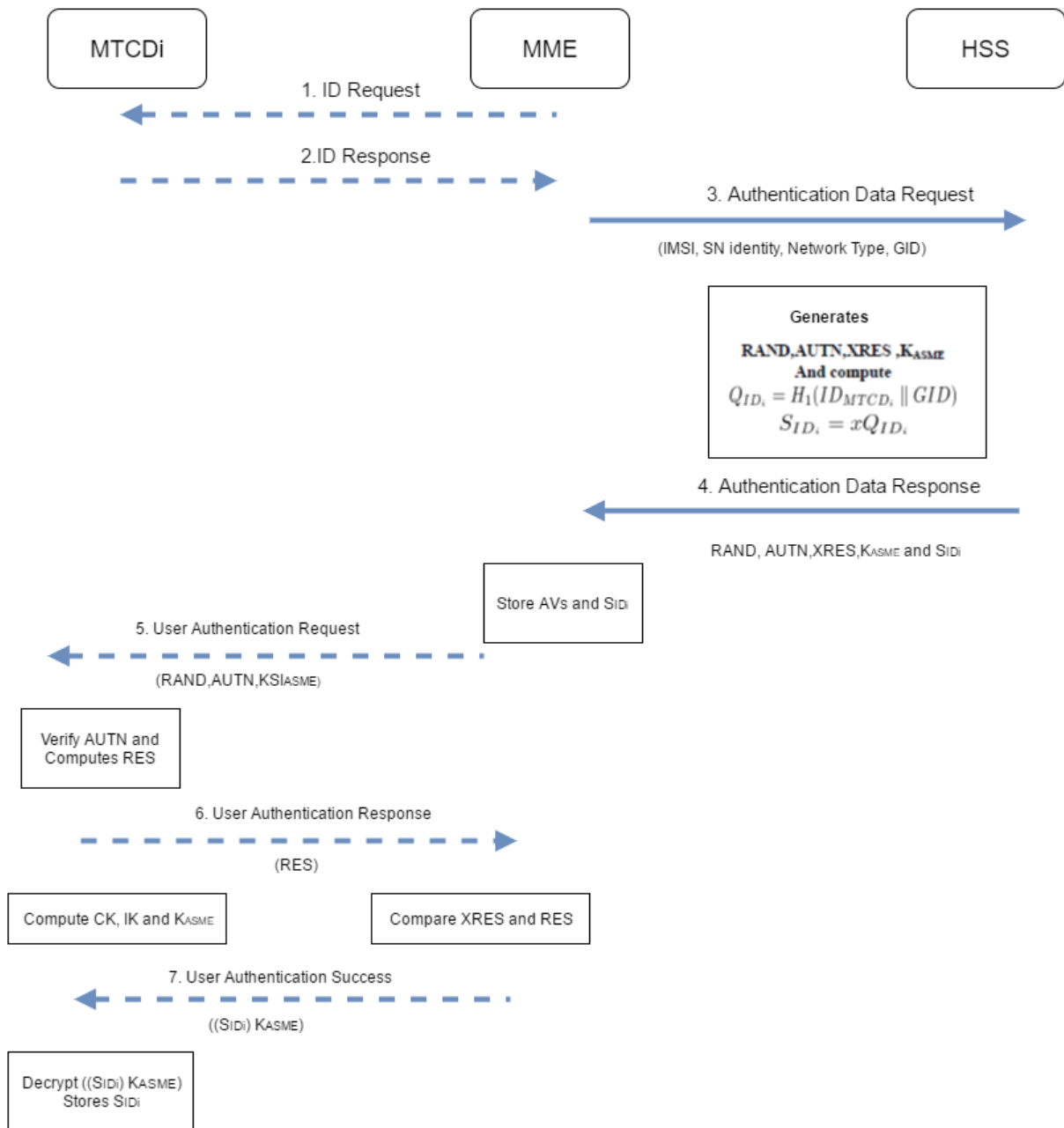


Figure 7 –Register phase in Cao et al. [9] protocol

**Group-based Access Authentication phase:** The mutual authentication between the group of MTCDs and MME is accomplished in this phase and the respective session keys are generated. Each MTCD generates a random number,  $r_{MTCDi}$  and a signature  $V_i$  that is sent to the group leader to be aggregated with other MTCDs signatures and then sent to MME to be verified. The MME generates an ECDSA signature that is encrypted with its private key and sent to the devices to be verified. Each device computes the MME public key and decrypts the signature for the verification of ECDSA. Figure 8 shows the detailed message exchange in the authentication phase.

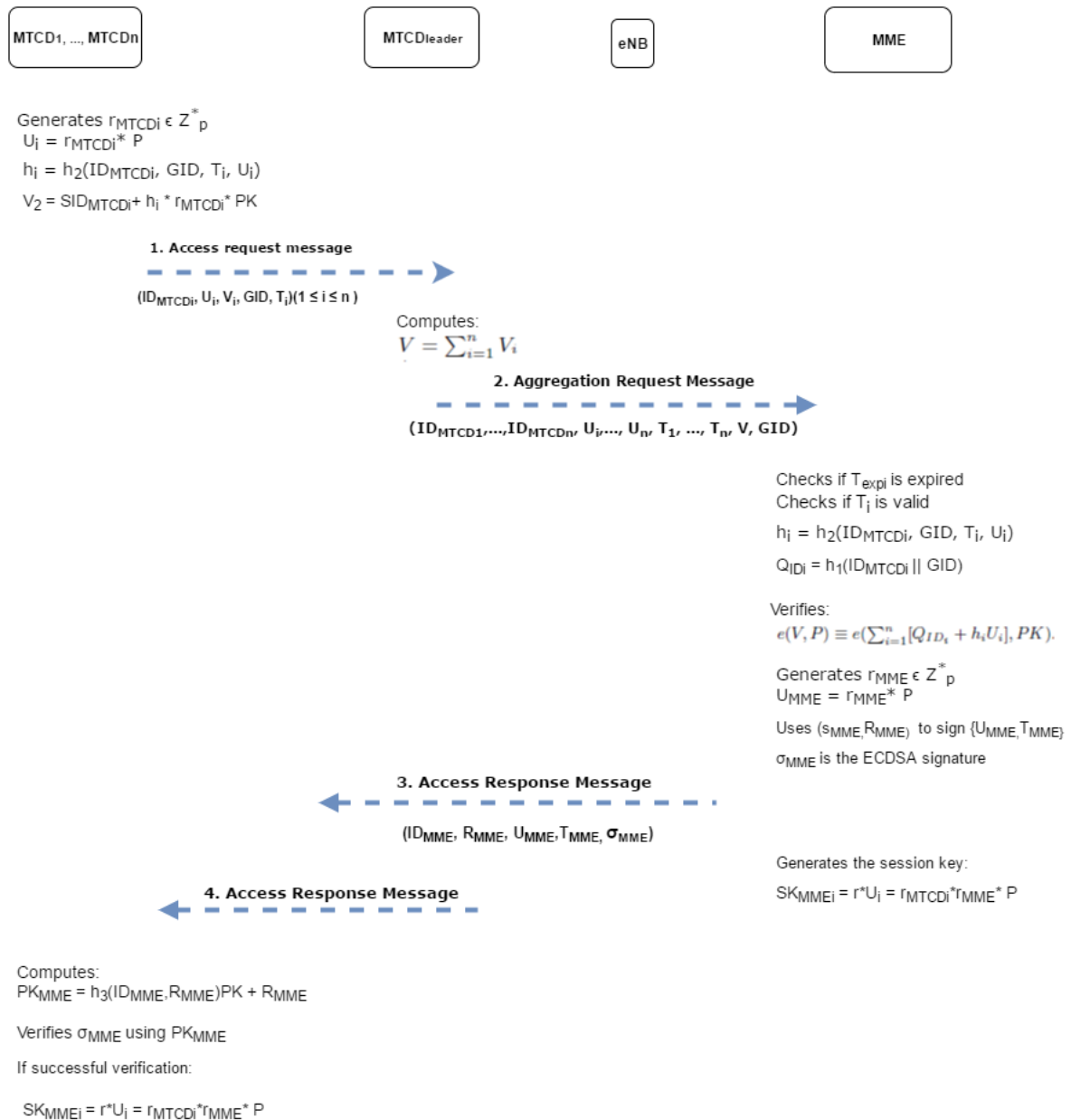


Figure 8– Group-based Access Authentication phase in Cao et al. [9] protocol

Cao et al. [9] do not guarantee the anonymity of the entities involved. All identities are sent in plaintext, which enable attackers to track and identify those involved in the authentication procedure. Additionally, if many attackers decide to send illegitimate signatures to the leader, it does not perform any type of verification to guarantee the legitimacy of the aggregated signature generated. Furthermore, when the MME receives the message with aggregated signature, it also receives the timestamps from all the MTCDs members of the group. It verifies all such timestamps, however, if any synchronization problem has occurred, the protection can suffer from loss of effectiveness. In cases with many invalid signatures, the MME executes the full process of MTCDs authentication several times until the denial of service.

According to Lai et al. [14] redirection attack may occur because neither the MME nor the HSS verify the authenticity of the Base Station involved in the process. For example, an attacker BS may trick the MTCDs by making them to believe it is legit and deviating the traffic to itself, so it can obtain user information. This attack could be avoided with the verification of the LAI' reported by the MTCD with the LAI of MME's knowledge. Giustolisi et al. [19] says Cao's protocol major issue is "...that registration and group-based access authentication must be executed with the same MME. This choice cancels the benefits provided by the group-based approach because the required signaling between MME and HSS is the same as required in traditional AKA."

### 3.2.2 GROUP AUTHENTICATION PROTOCOL DEVELOPED BY FU ET AL. [12]



Fu et al. [12] designed a privacy-preserving group authentication protocol based on ECDH that simultaneously authenticates all devices in a group. The scheme defines a family of pseudo identities for each device that protects their permanent identities. Despite its capacity to prevent DoS attacks, it cannot protect a redirection attack

The protocol main objective is to perform a secure and efficient mutual authentication and key agreement among groups of devices and a server MME and comprises two phases, namely initialization phase and mutual authentication phase. The architecture is similar to 3GPP's architecture and the MTC Server is located outside the EPC. All the entities involved in the protocol are presented on Table 3. Additionally, Table 4 provides all the notations described in Fu's paper.

Table 3 – Main entities involved in the architecture of Fu et al. [12] protocol

Abbreviation	Entity
$MTCD_{MEMBERS}$	Mobile Terminal Communication Device members of group
$MTCD_{leader}$	Mobile Terminal Communication Device's group leader
<b>HSS</b>	Home Subscriber Server
<b>MME</b>	Mobile Management Entity

Table 4 - Notations used by Fu et al. [12]

Notation	Definition
<b>p</b>	A k-bit prime
$Z_p$	A prime finite field
$G_1, G_2$	Two elliptic curve groups
<b>P</b>	Generator for group $G_1$
$h_1(.) - h_2(.)$	Message authentication function
$h_3(.) - h_4(.)$	Key generation function
<b>ID<sub>i</sub></b>	Identity of node i
$ID^z_{MTCDi-j}$	The z-th pseudo identity of $MTCD_{i-j}$
<b>x/PK</b>	Private/Public key of KGC ( $x \in Z_p^*$ ) ( $PK = xP$ )
<b>GK<sub>i</sub></b>	Group key of the i-th group
<b>GTK<sub>i</sub></b>	Group temporary key of the i-th group
<b>MAC<sub>x</sub></b>	Message authentication code computed by x
<b>XRES</b>	Expected authentication response computed by x
<b>RES</b>	Authentication response computed by x
<b>AUTN<sub>x</sub></b>	Authentication token generated by x
<b>  </b>	Concatenation operation
<b>*</b>	Elliptical curve scalar multiplication operation
	Secure channel
	Insecure channel

**Initialization phase:** In this phase, some important parameters used on authentication phase are generated  $\{p, G_1, G_2, e, P, PK, h_1, h_2, h_3, h_4\}$  by HSS and published. As in [12] the HSS also computes a family of unlikable pseudo identities  $\{ID^1MTCDi-j, ID^2MTCDi-j, \dots\}$  for each  $MTCD_{i-j}$  real identity  $IDMTCDi-j$ . The devices form groups based on common characteristics.

**Mutual authentication phase:** The devices select a group leader and proceed to mutual authentication phase. The objective of this phase is to accomplish a successful authentication procedure between each device and the MME and to generate a session key among them at the end of the process. The messages exchanged in this phase are described as follows:

1. Each device calculates a MAC and a random number and send it to the group leader.
2. The leader aggregates the messages received in one single message and send it to MME
3. The MME forwards the message received to the HSS, adding eNB's LAI.
4. The HSS verifies if the MAC informed by the devices are legit. If the verification passes, it calculates  $GTK, MAC_{HSS}, XRES$  and some others parameters and send them to the MME.
5. The MME calculates a  $MAC_{MME}$ , and forwards it with part of the information received by the HSS to the MTCDs.
6. Each MTCD calculates  $MAC'_{HSS}$  and  $MAC'_{MME}$  and verify if they are equal to  $MAC_{HSS}$  and  $MAC_{MME}$ . If the verification passes, the MTCD authenticates HSS and MME. Then, each of them calculate RES and send it to the group leader.
7. The group leader aggregates the messages received containing the RES of each device in one message and sends to the MME.
8. When the MME receives RES of the group, it verifies if it is equal to XRES received by the HSS. If the verification passes, the MME authenticates the devices and send them a success message. If it fails, the MME send them a failure message.
9. The session key is established among each MTCD and MME while the authentication procedure occurs. Figure 9 shows the mutual authentication message exchange.

Despite the protocol proposed by Fu et al. [12] being based on ECDH protocol, it does not explain how the management of devices entering and leaving the group occur, and consequently it does not specify a procedure to update the group keys in these cases. There is also a vulnerability to redirection attack, because it does not verify the legitimacy of the eNB in the authentication procedure, what make it susceptible to fake eNBs that can deviate the traffic to itself.

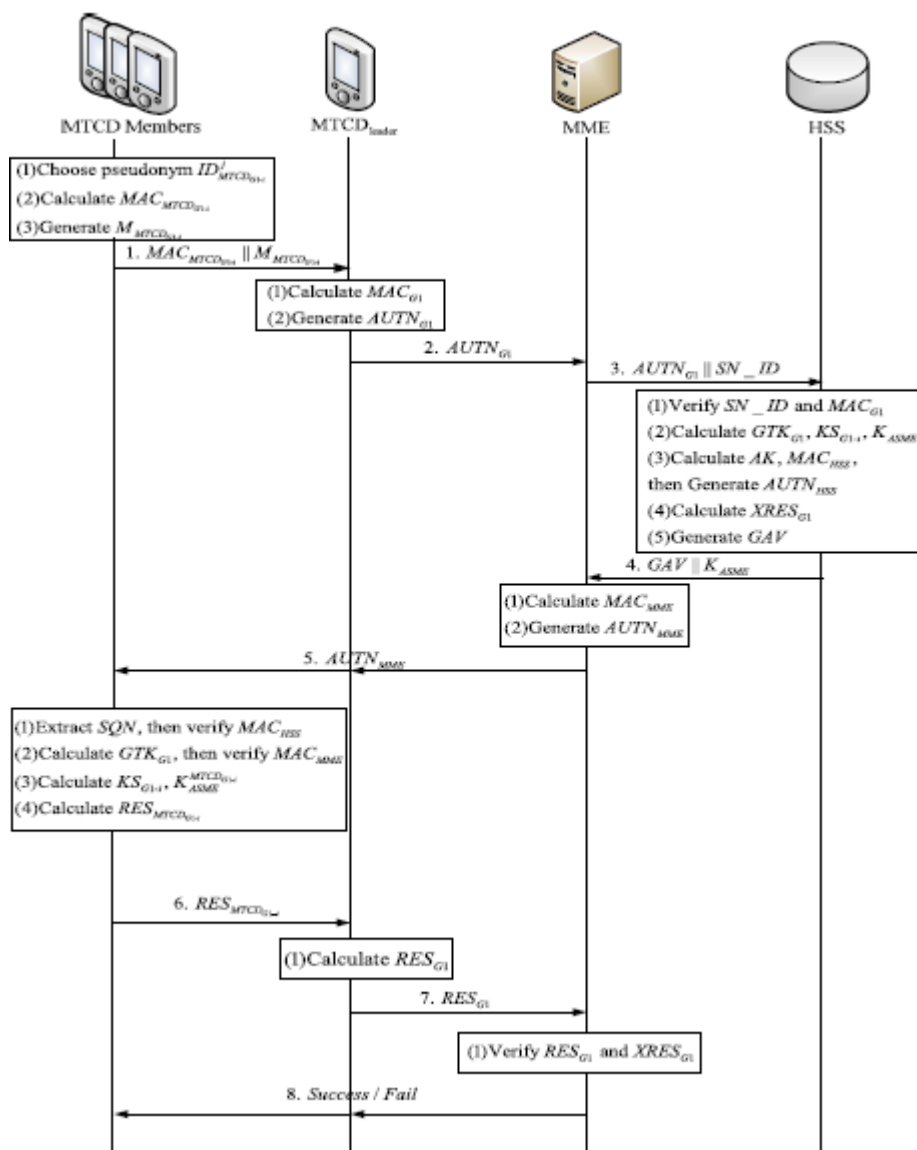


Figure 9 – Mutual authentication phase in Fu et al. protocol (source: [12]).

### 3.2.3 OTHER GROUP AUTHENTICATION PROTOCOLS

Lai et al. [14] created a secure protocol that employs ECDH and can authenticate groups of devices. However, it cannot authenticate all the devices simultaneously, because its authentication phase is divided into two procedures, i.e., one that authenticates the first MTCD that arrives in the server network and another to authenticate the other devices of the group. Only the first procedure involves the HSS, whereas the other involves the MTCDs and the MME. Although it is a robust protocol, it presents one of the highest costs in comparison to other proposals. Additionally, it does not elect a group leader; consequently, it does not assure the first device that arrives has the requisites to perform an important task.

Other protocols can be found in the literature. Among such protocols, we can mention for example Cao et al. [8].

### 3.2.4 COMPARATIVE EVALUATION

To provide a better visualization of important properties in the described protocols, Table 5 summarizes the most relevant characteristics of each protocol. Some of the properties are not considered to EPS-AKA [1] because it is not a group protocol. The MTC server is the entity responsible to manage the

MTC devices and may be located inside or outside the EPC. The group management field presents how the MTC server and HSS perform this management.

Table 5 - Described protocols' comparative table

	<i>EPS-AKA[1]</i>	<i>Lai et al. [14]</i>	<i>Cao et al. [9]</i>	<i>Fu et al. [12]</i>
<b>Group Authentication</b>	No	Yes	Yes	Yes
<b>Type of Cryptography</b>	Symmetric	Asymmetric ECDH	Asymmetric ECDH	Asymmetric ECDH
<b>MTC Server</b>	-	Inside EPC	Both	Outside EPC
<b>Leader Election</b>	-	No	Yes	Yes
<b>Group Management</b>	-	Table	Dynamic	Not Mentioned
<b>Bilinear Pairing</b>	No	No	Yes	No
<b>Location Area Verification (use of LAI)</b>	No	Yes	No	No

### 3.3 PROPOSED PROTOCOL

The proposed protocol is based in assymmetric cryptography, ECDH and bilinear pairing. It consists of two phases, namely initialization and mutual authentication and key agreement. The following basic assumptions are considered, related to the entities involved shown in Table 6:

1. The KGC is a trustful authority integrated with the HSS.
2. The channel between MME and HSS is secure.
3. The MTC Server is outside the EPC.



The network architecture, shown in Figure 10, is derived from the 3GPP [1] standards. Table 7 provides the notations used in the protocol.

Table 6 - Main entities involved in the architecture of the proposed protocol

<b>Abbreviation</b>	<b>Entity</b>
<b>MTC<sub>D<sub>i-j</sub></sub></b>	Mobile Terminal Communication Device $j$ of group $i$
<b>MTC<sub>D<sub>leader</sub></sub></b>	Mobile Terminal Communication Device's group leader
<b>HSS</b>	Home Subscriber Server
<b>MME</b>	Mobile Management Entity
<b>eNB</b>	Evolved Node B



Table 7 - Notations used in the proposed protocol

Notation	Definition
$p$	Large $k$ -bit prime number
$Z_p$	Prime finite field
$G_1, G_2$	Two elliptic curve groups
$P$	Generator of group $G_1$
$e(-,-)$	Bilinear pairing function $G_1 \times G_1 \xrightarrow{\quad} G_2$
$x/PK$	Private/Public key of KGC
$ID_a, TID_a$	Identity of entity $a$
$LAI$	Location Area Identification
$GK_i, GTK_i$	Group key / Group temporary key
$MAC_a$	Message Authentication Code of entity $a$
$r_a$	Random number generated by entity $a$
$SEC_y$	Secret value of node $y$
$SEK_{leader}$	Secret shared among HSS and the group leader
$C$	Group's verification value
$T_i$	Timestamp of group $i$
$h_1(.)$	Secure hash function
$h_2(.)$	Message authentication hash function
$h_3(.)$	Key generation hash function
$h_4(.)$	Session key hash function
$\parallel$	Concatenation operation
$\oplus$	XOR operation
$*$	Elliptical curve multiplication operation
	Secure channel
	Insecure channel

This work is based in the use of ECDH and challenge-response, as proposed by Fu et al. [12] and Lai et al. [14]. The ECDH was chosen because it can provide the generation of a secret shared among the devices and MME, which is used to provide forward/backward secrecy (FS/BS) to the session key. In addition, similarly to Cao et al. [9], bilinear pairing is used by the MME to provide simultaneous efficient and secure authentication of the group of devices. However, our protocol do not use identity-based signatures as proposed by [9]. Finally, our protocol's group organization and management is the same proposed by Choi et al. [11] and is used because it facilitates the group management and guarantees extra security to the protocol.

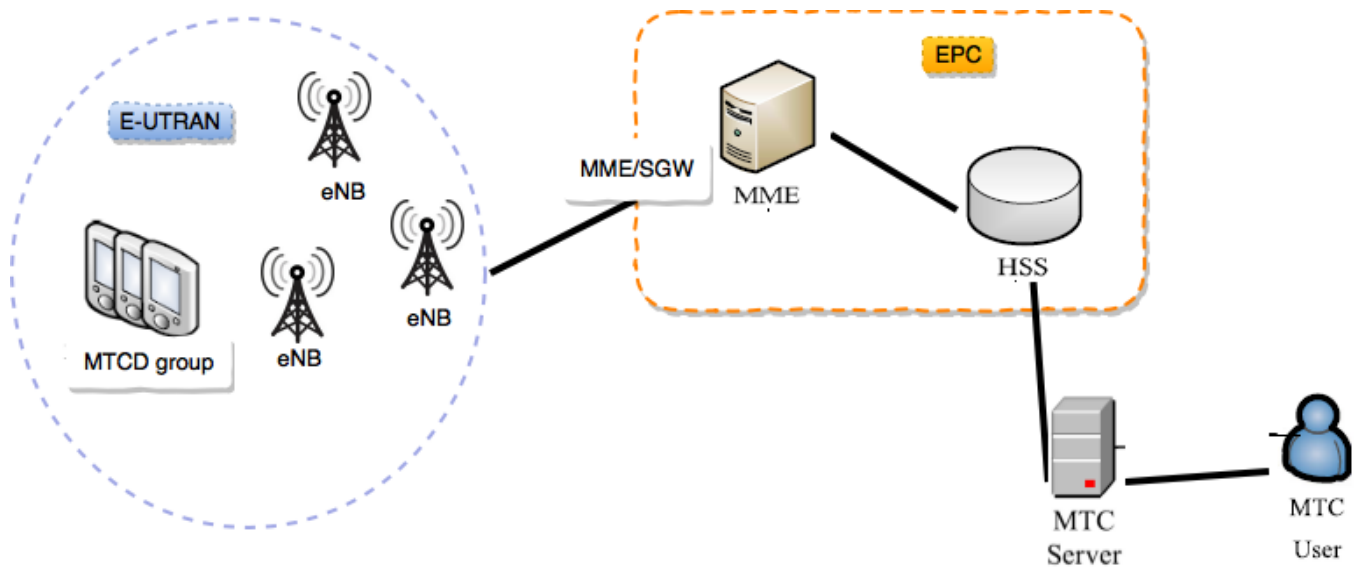


Figure 10 – Network architecture of the proposed protocol.

### 3.3.1 INITIALIZATION PHASE

A safe channel between MTCs, MME and HSS is considered in this phase. The manufacturer or supplier defines the  $IDI_j$  of each  $MTC_{i-j}$ . The devices organize themselves in groups, depending on their similar characteristics, and a group leader is chosen, based on battery life, location, computing power, communication and storage capacity. Examples of common procedures for leader election are presented by Abbasi et al. [28] and Chatterjee et al.[29] and are not detailed in this work because the discussion of this question is out of our objective. The group of  $MTC_{i-j}$  and the HSS are configured on the binary tree structure, presented in Choi et al. [11]. Each node of the tree has a secret value  $SECy$ , derived from its parents' secret nodes. The members of the group are placed on leaves and should never know their secret value. To guarantee that, they cannot know the secret values of their parents. Figure 11 shows the binary tree configuration and the secret values of the nodes (dotted circles) that member 4 cannot know.

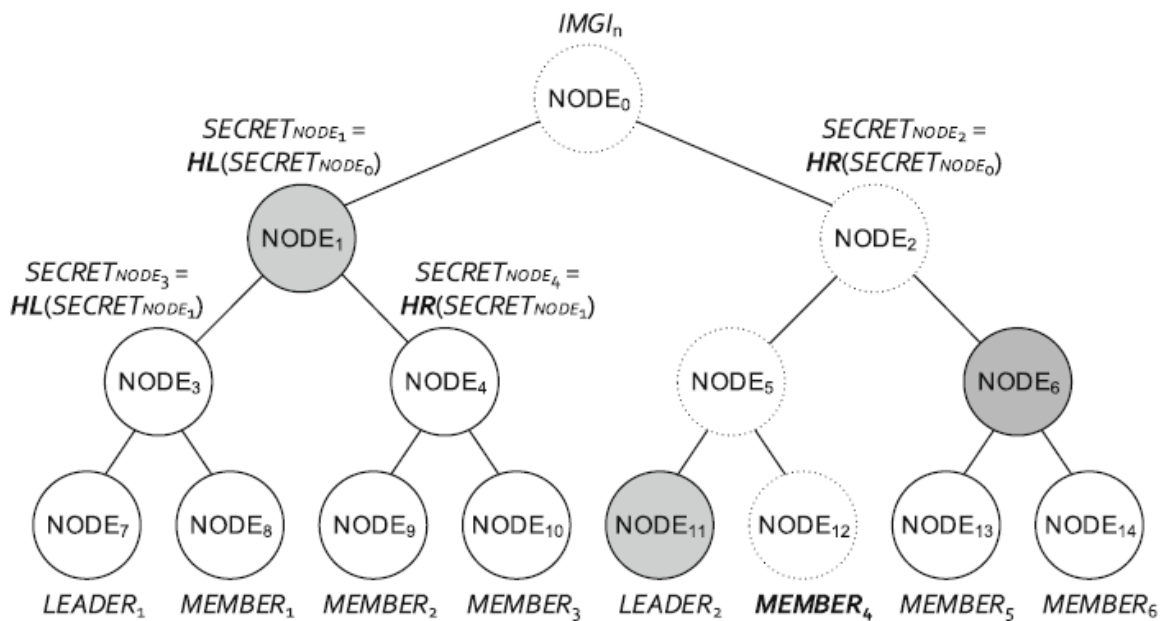


Figure 11 – Binary tree presented for group organization (source: [11]).

In this phase, the HSS proceeds as following:

1. Choses a random number  $x \in Z_p^*$  as the system master key and computes the system public key  $PK = x * P$ ;
2. Generates k random numbers,  $R_k \in Z_p^*$ , ( $k = 1, 2, \dots, i$ ) and calculates a set of temporary identities  $TID_{MTCDi-j}$  to each  $MTCD_{i-j}$ , as it follows:

$$TID = h_1(ID_{MTDCDi}) \oplus (R_k * x) \quad (3.1)$$

Defines a group identity  $ID_{Gi}$  and a group temporary identity  $TID_{Gi}$ . A different TID is used every time an authentication and key agreement procedure is executed.

3.

Generates a random number  $g \in Z_p^*$  and calculates the group key:

$$GK_i = h_3(SEC_{i-1} \oplus SEC_{i-2} \oplus \dots \oplus SEC_{i-j} \oplus g * PK) \quad (3.2)$$

Chooses  $SEK_{leader}$ , a secret shared among the HSS and the  $MTCD_{leader}$ .

Calculates

$$LMK_{MTCDleader} = GK_i * SEK_{leader} * P \quad (3.3)$$

4. Choses a random k-bits prime number and generates two elliptic curve groups, G1 and G2 of order p, and a generator point P in G1.
5. Selects four hash functions  $h_1(\cdot)$ ,  $h_2(\cdot)$ ,  $h_3(\cdot)$  and  $h_4(\cdot)$ ;
6. Selects a bilinear pairing function  $e(-, -)$
7. Publishes the system parameters  $\{p, P, PK, G1, G2, e, h_1, h_2, h_3, h_4\}$

### 3.3.2 MUTUAL AUTHENTICATION AND KEY AGREEMENT PHASE

In the proposed protocol, a group of  $MTCD_{i-j}$  who wants to be authenticated on a server network needs to proceed as follows:

1.  $MTCD_{i-j} \xrightarrow{(MAC_{MTCDi-j}, M_{MTCDi-j})} MTCD_{leader}$

Each  $MTCD_{i-j}$  chooses a random number  $r_{MTCDi-j} \in Z_p^*$  and computes:

$$A_{i-j} = r_{MTCDi-j} * P \quad (3.4)$$

$$MAC_{MTCDi-j} = h_2(ID_{MTDCDi-j} || ID_{Gi} || A) \quad (3.5)$$

$$M_{MTCDi-j} = (TID_{MTDCDi-j} || A) \quad (3.6)$$

Then, they send  $MAC_{MTCDi-j}$  and  $M_{MTCDi-j}$  to the  $MTCD_{leader}$ .

2.  $MTCD_{leader} \xrightarrow{(AUTH_{Gi})} MME$

The leader execute a XOR operation to join all the  $MAC_{MTCDi-j}$ , calculating the message authentication of the group  $MAC_{Gi}$ :

$$MAC_{Gi} = (MAC_{MTCDi-1} \oplus MAC_{MTCDi-2} \oplus \dots \oplus MAC_{MTCDi-j}) \quad (3.7)$$

It also calculates  $L_h$ , a challenge to HSS and  $AUTH_{Gi}$  that contains all the information about the group of devices:

$$L_h = h_1(LAI || ID_{Gi}) \quad (3.8)$$

$$AUTH_{Gi} = (MAC_{Gi} || M_{MTCDi-1} || M_{MTCDi-2} || \dots || M_{MTCDi-j} || TID_{Gi} || L_h) \quad (3.9)$$

The leader stores  $MAC_{Gi}$  and  $A_{i-j}$  and finally sends  $AUTH_{Gi}$  to the MME.

3.  $MME \xrightarrow{(AUTH_{Gi}, LAI')} HSS$

The MME knows the base station's LAI and send it to the HSS with  $AUTH_{Gi}$  so it can confirm the authenticity of the LAI reported by the group.

It stores  $MAC_{Gi}$  and  $A_{i-j}$  and sends  $(AUTH_{Gi}, LAI')$  to the HSS.

4. HSS  $\xrightarrow{(GTK_i, LMK_{leader}, r_{HSS})}$  MME

The first thing the HSS do when receive the message from the MME is to verify the authenticity of the LAI reported by the devices and the validity of the message. Then it calculates  $L'_h = h_1(LAI' || ID_{Gi})$  and verifies  $L'_h = L_h$ . If the verification fails, it sends a failing message to the group of devices and terminates the authentication procedure. If it passes, the HSS calculates all the  $MAC'_{MTCDi-j} = h_2(ID_{MTCDi-j} || ID_{Gi} || A)$  and in the sequence, it calculates  $MAC'_{Gi} = (MAC_{MTCDi-1} \oplus MAC_{MTCDi-2} \oplus \dots \oplus MAC_{MTCDi-j})$ . Next, it verifies if  $MAC_{Gi} = MAC'_{Gi}$ . If the verification fails, it sends to the group a MAC failing message. If it holds, the HSS chooses a random number  $r_{HSS}$ , calculates the group temporary key and a verification value, so the MME can authenticate the group:

$$GTK_i = h_3(GK_i || r_{HSS}) \quad (3.10)$$

A new group temporary key is generated at each session. Then, it sends  $GTK_i$ ,  $r_{HSS}$  and  $LMK_{leader}$  calculated in initialization phase to the MME.

5. MME  $\xrightarrow{(AUTH_{MME})}$   $MTCD_{leader}/MTCD_{i-j}$

After receiving the message from HSS, the MME stores  $LMK_{leader}$ , chooses a random number  $r_{MME} \in Z_p^*$  and calculates:

$$MAC_{MME} = h_2(r_{MME} * P || GTK_i) \quad (3.11)$$

$$AUTH_{MME} = (r_{MME} * P || MAC_{MME} || r_{HSS}) \quad (3.12)$$

Next, it broadcasts  $AUTH_{MME}$  to all the group members.

6.  $MTCD_{leader}/MTCD_{i-j}$   $\xrightarrow{\text{Success/Failure}}$  MME

When each  $MTCD_{i-j}$  receives the message, it computes:

$$GTK_i = h_3(GK_i || r_{HSS}) \quad (3.13)$$

$$MAC'_{MME} = h_2(r_{MME} * P || GTK_i) \quad (3.14)$$

Then, they verify if  $MAC_{MME} = MAC'_{MME}$ . If the verification fails, they send a MAC failure message to the MME. In the other hand, if the verification pass, the MME is authenticated by the devices.

7.  $MTCD_{leader}$   $\xrightarrow{C, Ti}$  MME

Now, the leader prepares the devices' verification value C, so the MME can authenticate the group. It uses  $A_{i-j}$  and  $MAC_{Gi}$  that it previously received and stored inside  $AUTH_{Gi}$ . Next, it computes:

$$C = (A_{i-1} \oplus A_{i-2} \oplus \dots \oplus A_{i-j} \oplus MAC_{Gi}) * SEK_{leader} * GK_{Gi} \quad (3.15)$$

Then, it sends (C,Ti) to the MME.

8. MME  $\xrightarrow{\text{Success/Failure}}$   $MTCD_{leader}$

The first thing the MME does is to check if Ti still is valid. If it is not, the MME send a failure message to the  $MTCD_{leader}$ . Using the information it stored about the devices, verification value C received from the leader and the verification value received from the HSS, it verifies the authenticity of the devices by calculating D and executing the bilinear pairing below:

$$D = (A_{i-1} \oplus A_{i-2} \oplus \dots \oplus A_{i-j} \oplus MAC_{Gi}) \quad (3.16)$$

$$e(C, P) \equiv e(D, LMK) \quad (3.17)$$

If the verification does not pass, it sends a failure message to the  $MTCD_{leader}$ . If it passes, authenticates all the devices in the group at the same time. The mutual authentication is finished.

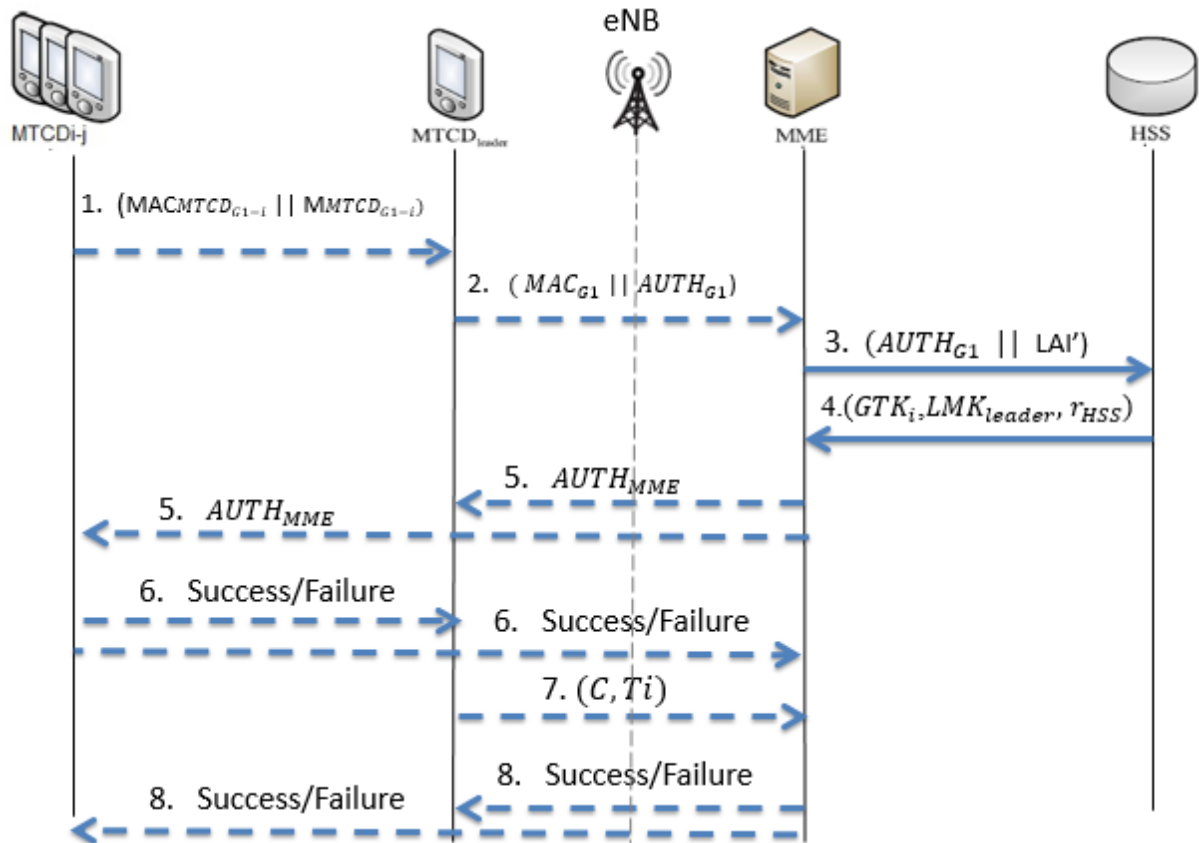


Figure 12 – The authentication phase of the proposed protocol.

By the model of binary tree described in Choi et al. [11], the MME joins the binary tree at the end of the authentication phase. It is associated to an empty leaf, with a secret value  $SEC_y$ , as showed in Figure 13.

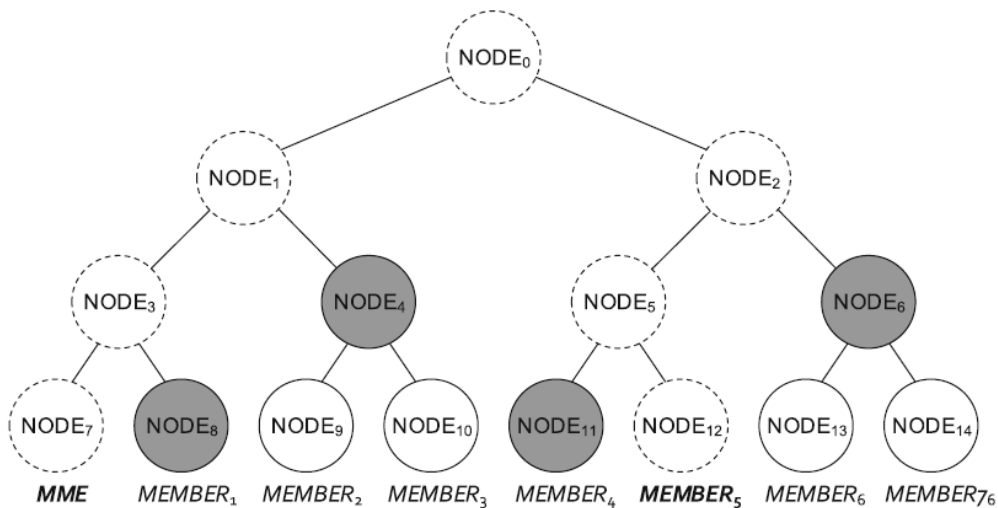


Figure 13 – The binary tree after the entrance of MME (source [11]).

The MME and  $MTCD_{i-j}$  compare the secret values they know and discover the ones in common. The grey circles in Figure 13, are the common values between them. At this point, the session key  $SK_{j-MME}$  between MME and each  $MTCD_{i-j}$  is computed:

$$SK_{i-j-MME} = (SEC_{i-a} \oplus SEC_{i-b} \oplus \dots \oplus SEC_{i-z}) * r_{MTCDi-j} * r_{MME} * P \quad (3.18)$$

Where  $SEC_a, SEC_b \dots SEC_z$  are the common secret values between  $MTCD_{i-j}$  and  $MME$ . This model of session key is based on the session key presented by Choi et al. [11] and also can be used to device-to-device communication ( $D2D$ ) between all the  $MTCD_{i-j}$ . A different session key is generated at the end of each session performed by the group. All the keys generated by our protocol must be refreshed at some point, depending on the security policy adopted by the carrier or company responsible for the network. Additionally, the group key needs an update with MME node's secret value, because it is now part of the tree, and considered a group member. The procedure is the same as described in session 3.3.3.

### 3.3.3 GROUP KEY UPDATE

The group key must be updated every time a member enters or leaves the group of devices. The procedures are based on the binary tree structure, as described by Choi et al. [11].

#### 3.3.3.1 MEMBERS JOINING THE GROUP

When a new member joins the group, the current  $GK_i$  needs an update so the new member do not have access to the messages exchanged before its entrance. This procedure is shown on the equation below and is necessary to guarantee the backward secrecy.

$$GK'_i = h_3(GK_i \oplus SEC_{i-y}) \quad (3.19)$$

Where  $GK'_i$  is the new group key and  $SEC_{i-y}$  is the secret value of the node where the new member is located. The new member is not able to discover the old  $GK_i$  because it does not know its own  $SEC_{i-y}$  and cannot revert the hash function.

#### 3.3.3.2 MEMBERS LEAVING THE GROUP

There is also the possibility that a device needs to leave the group. This may be justified by many reasons, as the loss of common characteristics that are the requisites to be part of the group or simple when finished their tasks. This includes the MME that can be changed at any time, as the group's interests changes. The procedure is shown on the equation below and is necessary to guarantee the forward secrecy.

$$GK''_i = GK_i \oplus SEC_{i-y} \quad (3.20)$$

In this case, the hash function is not necessary, because the hash function executed in the joining procedure is enough to prevent the reversal of old group keys. The member leaving the group cannot discover the group key because it does not know its own  $SEC_{i-y}$ .

## 3.4 SECURITY ANALYSIS OF THE PROPOSED PROTOCOL AUTHENTICATION PROCEDURE

In this section, we present an analysis of the proposed protocol security objectives, followed by a comparative table of these objectives in the proposed protocol and the analyzed protocols [1], [9], [12] and [14]. Table 7 sums up all the security objectives analyzed in this section.

**Mutual Authentication:** Each  $MTCD$  device calculates  $MAC_{MTCDi-j}$  and send it to  $MTCD_{leader}$ . The leader aggregates all the messages using a XOR operation and send  $MAC_{Gi}$  to the HSS. The HSS verify all the members of the group by verifying  $MAC_{Gi}$ . Then, calculates  $GK_i$  and sends it with  $LMK_{leader}$  to MME. Next, the MME calculates  $MAC_{MME}$  and  $AUTH_{MME}$  and broadcasts them to each  $MTCD$ . The  $MTCDs$  first calculate  $GK_i$ , so they can calculate  $MAC'_{MME}$  and authenticate the MME by verifying if  $MAC'_{MME} = MAC_{MME}$ . If the verification passes, the leader gathers all the information it knows about

each MTCD, calculates the verification value  $C$  and in the sequence, sends it to the MME. The MME gathers all the information it knows about the group of MTCDs, calculates  $D = (A_{i-1} \oplus A_{i-2} \oplus \dots \oplus A_{i-j} \oplus MAC_{Gi})$  and then executes bilinear pairing:  $e(C, P) \equiv e(D, LMK_{MTCDi-j})$ . If the verification passes, it authenticates the all devices in the group and the mutual authentication process is complete.

**Confidentiality and Integrity:** All the data exchanged between an MTCD and the MME is encrypted and protected by the session key, generated at the end of the authentication process, guaranteeing its confidentiality and integrity.

**Privacy (Anonymity):** Each MTCD has a group of TIDs set in the initialization phase that is used on the authentication process to protect its permanent identity. Nothing is harmed if an attacker obtains one of these TIDs. Only the HSS can access permanent IDs.

**Perfect FS/BS:** Every time a member enters or leaves the group its group key is updated, following the process described in section 3.3.3. Hence, even if a member entering the group have access to an old GK, it cannot access the messages exchanged before its entrance. Likewise, to a member that leaves the group, even with the old keys it cannot access the messages exchanged in the future. Additionally, the session key is calculated using the ECDH problem, that guarantees strong FS/BS, and secret values of the binary tree that only the respective MTCD and the MME knows.

**Replay Attack:** All the entities involved use different random values, freshly calculated in each authentication process. An attacker cannot forge messages using old random values.

**DoS Attack:** The HSS only starts the verification process of  $MAC_{Gi}$  if  $L_h$  is valid. It calculates  $L'_h$  with the LAI received from MME and  $ID_{Gi}$ . Next, it compares  $L'_h = L_h$ . If it is not valid, the HSS send a failing message to the MTCDs and then terminates the authentication procedure. Additionally, the  $MTCD_{leader}$  sends SQN to the MME and it only start the verification of  $C$  if SQN is valid.

**Man-in-the-Middle attack:** The session key cannot be forged using values exchanged on the communication channel because it uses the secret values from the binary tree and the ECDH problem to be calculated. An attacker also is not able to forge neither GK nor GTK, because they are not exposed.

**Redirection attack:** Each MTCD includes the base station LAI in  $MAC_{MTCDi-j}$  and the MME, that also knows the devices base station LAI send it to the HSS in secure channel. If an attacker tries to forge LAI the verification of  $MAC_{MTCDi-j}$  fails and the redirection attack is avoided.

**Impersonation attack:** Neither the devices or the group identity is disclosed. Hence, an attacker cannot impersonate any of them. Additionally, it cannot generate a genuine verification value  $C$ .

Table 8 - Comparison of security objectives between protocols

Security Objectives	EPS-AKA [1]	SE-AKA[14]	FU[12]	GBAAM [9]	Proposed Protocol
<b>Mutual authentication and Key Agreement</b>	Yes	Yes	Yes	Yes	Yes
<b>Confidentiality</b>	No	Yes	Yes	Yes	Yes
<b>Integrity</b>	No	Yes	Yes	Yes	Yes
<b>Privacy (Anonymity)</b>	No	Yes	Yes	No	Yes
<b>Perfect FS/BS</b>	No	Yes	No	Yes	Yes
<b>Resistant to replay attack</b>	No	Yes	Yes	Yes	Yes
<b>Resistant to DoS attack</b>	No	Yes	Yes	No	Yes
<b>Resistant to Man-in-the-Middle attack</b>	No	Yes	Yes	Yes	Yes
<b>Resistant to redirection attack</b>	No	Yes	No	No	Yes
<b>Resistant to impersonation attack</b>	No	Yes	Yes	Yes	Yes

### 3.5 PERFORMANCE EVALUATION AND COMPARISONS

In this section, the performance of the proposed protocol is evaluated and compared to the performance of 3GPP's standard [1] and to some other group protocols, [9], [12] and [14]. All the group protocols were designed based on the same assumptions considered in our protocol, as follows:

- They are based in the MTC adapted network architecture presented in Figure 10 and consider that a secure channel between the MME and the HSS is established;
- Present an initialization/registration phase responsible of generating the parameters used in the authentication and key agreement phase;
- Present an authentication and key agreement phase, performed among the group of devices and the MME, where a different session key is generated every time it is executed.

#### 3.5.1 COMPUTATIONAL COST

The comparison of the computational cost of the proposed protocol with the other schemes analyzed is presented on Table 9. The values adopted were carefully chosen, based on the values used by [9],[11] and [14]. The cost of each operation is showed on Table 10. The time to perform an XOR operation was omitted, since it is negligible if compared with others and the following operations are considered:

Table 9 - Time cost of each operation considered in *ms*.

Notation	Cost (ms)	Description
$T_{hash}$	0.06 ms [11]	Cost of a one-way hash operation
$T_{mul}$ (MTC/Core)	1.537/0.475 ms [9][14]	Cost of a multiplication operation over elliptical curve
$T_{mtp}$ (MTC/Core)	1.537/0.475 ms[9][14]	Cost of a map to point hash operation
$T_{pair}$	4.5 ms [11]	Cost of a bilinear pairing operation
$T_{mod}$	0.12 ms [11]	Cost of a modular operation
$T_{add}$	0.12 ms [11]	Cost of additive value over elliptical curve
$T_{aes}$	0.16 ms [11]	Cost of AES encryption operation.

It is considered an environment with  $n$  devices, divided into  $m$  groups. The proposed protocol takes  $3.78n + 11.26m$  milliseconds to successful complete an authentication process. The MTCs perform hash, multiplicative value over elliptical curve and additive value over elliptical curve operations, in a total of:  $(2n+m)T_{mul} + (3n+m)T_{hash} = 3.25n + 1.76m$  milliseconds in operations. The core network comprises MME and HSS. The MME performs hash, multiplicative and additive value over elliptical curve and bilinear pairing operations. The HSS just perform hash operations. Together, they execute a total of  $(n+m)T_{mul} + (n+3m)T_{hash} + 2mT_{pair} = 0.53n + 9.66m$  milliseconds in operations.



Table 10 – Computational cost comparison between protocols in authentication phase

Protocol	MTCDs (ms)	Core Network (ms)	Total (ms)
EPS-AKA[1]	$6n\text{Thash} + n\text{Taes} = 0.52n$	$6n\text{Thash} + n\text{Taes} = 0.52n$	$1.04n$
SE-AKA[14]	$2n\text{Tmul} + (4n+m)\text{Thash} = 3.31n + 0.06m$	$2n\text{Tmul} + (2n+3m)\text{Thash} = 1.07n + 0.18m$	$4.38n + 0.24m$
GBAAM[9]	$4n\text{Tmul} + 2n\text{Thash} + 3n\text{Tadd} + n\text{Taes} = 6.79n$	$(2n+m)\text{Tmul} + n\text{Thash} + 2n\text{Tadd} + n\text{Tmtp} + 2m\text{Tpair} + m\text{Taes} = 1.73n + 9.64m$	$8.52n + 9.64m$
FU[12]	$2n\text{Tmul} + 7n\text{Thash} = 3.5n$	$(n+m)\text{Tmul} + (3n+4m)\text{Thash} = 0.66n + 0.72m$	$4.16n + 0.72m$
Proposed Protocol	$(2n+m)\text{Tmul} + (3n+m)\text{Thash} = 3.25n + 1.6m$	$(n+m)\text{Tmul} + (n+3m)\text{Thash} + 2m\text{Tpair} = 0.53n + 9.66m$	$3.78n + 11.26m$

In Table 10 the proposed protocol has lower costs than [9], [12] and [14]. It only presents higher costs compared to 3GPP EPS-AKA [1], that is not a group authentication protocol, or if the number of groups (m) is high, because the bilinear pairing operation has high costs that varies depending of m. To be more specific, its performance decreases to groups with less than 28 devices and present the best performance to groups with more than 28 devices. Even with higher computational costs, it must be considered that the proposed protocol offers lower communication costs than the other studied protocols. Additionally, our protocol has accomplished more security features than [1], which has no protection against several attacks, as showed on Table 7. This comparison is also showed in Figure 14.

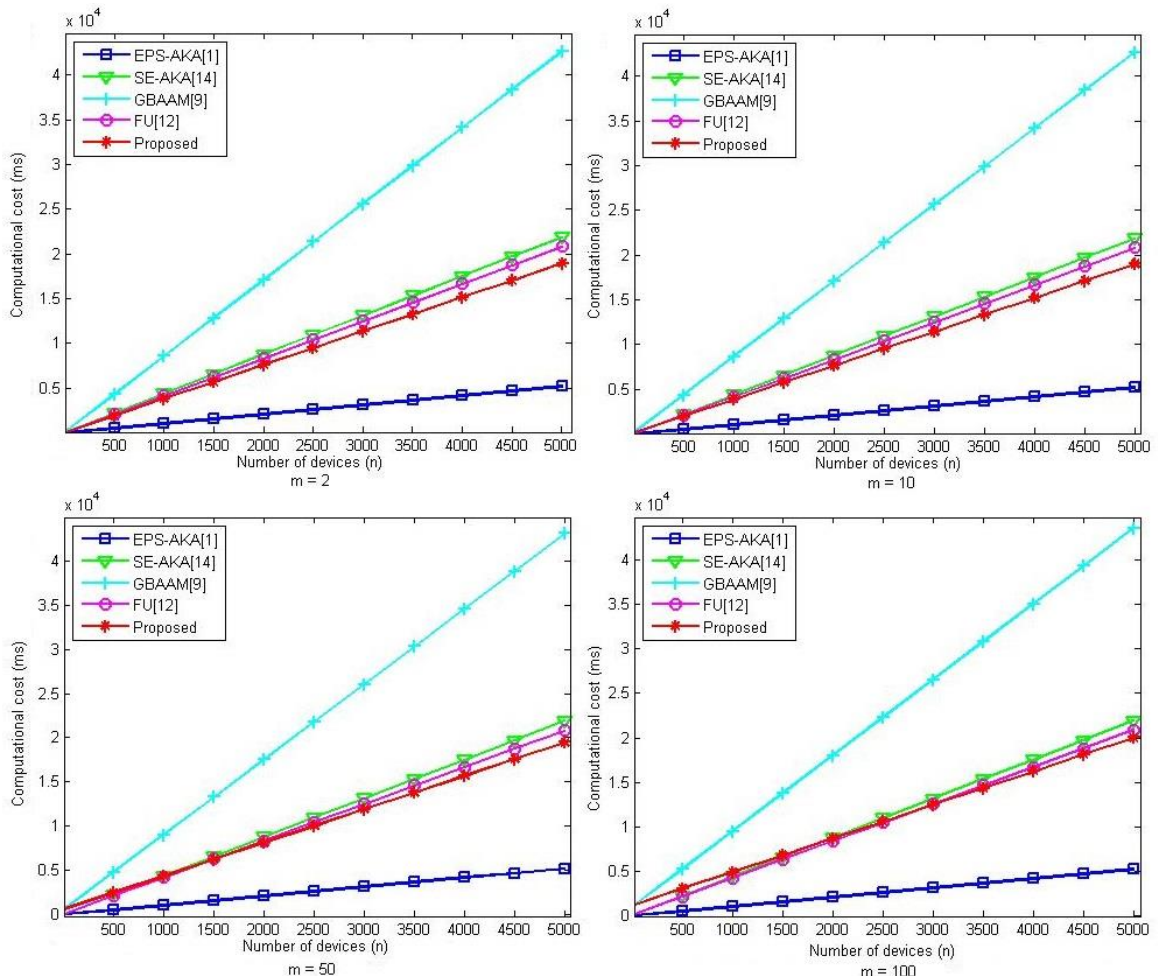


Figure 14 – Comparison of computational cost.

### 3.5.2 COMMUNICATION COST

The communication cost was measured in bits, by message exchanged. The values adopted to each parameter transmitted is presented on Table 11. The values adopted were carefully chosen, based on the values used by [9],[11] and [14].

Table 11 – Communication cost of each parameter transmitted.

Parameter	Size (bits)	Parameter	Size (bits)
<b>ID/TID</b>	128	<b>KDF</b>	128
<b>ECDH</b>	192	<b>AMF</b>	48
<b>MAC</b>	64	<b>Rand</b>	128
<b>Hash</b>	128	<b>LAI</b>	40
<b>PN, SQN</b>	128	<b>KL</b>	256
<b>Ti</b>	32	<b>AES</b>	256

It is considered an environment with n devices, divided into m groups. The calculations were based on the quantity of parameters exchanged in each message. For example, in our protocol, the second message is sent from all the MTCDs to the MTCD<sub>leader</sub>. Each of them send  $MAC_{MTCDi-j} = H_1(ID_{MTDCDi-j} || ID_{Gi} || A)$  and  $M_{MTCDi-j} = (TID_{MTDCDi-j} || A)$  to the leader. Hence, the message has a TID with 128 bits, a MAC function with 64 bits and an ECDH random number with 192 bits accounted, in a total of 384(n-m) bits. It is important to notice that the n devices considered include the leader and for that reason, only n-1 devices send this message to the leader. For m groups, n-m devices will send that message to their respective leaders. Table 12 compares the communication cost of the proposed protocol with other protocols analyzed.

Table 12 - Communication cost in bits by message and in total.

	M1	M2	M3	M4	M5	M6	M7	M8	TOTAL
<b>EPS- AKA[1]</b>	-	128 bits	256 bits	704 bits	576 bits	128 bits	-	-	1792n bits
<b>SE- AKA[14]</b>	-	-	384n + 96m bits	1072n -552m bits	256n+ 176m bits	1072m bits	256m bits	-	1712n + 1048m bits
<b>GBAAM[9]</b>	672(n-m) bits	352n + 320m bits	992m bits	992m bits	-	-	-	-	1024n + 1632m bits
<b>FU[12]</b>	512(n-m) bits	448n + 64m bits	448n + 192m bits	944m bits	880m bits	880m bits	128(n-m) bits	128m bits	1536n + 2448m bits
<b>Proposed Protocol</b>	384(n-m) bits	320(n+m) bits	320n +360m bits	448m bits	384m bits	224m bits	-	-	1024n + 1352m bits

From Table 12 it is possible to conclude that the proposed protocol has the lowest communication cost when compared to the other protocols analyzed. It has an almost equal cost to the protocol proposed by [9]. However, [9] has a much higher computational cost than our protocol. Furthermore, it has some lack of security. It does not avoid redirection attack and does not guarantee anonymity of the entities involved during the authentication process. Figure 15 also shows this comparison.

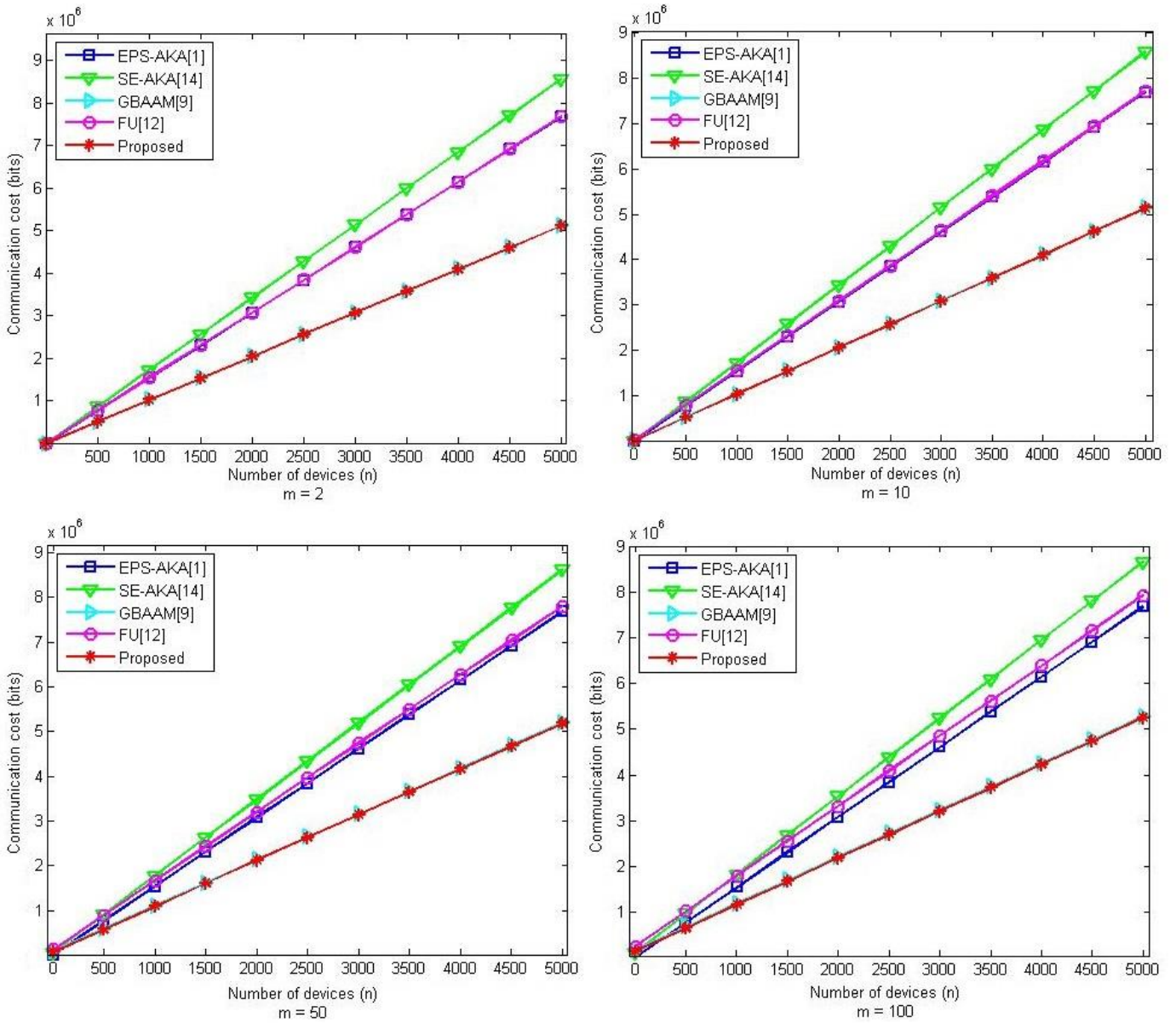


Figure 15 – Comparison of communication cost.

To better visualize the communication costs enhancements accomplished by our protocol when compared to 3GPP EPS-AKA[1], we also present an improvement rate ( $IR$ ), as described in [18]. The  $IR$  equation can be seen below:

$$IR = \frac{EPS\backslash AKA - Proposed}{EPS\backslash AKA} \quad (3.21)$$

After replacing the respective values in the equation, we obtained the following:

$$IR = 0.42 - 0.75 \frac{m}{n} \quad (3.22)$$

From the equation above, it can be deduced that the maximum improvement the proposed protocol can accomplish in relation to EPS-AKA is 42%. Hence, the proposed protocol stabilizes its enhancements in 0.42. This stabilization is showed in Figure 16, which also shows that the proposed protocol also has a better  $IR$  than the other protocols studied.

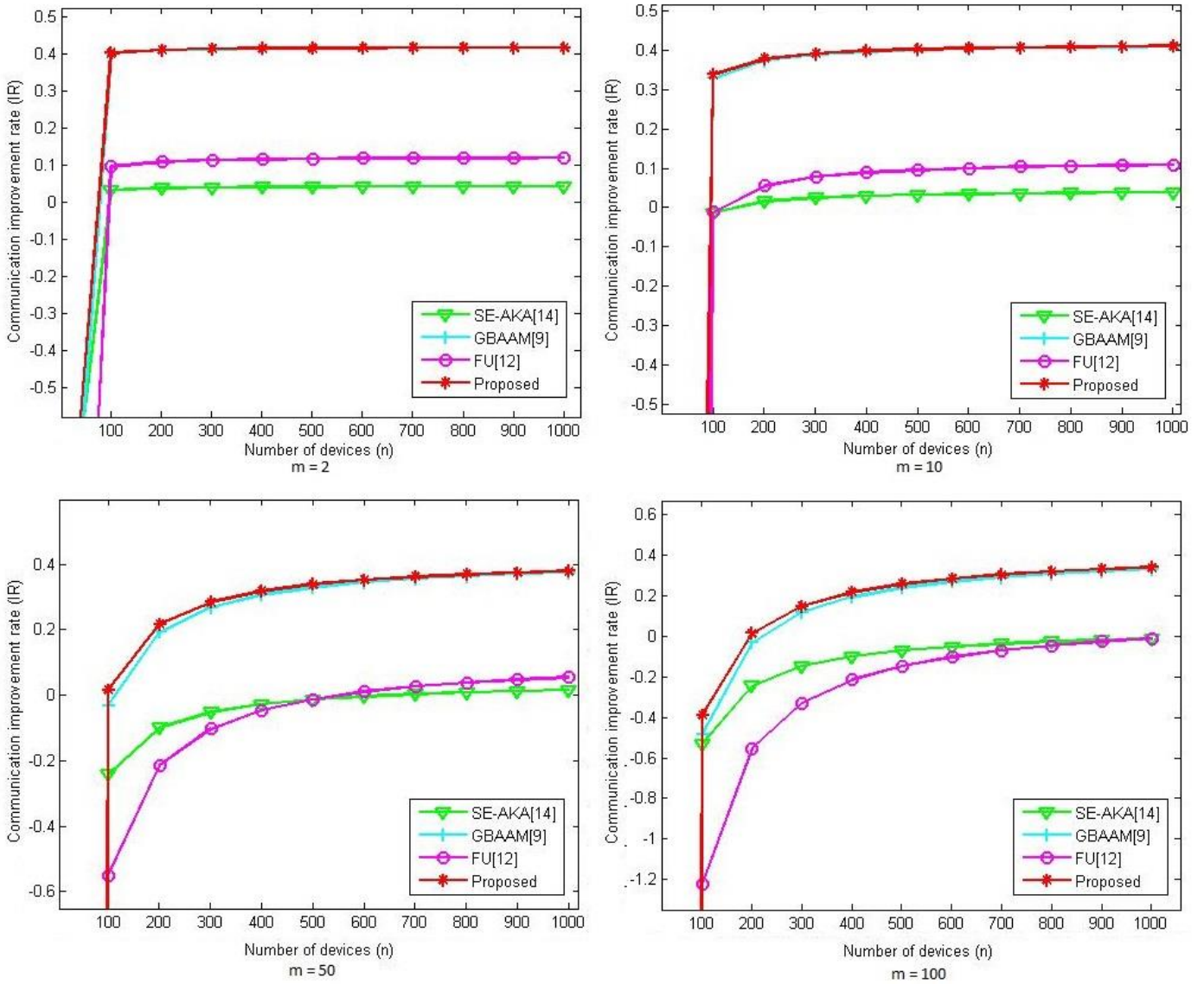


Figure 16 – Comparison of improvement rate for communication cost.

### 3.5.3 VERIFICATION COST

This section evaluates the verification cost of a case with one invalid device. If an attacker tries to authenticate itself as a MTCD in the proposed protocol, the bilinear pairing verification will fail. The MME sends a failure message to the leader and asks for all the verification values  $C_{i-j}$ ,  $MAC_{MTCDi-j}$  and  $A_{i-j}$ .

In order to discover which MTCD is an attacker, the devices  $MAC_{MTCDi-j}$  are divided into two groups, and two  $MAC_{Gi}$  are generated. The information batch  $D = (A_{i-1} \oplus A_{i-2} \oplus \dots \oplus A_{i-j} \oplus MAC_{Gi})$  and the verification values C are organized in two groups each and then, the bilinear pairing verification is executed to both. One of the groups will still fail the verification. Then, this group is divided into two and the procedure is repeated. The time to perform an XOR operation was omitted, since it is negligible if compared with others. The re-verification procedure takes  $2T_{pair}$ . This means that one invalid verification value is discovered after  $T_{verif} = T_{first} + 2[\log_2 n] * 2T_{pair}$ , where  $T_{first} = mT_{mul} + (n + 3m)T_{hash} + 2T_{pair}$ .

Table 13 shows the comparison of the verification cost between the proposed protocol and the other protocols analyzed. This comparison is better visualized in Figure 17, proving that the proposed protocol has better verification cost than the other protocols studied.

Table 13 – Verification cost in bits by message and in total.

Protocol	First verification (ms)	Total verification cost (ms)
SE-AKA[14]	$2nT_{mul} + (2n + 3n)T_{hash}$	$T_{first} + 2[\log_2 n] * T_{hash}$
GBAAM[9]	$nT_{mul} + nT_{hash} + nT_{mtp} + 2nT_{add} + 2T_{pair}$	$T_{first} + 2[\log_2 n] * 2T_{pair}$
FU[12]	$(3n + 4)T_{hash} + (n + 1)T_{mul}$	$T_{first} + 2[\log_2 n] * T_{hash}$
Proposed protocol	$T_{mul} + (n + 3)T_{hash} + 2T_{pair}$	$T_{first} + 2[\log_2 n] * 2T_{pair}$

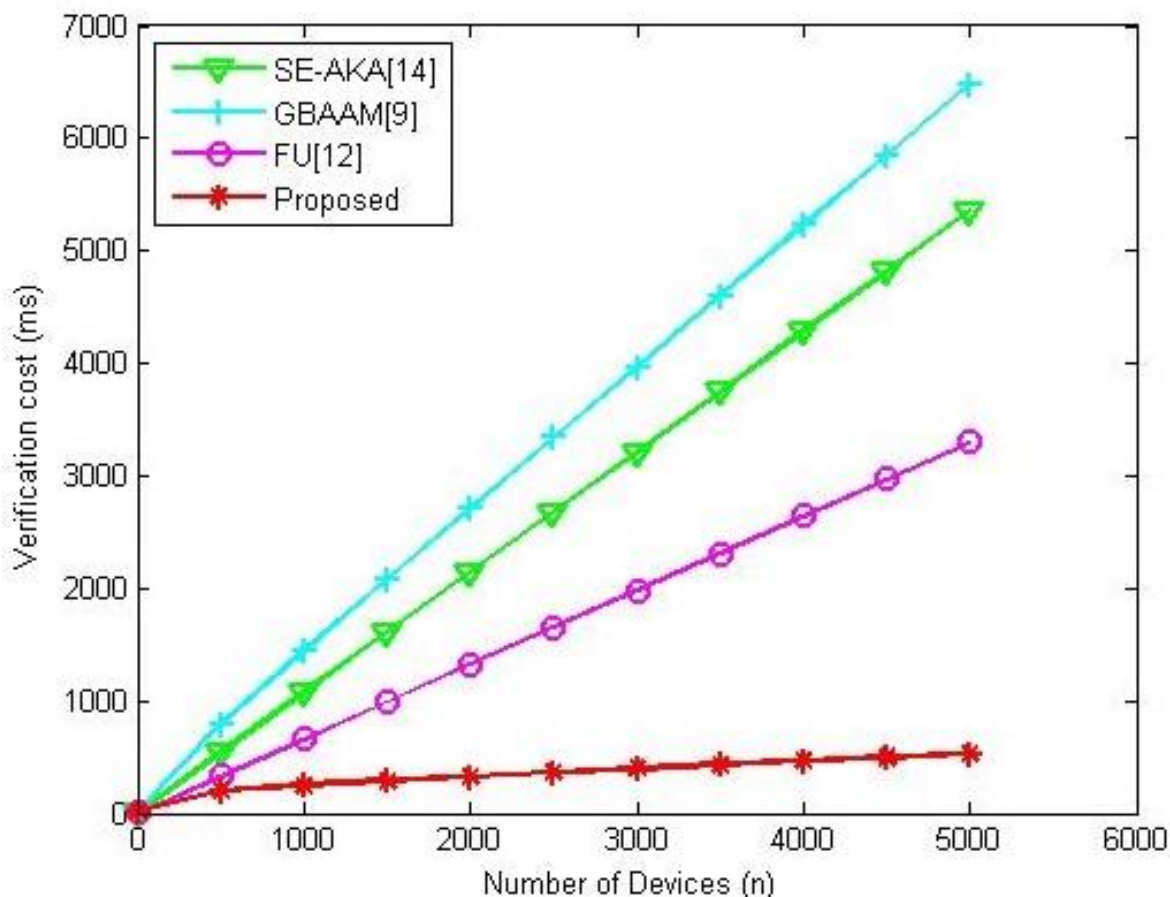


Figure 17 – Comparison of verification cost.

### 3.5.4 STORAGE COST

This section compares the storage cost of the proposed protocol with the protocols analyzed. Only the parameters derived from the authentication procedure are considered. Each entity may need to store some parameters in order to perform the authentication procedure. The storage cost comparison with the analyzed protocols can be seen on Table 14.

Table 14 - Storage cost in bits by entity.

Protocol	$MTCD_{i-j}$	$MTCD_{leader}$	MME
SE-AKA[14]	128 bits	-	432 bits
GBAAM[9]	192 bits	192n bits	-
FU[12]	192bits	-	256n +872 bits
Proposed protocol	192 bits	320n bits	192n+256 bits

From Table 14 it is possible to see that is not a standard to what each entity stores in all the protocols, including the proposed protocol. The storage cost of the proposed protocol is higher than the other studied protocols. Part of this is because the storage cost of the group leader in our protocol is higher, if compared to the others and it can be explained because the leader performs many calculations. It is acceptable, as we are considering that the group leader elected has more resources than the other devices.

## 3.6 PROPOSED PROTOCOL FORMAL VERIFICATION

This section presents the formal verification of the proposed protocol, using AVISPA tool.

### 3.6.1 PROTOCOL SIMULATION

The simulation was based on HLSPL language and following its logic, each entity has a role. The entities roles can be seen on Figure 18, 19, 20 and 21 to a normal MTCD, MTCD leader, MME and HSS. The transitions from a state to another occur at same time messages are exchanged in the proposed protocol.

```

role
role_MTC Dij(MTC Dij:agent,MTCDI:agent,MME:agent,P:text,HSS:agent,IDm:text,IDg:text,TID
m:text,TIDg:text,GK:text,SNDR,RCV:channel(dy))
played_by MTC Dij
def=
    local
        State:nat,Rij:text,MACmme,MACij:text,GTK,SEC1,SEC2:text,Rmme:text,H2:function,
Rhss:text,SKij:symmetric_key
    init
        State := 0
    transition
        1. State=0  $\wedge$  RCV(start)  $\Rightarrow$  State':=1  $\wedge$  secret(IDg',sec_6,{})  $\wedge$ 
secret(IDm',sec_5,{})  $\wedge$  Rij':=new()  $\wedge$  MACij' := H2(mul(P,Rij').IDm.IDg)  $\wedge$  secret(Rij',sec_4,{})
 $\wedge$  SND(mul(P,Rij').MACij'.TIDm)
        6. State=1  $\wedge$  RCV(mul(P,Rmme').Rhss'.MACmme')  $\Rightarrow$  State':=2  $\wedge$ 
secret(GTK',sec_7,{})  $\wedge$  secret(Rmme',sec_3,{})  $\wedge$  SKij' := mul(xor(SEC1',SEC2'),Rij',Rmme',P)
 $\wedge$  secret(SKij',sec_10,{})
    end role

```

Figure 18 – Role of each MTCD in HLSPL.

```

role
role_MTCDI(MTCDij:agent,MTCDI:agent,MME:agent,P:text,HSS:agent,IDg:text,TIDg:text,SEK:te
xt,GK:text,SNDRCV:channel(dy))
played_by MTCDI
def=
    local
        State:nat,LAI,Lh:text,H1:function,TIDm,C:text,MACmme,MACij:text,GTK:text,Rmme:text,
Rhss:text,Ti:text,H2:function,IDm:text,Rij:text
    init
        State := 0
    transition
        1. State=0  $\wedge$  RCV(mul(P,Rij').MACij'.TIDm')  $\Rightarrow$  State':=1  $\wedge$  secret(IDg',sec_6,{})  $\wedge$ 
secret(IDm',sec_5,{})  $\wedge$  secret(Rij',sec_4,{})  $\wedge$  LAI':=new()  $\wedge$  secret(LAI',sec_8,{})  $\wedge$  Lh' :=
H1(LAI',IDg')  $\wedge$  SND(mul(P,Rij').MACij'.Lh'.TIDm'.TIDg')
        5. State=1  $\wedge$  RCV(mul(P,Rmme').Rhss'.MACmme')  $\Rightarrow$  State':=2  $\wedge$ 
secret(GTK',sec_7,{})  $\wedge$  secret(Rmme',sec_3,{})  $\wedge$  Ti':=new()  $\wedge$  secret(IDg',sec_6,{})  $\wedge$ 
secret(IDm',sec_5,{})  $\wedge$  secret(Rij',sec_4,{})  $\wedge$  C' := mul(mul(xor(mul(P,Rij),MACij'),SEK),GK)
 $\wedge$  SND(C'.Ti')

```

Figure 19 – Role of MTCDI<sub>leader</sub> in HLSPL.

```

role
role_MME(MTCDij:agent,MTCDI:agent,MME:agent,P:text,HSS:agent,Key_set_MME_HSS:(symmetric_k
ey) set,Key_set_HSS_MME:(symmetric_key) set,SNDRCV:channel(dy))
played_by MME
def=
    local
        State:nat,TIDg:text,H1:function,TIDm:text,LAI,Lh,SEC1,SEC2:text,SecureChannel,SKij:symmetri
c_key,LMK:text,MACmme:text,GTK:text,Rmme,C:text,Rhss:text,Ti:text,H2:function,IDm:text,Rij:text,ID
g:text,Key_2:symmetric_key,Key_1:symmetric_key
    init
        State := 0
    transition
        2. State=0  $\wedge$  RCV(mul(P,Rij').H2(mul(P,Rij').IDm'.IDg').Lh'.TIDm'.TIDg')  $\Rightarrow$  State':=1  $\wedge$ 
secret(LAI',sec_8,{})  $\wedge$  secret(IDg',sec_6,{})  $\wedge$  secret(IDm',sec_5,{})  $\wedge$  secret(Rij',sec_4,{})  $\wedge$ 
SecureChannel':=new()  $\wedge$  Key_1':=new()  $\wedge$  Key_set_MME_HSS':=cons(Key_1',Key_set_MME_HSS)  $\wedge$ 
SND({ mul(P,Rij').H2(mul(P,Rij').IDm'.IDg').Lh'.TIDm'.TIDg'.LAI'}_SecureChannel'_Key_1')
        4. State=1  $\wedge$  in(Key_2',Key_set_HSS_MME)  $\wedge$ 
RCV({ GTK'.LMK'.Rhss'}_SecureChannel'_Key_2')  $\Rightarrow$  State':=2  $\wedge$ 
Key_set_HSS_MME':=delete(Key_2',Key_set_HSS_MME)  $\wedge$  secret(LMK',sec_9,{})  $\wedge$ 
secret(GTK',sec_7,{})  $\wedge$  MACmme':= H2(mul(P,Rmme'),GTK')  $\wedge$  Rmme':=new()  $\wedge$  secret(Rmme',sec_3,{})
 $\wedge$  SND(mul(P,Rmme').H2(mul(P,Rmme').GTK').Rhss'.MACmme')  $\wedge$ 
SND(mul(P,Rmme').H2(mul(P,Rmme').GTK').Rhss'.MACmme')
        7. State=2  $\wedge$  RCV(C'.Ti')  $\Rightarrow$  State':=3  $\wedge$  SKij' := mul(xor(SEC1',SEC2'),Rij',Rmme',P)  $\wedge$ 
secret(SKij',sec_10,{})  $\wedge$  secret(IDg',sec_6,{})  $\wedge$  secret(IDm',sec_5,{})  $\wedge$  secret(Rij',sec_4,{})
end role

```

Figure 20 – Role of the MME in HLSPL

```

role
role_HSS(MTCDij:agent,MTCDI:agent,MME:agent,P:text,HSS:agent,IDm:text,IDg:text,SEK:text
,GK:text,Key_set_MME_HSS:(symmetric_key)          set,Key_set_HSS_MME:(symmetric_key)
set,SND,RCV:channel(dy))
played_by HSS
def=
    local
        State:nat,TIDg:text,H1,H3:function,H2:function,Rij:text,TIDm:text,LAI,Lh:text,SecureCh
annel:symmetric_key,LMK:text,GTK:text,Rhss:text,Key_2:symmetric_key,Key_1:symmetric_key
    init
        State := 0
    transition
        3.      State=0          ∧          in(Key_1',Key_set_MME_HSS)          ∧
Rcv({ { mul(P,Rij').H2(mul(P,Rij')).IDm.IDg).Lh'.TIDm'.TIDg'.LAI'}_SecureChannel'}_Key_1')
=> State':=1 ∧ Key_set_MME_HSS':=delete(Key_1',Key_set_MME_HSS) ∧ secret(LAI',sec_8,{ })
∧ secret(IDg',sec_6,{ }) ∧ secret(IDm',sec_5,{ }) ∧ secret(Rij',sec_4,{ }) ∧ Rhss':=new() ∧
LMK':=new() ∧ secret(LMK',sec_9,{ }) ∧ GTK':= H3(GK',Rhss') ∧ secret(GTK',sec_7,{ }) ∧
Key_2':=new()          ∧          Key_set_HSS_MME':=cons(Key_2',Key_set_HSS_MME)          ∧
SND({ { GTK'.LMK'.Rhss'}_SecureChannel'}_Key_2')
    end role

```

Figure 21 – Role of the HSS in HLSPL

Figure 22 presents the session role, which describes how a session is established, combining all the entities involved in the authentication procedure and the environment role, which describes the environment where the proposed protocol is executed.

```

role
session1(TIDm:text,TIDg:text,MTCDij:agent,MTCDI:agent,MME:agent,P:text,HSS:agent,IDm:text,I
Dg:text,SEK:text,GK:text,Key_set_MME_HSS:(symmetric_key)
set,Key_set_HSS_MME:(symmetric_key) set)
def=
    local
        SND4,RCV4,SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
    composition
        role_HSS(MTCDij,MTCDI,MME,P,HSS,IDm,IDg,SEK,GK,Key_set_MME_HSS,Key_set_H
SS_MME,SND4,RCV4)
        role_MME(MTCDij,MTCDI,MME,P,HSS,Key_set_MME_HSS,Key_set_HSS_MME,SND3,RCV3)
        role_MTCDI(MTCDij,MTCDI,MME,P,HSS,IDg,TIDg,SEK,GK,SND2,RCV2)
        role_MTCDij(MTCDij,MTCDI,MME,P,HSS,IDm,IDg,TIDm,TIDg,GK,SND1,RCV1)
    end role

role environment()
def=
    const
        hash_0:function,idm:text,p:text,leader:agent,tidg:text,tidm:text,device:agent,mme:agent,hss:ag
ent,idg:text,const_1:text,const_1:text,auth_1:protocol_id,auth_2:protocol_id,sec_3:protocol_id,sec_4:p
rotocol_id,sec_5:protocol_id,sec_6:protocol_id,sec_7:protocol_id,sec_8:protocol_id,sec_9:protocol_id
,sec_10:symmetric_key
        intruder_knowledge = { device,leader,mme,p}
    composition
        session1(tidm,tidg,device,leader,mme,p,hss,idm,idg,const_1,const_1,{ },{ })
    end role

```

Figure 22 – Role specification for the session and environment in HLSPL.



Finally, Figure 23 presents the security goals that the protocol must accomplish. The goals include the mutual authentication between MTC and MME and the secrecy of important parameters, as group temporary key, devices' random number, permanent identities and the session key.

```
goal
  authentication_on auth_1
  authentication_on auth_2
  secrecy_of sec_3
  secrecy_of sec_4
  secrecy_of sec_5
  secrecy_of sec_6
  secrecy_of sec_7
  secrecy_of sec_8
  secrecy_of sec_9
  secrecy_of sec_10
end goal
environment()
```

Figure 23 – Security goals established in HLSP.

### 3.6.2 SECURITY VERIFICATION RESULTS

Two security simulation were executed, OFMC and CL-AtSe. The simulation results showed that the proposed protocol is safe to both checker mechanisms. These results can be seen in Figure 24 and 25.

```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/testedif.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.04s
  visitedNodes: 9 nodes
  depth: 4 plies
```

Figure 24 – Security simulation results for OFMC.

**SUMMARY**

**SAFE**

**DETAILS**

BOUNDED\_NUMBER\_OF\_SESSIONS  
TYPED\_MODEL

**PROTOCOL**

/home/span/span/testsuite/results/testedif.if

**GOAL**

As Specified

**BACKEND**

CL-AtSe

**STATISTICS**

**Analysed** : 17 states  
Reachable : 8 states  
Translation: 0.03 seconds  
Computation: 0.00 seconds

Figure 25 – Security simulation results for CL-AtSe.

A graphical animation was executed, so the messages exchanges could be better visualized. Figure 26 presents the protocol execution on SPAN (Security Protocol Animator for AVISPA), message by message and Figure 27 presents the protocol execution process with the addition of an intruder as another entity.

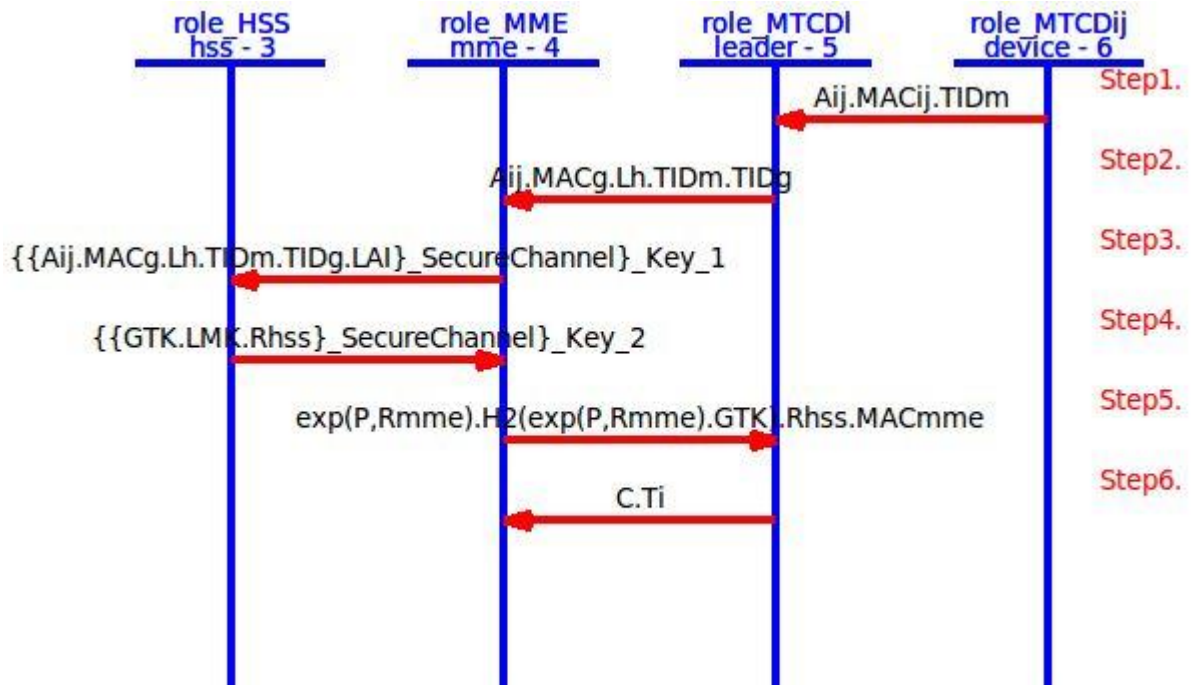


Figure 26 – Protocol’s message exchange in SPAN

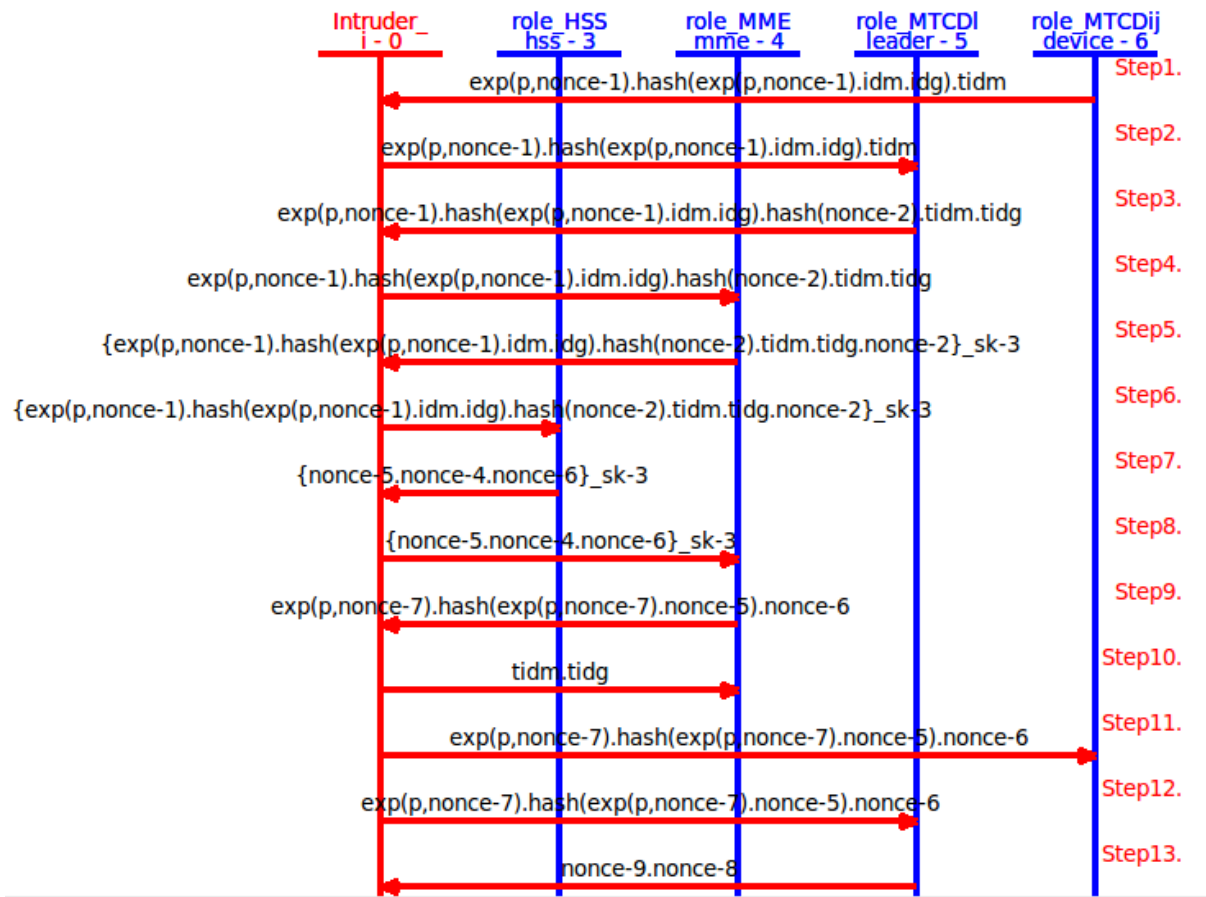


Figure 27 – Intruder’s simulation in SPAN

### 3.7 CONCLUSION

This chapter was destined to the presentation of a new group authentication protocol for MTC, based on ECDH and bilinear pairing. It started with a brief description of some protocols in the group authentication context that already have been proposed by the authors in [9],[12] e [14].

Next, a more detailed description and analysis of the protocols proposed by Cao et al. [9] and FU et al. [12] was presented, additionally exposing some of their security vulnerabilities. Then, the proposed protocol was presented. First, that was a description of its initiation and authentication phases. Next, its security objectives and performance were evaluated and compared with the other protocols described in this chapter. The performance evaluation comprised the computation, communication, verification and storage costs and the comparison of the proposed protocol with the 3GPP standard EPS-AKA [1] and [9], [12] and [14].

The proposed protocol just decreases its computational performance to groups with less than 28 devices. Then, it can be affirmed that it is better in computation cost than all the other protocols listed in the comparison, if groups with more than 28 devices are considered. It presents better communication and verification costs than all the protocols analyzed. Its storage cost is higher than the other protocols, because the group leader perform more operations, which demand some additional parameters to be stored. However, these protocols have some security vulnerabilities that our protocol does not have, as presented in Table 8, compensating the costs. Finally, a formal verification using AVISPA simulation tool was executed proving that the proposed protocol accomplishes the security objectives necessary to a successful group authentication.

In summary, the proposed protocol successfully accomplishes its objectives. It presents excellent results in security and performance, proving itself as a secure and efficient choice when compared to other group authentication protocols described in this section.

# Chapter 4

## AUTHENTICATION AND KEY AGREEMENT PROTOCOL BASED ON SECRET SHARING FOR MACHINE TYPE COMMUNICATIONS

**Resumo:** Este capítulo apresentará uma breve descrição de alguns protocolos propostos para a autenticação de grupos de dispositivos MTC em LTE/LTE-A e a proposta de um novo protocolo de autenticação de grupos baseado no Segredo de Shamir e na fórmula de interpolação de Lagrange. Além disso, são apresentadas uma análise de segurança e comparações entre o protocolo proposto, o protocolo de referência 3GPP EPS-AKA e outros dois protocolos estudados. Serão comparados as propriedades de segurança e os custos computacionais e de comunicação e armazenamento. Finalmente, é apresentada a validação formal do protocolo proposto utilizando o AVISPA, uma ferramenta para simulação de objetivos de segurança de protocolos de autenticação.

**Abstract:** This chapter will present a brief description of some proposed protocols to MTC groups of devices authentication in LTE/LTE-A and the proposal of a new group authentication protocol, based on Shamir's secret and in Lagrange interpolating formula. Besides that, a security analysis and comparisons between the proposed protocol, the 3GPP EPS-AKA reference and other two protocols studied are presented. The comparisons comprise the security properties and the computation, communication, storage and verification costs. Finally, a formal validation is exposed at the end of the chapter, using AVISPA, a simulation tool for security objectives to authentication protocols.

### 4.1 INTRODUCTION

The Internet of Things (IoT) is increasing the number of devices in our daily life and the technology advances are leading to a type of communication without human intervention and defined as Machine Type Communication (MTC). In MTC at least one of the parties is a machine and do not need any human intervention. Several MTC devices collect and send information to a MTC server, where this information will be analyzed. Currently, MTC is being widely used for many applications related to IoT, the main applications are large-scale real time applications, as monitoring, sensing and metering. These applications, generally, receive more attention because of the characteristics of their devices that are low power consumption, low cost and easy installation. Then, to attend the specification of these MTC applications, a good alternative is to use LTE/LTE-A network, because it offers large coverage, high data rates, high throughput, high signal strength and low latency.

The problem using LTE/LTE-A network to support MTC is that the increasing of devices may overload the network with signaling from the authentication and controlling of each device. To solve this problem, it is necessary to group the devices, based on some criteria like same application type, localization, same MTC server and others. Then, instead of authenticating each device separately, the network authenticates all the devices in the MTC group at same time, consequently reducing the signaling traffic. However, the main problem of group authentication in LTE/LTE-A network is its standard authentication protocol, EPS-AKA [1] that does not support group authentication because it was designed to authenticate one device at a time. Rising the necessity of new authentication and key agreement protocols, adapted to group of devices.

Some protocols in the literature already provide group authentication, present some congestion prevention and a concern about safety. However, they have some security issues and their performance still need improvements. Any communication in a public link can be a target of attacks, which is why protection is so important. Protection must be efficient, consuming less bandwidth and computational resources. This characterizes the target of this work: to achieve a protocol robust in security and great in performance.

In this work, we propose a fast authentication and key agreement protocol for MTC groups, based on Shamir's secret and on a binary tree group management, guaranteeing security protection and

performance improvements. The proposed protocol can resist many attacks, with low bandwidth consumption.

The remainder of this chapter is organized as follows: Section 4.2 presents some related and relevant works; Section 4.3 presents the proposed protocol description; Section 4.4 presents its security analysis and comparison to other protocols; Section 4.5 presents its performance analysis and comparison to other protocols; Section 4.6 presents the formal verification using AVISPA tool and finally, Section 4.7 presents the conclusion.

## 4.2 RELATED WORK

Through the group authentication development, complex and robust protocols emerged for MTC in LTE/LTE-A, with higher security protection and greater performance, bringing innovations in this field. Among these protocols, we selected the following references:

- Harn [16] – Group Authentication;
- Lai et al. [13] - GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications;
- Li et al. [15] - Group-Based Authentication and Key Agreement with Dynamic Policy Updating for MTC in LTE-A Networks;
- Choi et al. [11] - A group-based security protocol for machine-type communications in LTE-advanced.

The criteria adopted for the selection of papers were the employment of symmetric cryptography and challenge-response. As well as good performance and the publication in periodicals or events of good quality. Additionally, [11] was selected for a more detailed analysis because it presents the binary tree group organization adopted in this work.

### 4.2.1 SHAMIR'S SECRET BASED SCHEMES PROPOSED BY HARN [16]

Harn [16] published a contribution to group authentication, based on Shamir's secret. The Shamir's secret is a scheme based on polynomial and Lagrange interpolating formula. The scheme allows a group manager to generate a secret token, based on random polynomial, to each member of the group, where all tokens have a secret value in common. Therefore, in possession of these tokens, all the members of a group can authenticate each other, reconstructing the secret value through the Lagrange interpolating formula. Only if all the members are legitimate, that is, only if all the members have legitimate tokens, the right secret will be reconstructed through the Lagrange interpolating formula. A brief summary of the Shamir's secret is showed in Figure 28:

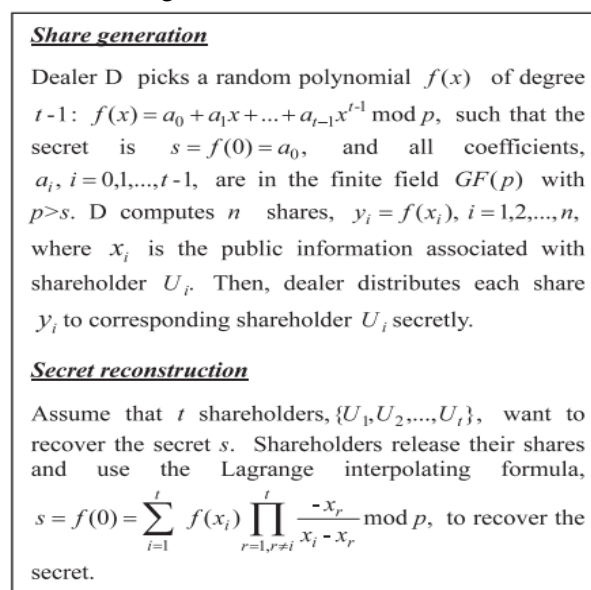


Figure 28 –Shamir's Secret in Harn scheme (source: [16]).

In his paper, Harn [16] uses Shamir's scheme to propose (t,m,n) GAS. This scheme guarantees authentication to  $m$  members of a group with  $n$  group members and it is resilient until  $t$  tokens are compromised. The idea is like Shamir's, but each member's token remains secret. The Group manager generates  $l$  tokens for each member, where each token is constructed by a different polynomial and all polynomial have the same secret. Each member generates a Lagrange component, based on the token it received. The Lagrange component is based on the Lagrange interpolating formula and even if someone knows the Lagrange component, it will be impossible to recover its correspondent secret token. Then, each member send its own component to other members and only if all the Lagrange components are legitimate, their sum will generate the secret value. It is a scheme that allows members of the same group to authenticate each other without the participation of the group manager. A summary of the scheme is presented in Figure 29:

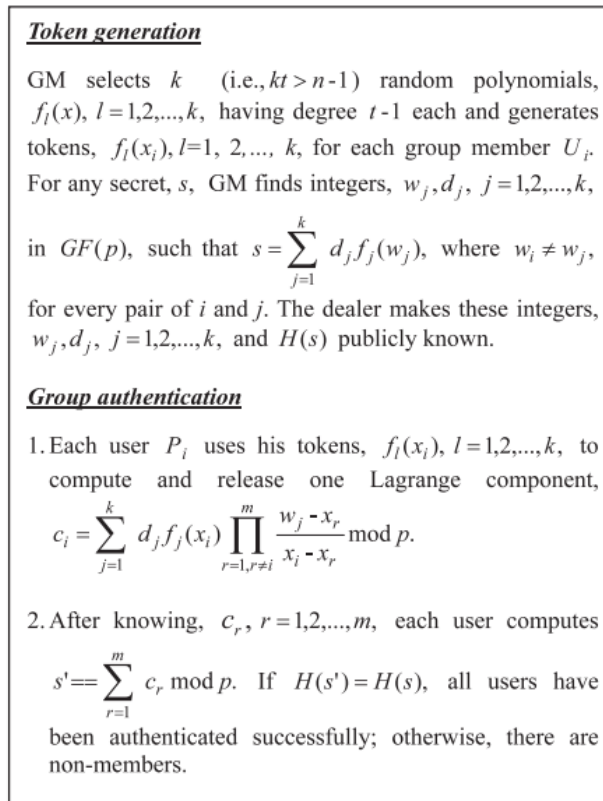


Figure 29 – (t,m,n) GAS in Harn scheme (source: [16]).

#### 4.2.2 GROUP AUTHENTICATION PROTOCOL DEVELOPED BY LAI ET AL. [13]

Lai et al. [13] created a protocol totally based on symmetric keys and hash functions, which provide mutual and fast group authentication and key agreement, named GLARM. The architecture is similar to 3GPP's architecture. The scheme consists of two phases: Initialization and group authentication and key agreement. Table 15 describes the entities involved in the authentication procedure and Table 16 presents the notations used.

Table 15 – Main entities involved in the architecture of Lai et al. [13] protocol

Abbreviation	Entity
MTCD <sub>Gi-j</sub>	Mobile Terminal Communication Device $j$ of group $i$
MTCD <sub>leader</sub>	Mobile Terminal Communication Device's group leader
HSS	Home Subscriber Server
MME	Mobile Management Entity

Table 16 - Notations used by Lai et al. [13]

Notation	Definition
$h_1(.) - h_5(.)$	Authentication and key generation function
<b>ID<sub>x</sub></b>	Identity of x
<b>TID<sub>i</sub></b>	Temporary identity of x
<b>GK<sub>i</sub></b>	Group key of the i-th group
<b>LAI</b>	Location Area Identification
<b>GTK<sub>i</sub></b>	Group temporary key of the i-th group
<b>IK</b>	Integrity key
<b>CK</b>	Cipher key
<b>AK</b>	Authentication key
<b>MSK</b>	Master session key
<b>MAC<sub>x</sub></b>	Message authentication code computed by x
<b>XRES<sub>x</sub></b>	Expected authentication response computed by x
<b>RES<sub>x</sub></b>	Authentication response computed by x
<b>AMF</b>	Authentication Management Field
<b>AUTH<sub>x</sub></b>	Authentication token generated by x

**Initialization phase:** It is considered that each device has a symmetric key pre-shared with HSS and the link between HSS and MME is secure. The groups are formed, based on common characteristics, receive a group key, an identifier and chose a group leader. A group key  $GK_i$  and a group identity  $ID_{gi}$  is provided by the supplier.

**Group authentication and key agreement phase:** GLARM's authentication phase can be seen in Figure 30 and it is described below:

1. When the process of authentication begins, each device calculates its MAC, based on its ID, and send it to the group leader.
2. The leader aggregates the device's MACs in one single group MAC and send it to MME.
3. The MME adds the LAI of the group's base station, to avoid redirection attack and forward the message to HSS.
4. The HSS verifies the LAI and the group's MAC received. If the verification passes, the HSS generates a temporary group key, GTK, its own MAC, an expected answer from the group, XRES, and the session key between the device and MME. Then, it sends all these parameters to MME.
5. The MME generates its MAC and send to the devices. Each device authenticates the MME verifying the MME's MAC. Then, each one calculates a session key among itself and the MME, calculates the expected answer, RES, and send it to the group leader.

6. The leader aggregates all the RES received and send to MME.
7. The MME just compares the RES received from the group with XRES received from HSS. If the verification passes, the group is authenticated and the MME calculates a session key between each device and itself.

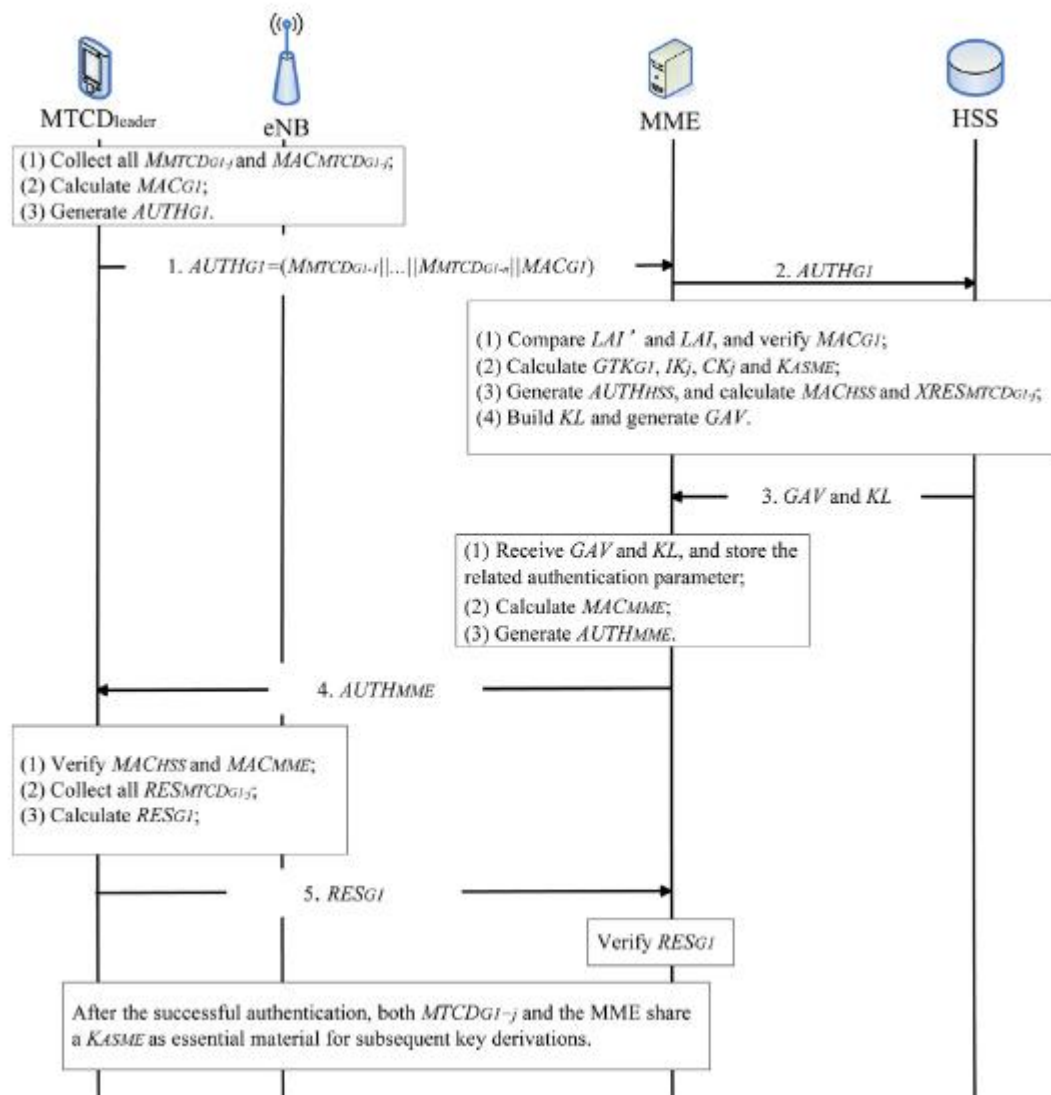


Figure 30 – Mutual authentication phase in Lai et al (source: [13]).

One of the security lacks of Lai et al. [13] is the devices and group identities that are not protected. They are sent in plaintext, so an attacker can discover their identities and target attacks to them.

#### 4.2.3 GROUP AUTHENTICATION PROTOCOL DEVELOPED BY LI ET AL. [15]

Li et al. [15] developed a group authentication protocol based on Shamir's secret, named GR-AKA. Table 15 presents the main entities involved in the protocol and Table 16 shows the notations used by GR-AKA protocol.

Its architecture is similar to 3GPP EPS-AKA [1], with the addition of a MTC Server that can be located inside or outside the LTE architecture. The KGC is integrated with the HSS. In this protocol, the HSS, MME and MTC Dleader are considered trusted and only MTC Ds and base stations can perform attacks. The protocol has two phases: Preparation and Register Phase and group-based access authentication phase. Table 17 presents the main entities involved in the scheme and Table 18 presents the notations used.



Table 17 - Main entities involved in the architecture of Li et al. [15] protocol

Abbreviation	Entity
MTCD	Mobile Terminal Communication Device
Group leader	Mobile Terminal Communication Device's group leader
HSS	Home Subscriber Server
MME	Mobile Management Entity

Table 18 - Notations used by Li et al. [15]

Notation	Definition
IDx	Identity of x
IDtx	Temporary identity of x
Tsx	Timestamp generated by x
Sk <sub>x-y</sub>	Session Key among x and y
POS	Location information of a base station
MACx	Message authentication code computed by x

**Preparation and Register Phase:** In this phase the parameters and keys are initialized. The KGC assign permanent identities (IDx) and temporary identities (TID) to all MTCDs and all MTC users. Then, the MTCDs form groups based on common characteristics. After the group is defined, the KGC establishes a secret message authentication  $S$  to each group and a leader is chosen. The leader is considered a trusted entity. Next, the KGC defines a hash function  $H()$ , a random prime number  $p$ , a finite field  $GF(p)$  and a random number  $l$ . Then, based on (t,m,n) GAS scheme, KGC generates  $l$  random polynomial functions  $f_\theta(x)$  of degree  $t-1$ .

$$f_\theta(x) = \sum_{i=0}^{t-1} a_{\theta i} x^i \text{ mod } p \quad (4.1)$$

Where  $1 \leq \theta \leq l$  and  $l * t > n-1$ . So, the secret  $S$  is defined,  $S = f_\theta(x) = a_0$ . All coefficients  $a_\theta$  are part of the finite field  $GF(p)$ . After that, KGC finds  $l$  pairs of integer  $w_\theta$  and  $d_\theta$  in  $GF(p)$ , where following condition is satisfied:

$$S = \sum_{\theta=1}^l d_\theta f_\theta(w_\theta) \quad (4.2)$$

When the devices are registered in network, each of them receive  $l$  tokens  $f_\theta(ID_{MTCDi})$ , where  $\theta=1,2,\dots,l$ .

Then, the KGC defines two groups, G1 and G2, with P as the generating point. It also chooses a random number  $r$ , calculates its own public key  $PK = r * P$  and a private key  $k_{MTCDi} = r * H(ID_{MTCDi})$  to each device. Then, it generates a private key  $k_{MME} = r * H(ID_{MME})$  to the MME, and establishes a  $\Delta t$  to each entity as the timestamp of each message. Finally, the KGC publishes:  $\{p, G1, G2, e, P, PK, H(S), w_\theta, d_\theta\}$ .

**Group-based Authentication and Key Agreement phase:** This phase has the objective of perform the authentication between each of the devices and the HSS, establishing a session key among them at the end of the process. The message exchange is presented in Figure 31 and described below:

1. Each device request access to MME through the group leader;
2. The MME sends an identity request to the leader. Then, each device calculates its own Lagrange component and its public key. Next, they all send these two parameters to the leader.

3. The leader verifies the devices, recalculating the secret  $S$  and comparing with  $H(S)$  published by KGC. If the devices are legitimate, then the leader makes a group MAC and send it to the MME.
4. The MME adds a location information to the message, in order to avoid redirection attack. Then, it sends the message to the HSS.
5. The HSS receives the message and checks the location information and the MAC received from the MTC group. If the verification passes, the MTC group is authenticated. Then, HSS calculates its own Lagrange component, creates its own MAC and send these parameters to the MME.
6. When the MME receives the message, it considers that the group is authentic, calculates its own public key, PK and send the message received from HSS, plus PK to the group leader.
7. The leader receives the HSS's Lagrange component, recalculates the secret  $S$  and verify if it is equal to the secret calculated previously. If the verification passes, the HSS is authenticated. Next, the leader sends the necessary information to calculate the session key to the MTCs and to the MME.

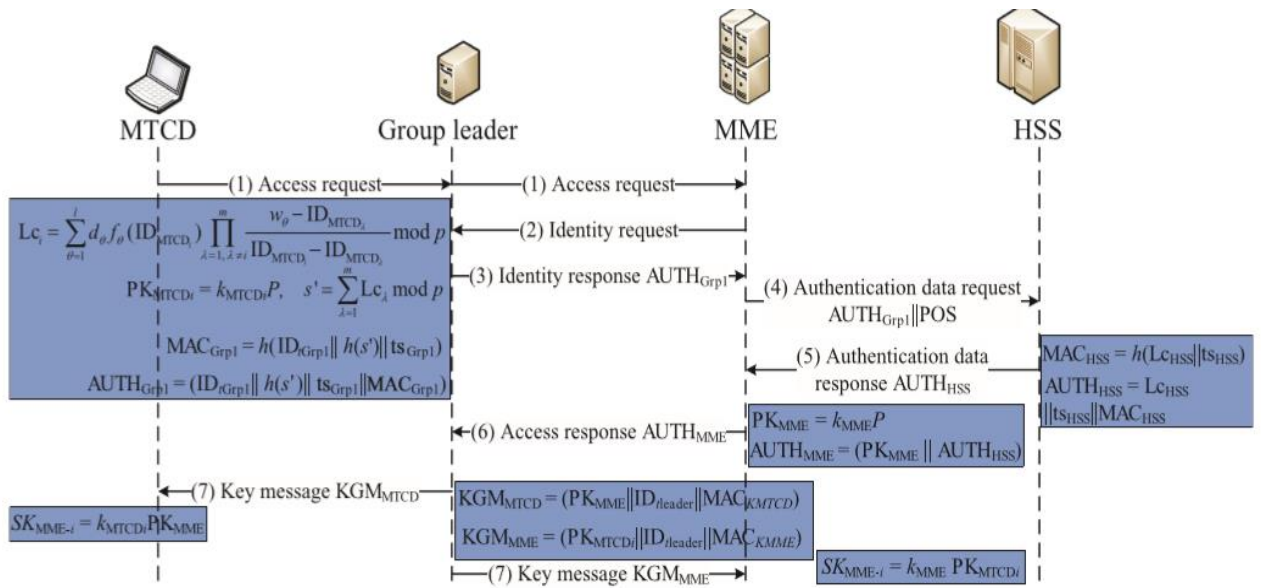


Figure 31 – Group-based Authentication and Agreement phase in Li et al. (source: [15]).

Li et al. [15] do not guarantee the anonymity of the MTC group and do not guarantee privacy in device's identities from other devices in the same group. The group's identity is sent in plaintext, providing to the attacker the opportunity to track and identify the groups involved in the authentication procedure. Devices in the same group know each other's real identities, so if a device leave the group it continues knowing all the other device's identities and can aim attacks to them. This issue also weakens the group  $MAC_{G1}$ , because it is made based on following parameters:

$$MAC_{G1} = H(ID_{G1} || H(S) || ts_{G1}) \quad (4.3)$$

Where  $ID_{G1}$  is the group's identity, the  $H(S)$  is the hash of the secret and  $ts_{G1}$  is a timestamp.  $ID_{G1}$  and  $H(S)$  are public parameters, that is, anyone can know these values. This means that the security of  $MAC_{G1}$  lies over the timestamp  $ts_{G1}$ , and anyone can try to impersonate the devices to obtain authentication or to get some important information.

This lack in  $MAC_{G1}$  also gives opportunity to the Denial of Service attack (DoS), because an attacker knows that  $ID_{G1}$  is a valid group identity and knows  $H(S)$  too, so it can generate many MACs with these two parameters, varying only the timestamp. It is obvious that HSS can quickly verify these timestamps, but all those MACs that are within the time interval will initiate an authentication process and the attacker just need to find one value in this interval to start the DoS attack.

#### 4.2.4 OTHER GROUP AUTHENTICATION PROTOCOLS

Choi et al. [11] protocol is based on symmetric cryptography and do the management of the group of devices through a binary tree, where each node is associated to a secret value derived of its parents. Each device is authenticated simultaneously with the group leader and a different session key among the MME and each of them is established. The session key is based on the secret values of the common nodes between each device and the MME and in a random number generated by the HSS in the authentication procedure. Its main contribution to this work lays on the binary tree management configuration that is used by our proposed protocol and better described in section 4.4.

Other protocols can be found in the literature. Among such protocols, we can mention for example Chen et al. [10].

#### 4.2.5 COMPARATIVE EVALUATION

Important properties are compared of the protocols described previously are described in this section and are present in Table 19. The schemes proposed by Harn [16] are not considered in this comparison because it is a descriptive and not a performance evaluation paper. Some of the properties are not considered to [1] because it is not a group protocol. The MTC server is the entity responsible to manage the MTC devices and may be located inside or outside the EPC and the group management field presents how the MTC server and HSS perform this management.

Table 19 – Comparative evaluation of the described protocol.

	EPS-AKA[1]	Lai et al. [13]	Li et al. [15]	Choi et al. [11]
<b>Group Authentication</b>	No	Yes	Yes	Yes
<b>Type of Cryptography</b>	Symmetric	Symmetric	Asymmetric	Symmetric
<b>MTC Server</b>	-	Inside EPC	Both	Outside EPC
<b>Leader Election</b>	-	Yes	Yes	Yes
<b>Group Management</b>	-	Table	Not Mentioned	Binary Tree
<b>Shamir's Secret</b>	No	No	Yes	No
<b>Location Area Identification (use of LAI)</b>	No	Yes	No	No

#### 4.3 PROPOSED PROTOCOL

A new group protocol based on symmetric cryptography, Shamir's secret and Lagrange's interpolating formula is presented in this section. Its objective is to provide secure and efficient authentication and key agreement to large groups of devices, with a good performance. This protocol will be compared to [11], [13] and [15].

The proposed protocol is composed of two phases: Registration and Authentication and Key Agreement. The Registration phase is performed first and it is responsible to provide important parameters used in the Authentication and Key Agreement phase. The following basic assumptions are considered:

1. The KGC is a trustful authority integrated with the HSS.
2. Secure channel between MME and HSS.
3. The MTC Server is outside the EPC.

The network architecture is presented in Figure 32 and it is adapted from the 3GPP [1] standards. Table 20 defines the notations used in the proposed protocol.

Table 20 - The main entities involved in the proposed protocol architecture

Abbreviation	Entity
MTCD <sub>i-j</sub>	Mobile Terminal Communication Device <i>j</i> of group <i>i</i>
MTCD <sub>leader</sub>	Mobile Terminal Communication Device's group leader
HSS	Home Subscriber Server
MME	Mobile Management Entity
eNB	Evolved Node B

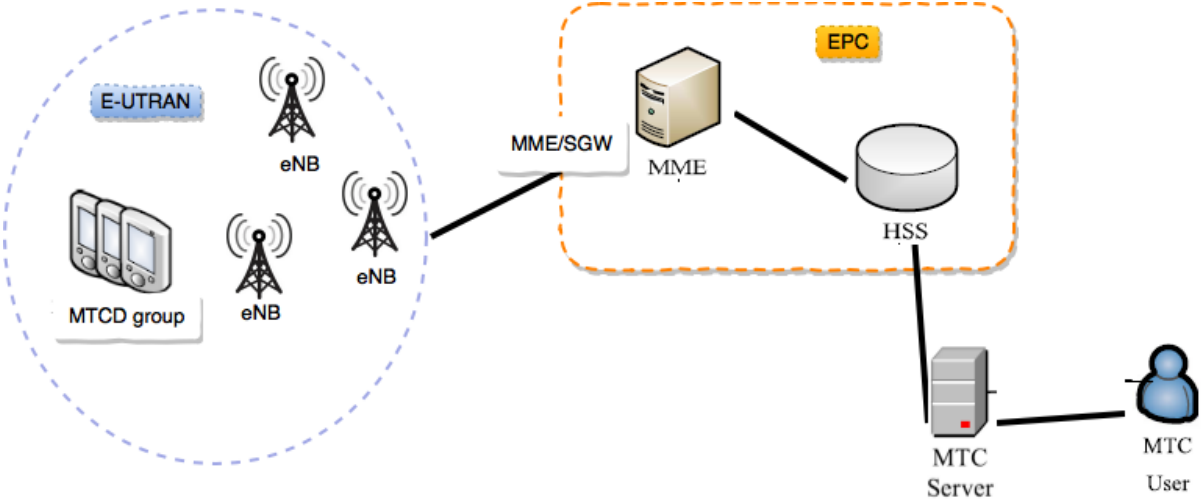




Figure 32 – The network architecture adopted.

In the proposed protocol, we use the Asynchronous (t; m; n) Group Authentication Scheme (GAS) scheme from Harn [16] to perform the group authentication, which is based on Shamir's secret [22]. The scheme proposed by Shamir [22] was chosen because it is a quick and effective way to share and rebuild a secret with many parts, which when associated to GAS, proposed by Harn [16], can provide mutual authentication to a group of devices.

The (t,m,n) GAS guarantees group authentication for m devices of a group with n members and it is resistant to t compromised tokens. In our protocol, it is considered that m has the same size of n, that is, all the members in a group are authenticated. Thus, it authenticates all the devices in a group simultaneously. Additionally, the scheme of dynamic secret update presented by Li et al. [15] is adopted to provide secret update when devices enter or leave the group. The group organization and management is the same by Choi et al. [11] and was adopted because it facilitates the group and devices management.

Table 21 - Notations used in the proposed protocol

Notation	Definition
$x/PK$	Private/Public key of KGC
$ID_a, TID_a$	Identity and Temporary Identity of entity $a$
$LAI$	Location Area Identification
$GK_i, GTK_i$	Group key / Group temporary key
$MAC_a$	Message Authentication Code of entity $a$
$r_a$	Random number generated by entity $a$
$LC_a$	Lagrange component of entity $a$
$S$	Shamir's secret among devices and MME
$f(x)$	random polynomial function of degree $t-1$
$SEK_{i-j}$	Secret key shared between $MTCD_{i-j}$ and HSS
$SEC_y$	Secret value of node $y$
$h_1(.)$	Secure hash function
$h_2(.)$	Message authentication hash function
$h_3(.)$	Key generation hash function
$h_4(.)$	Session key hash function
$\parallel$	Concatenation operation
$\oplus$	XOR operation
	Secure channel
	Insecure channel

#### 4.3.1 REGISTRATION PHASE

The registration phase establishes and configures all the parameters that will be necessary to the MTCD groups be authenticated by the network. This phase occurs over a secure channel.

The HSS generates  $z$  random numbers,  $R_z \in \mathbb{Z}_p^*$ , ( $z = 1, 2, \dots, i$ ) and calculates a set of temporary identities  $TID_{MTCD_{i,j}}$  to each  $MTCD_{i,j}$ , as it follows:

$$TID_z = h_1(ID_{MTDCDi} \parallel R_z * x), \text{ where } x \text{ is the secret key of HSS.} \quad (4.4)$$

Then, the devices storage each  $TID_z$  related with its respective  $R_z$ . A different TID is used every time an authentication and key agreement procedure is executed.

After that, the MTCDs form a group based on common characteristics and a group leader is elected. Common procedures for leader election are defined, for example, in Abbasi et al. [28] and Chatterjee et al. [29]. However, it is not our objective to discuss this question. Some of the devices characteristics used to group definition could be localization, type of application and if they managed by the same MTC server. The criteria used to select the group leader could be higher storage capacity, longer battery, higher computational power, higher communication capacity, for example.

Next, the HSS creates a binary tree, as described in Choi et al. [11], to organize each MTC group registered in the network. Each device is placed in an empty leaf. Each node of the tree has a secret that is defined by HSS. The devices know all these secrets, except those that form a path between the device and the root of the tree. Choi's binary tree is presented in Figure 33.

The HSS defines a group identity  $ID_{Gi}$  and temporary group identity  $TID_{Gi}$ . Then, it generates a random number  $g$  and calculates the group key,  $GK$ :

$$GK_i = h_3(SEC_{i-1} \oplus SEC_{i-2} \oplus \dots \oplus SEC_{i-j} \oplus g * x) \quad (4.5)$$

Next, the HSS selects and publishes three hash functions  $h_1(.)$ ,  $h_2(.)$  and  $h_3(.)$ . Then, the KGC chooses a random prime number  $p$  and defines a finite field  $GF(p)$ . After that, it generates an authentication message  $S$ , which is a secret parameter that will be essential to perform group authentication, and selects a random polynomial function  $f(x)$  of degree  $t-1$  to each group, where  $t \leq n$ , and also is the number of tokens necessary to recover the secret  $S$ . The polynomial function is described as following:

$$f(x) = \sum_{i=0}^{t-1} a_i x^i \text{ mod } p \quad (4.6)$$

And the secret  $S$  is:

$$S = f(0) = a_0 \quad (4.7)$$

All coefficients  $a_i$  are in the finite field  $GF(p)$ . The KGC guarantees that the following condition is achieved:

$$S = \sum_{c=1}^n f(x_k) \prod_{q=1; q \neq c}^n \frac{-x_q}{x_c - x_q} \text{ mod } p \quad (4.8)$$

Then, the KGC generates  $k$  tokens  $f(TID_{l_{MTCDi}})$  to each device, where  $l = 1, 2, \dots, k$ , one token for each TID that a device have. The devices storage their  $k$  tokens with the respective TIDs. The tokens must remain secret to any device that is outside the group and will be used to authenticate the devices in the authentication phase.

Finally, the KGC calculates and publishes the hash of secret  $S$ ,  $H(S)$ , and the hash function  $H()$ , that will be used to verify if all devices in group are valid.

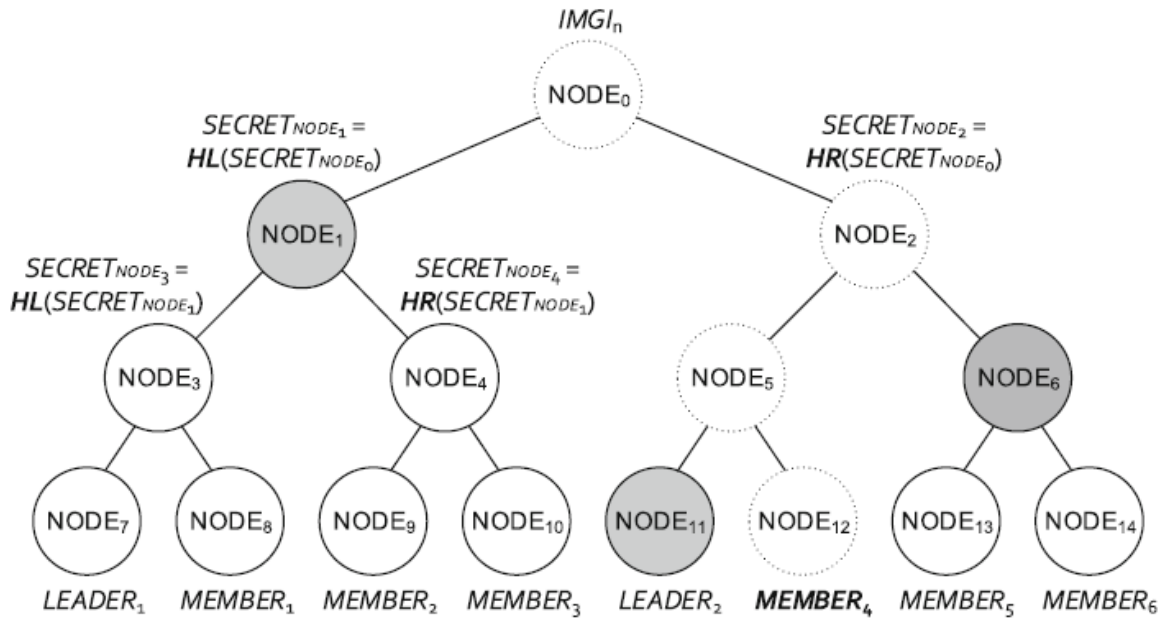


Figure 33 – Binary tree presented for group organization (source: [11]).

### 4.3.2 AUTHENTICATION AND KEY AGREEMENT PHASE

This phase begins when a MTCD group enters an eNB coverage area and wants to access the network. The protocol proceeds as follows:

1.  $MTCD_{i-j} \xrightarrow{(TID_{MTCD_{i-j}})} MTCD_{i-j}$

Each device chooses a non-used  $TID_{MTCD_{i-j}}$  with its respective associated token,  $f(TID_{MTCD_{i-j}})$ . Then, broadcasts its own  $TID_{MTCD_{i-j}}$  to the other devices in the group, enabling them to calculate their Lagrange component  $LC_{i-j}$ .

2.  $MTCD_{i-j} \xrightarrow{(LC_{MTCD_{i-j}})} MTCD_{i-j}$

Each  $MTCD_{i-j}$  computes a Lagrange component,  $LC_{i-j}$ , using the selected token  $f(TID_{MTCD_{i-j}})$  received from KGC, through the Lagrange interpolating formula:

$$LC_{MTCDi-j} = f(TID_{MTCDi-j}) \prod_{q=1; q \neq i}^n \frac{-TID_{MTCDi-q}}{TID_{MTCDi-j} - TID_{MTCDi-q}} \text{ mod } p \quad (4.9)$$

They use the  $TID_{MTCDi-j}$  received from the other devices in the group to generate a valid Lagrange component. Then, MTCDs broadcast their  $LC_{MTCDi-j}$  to all the group members.

### Group's members authenticate themselves

After receiving the Lagrange components from the other group members, each  $MTCD_{i-j}$  verify if they all are legitimate devices and, consequently, if the group is legit. The verification is made calculating a secret  $S$  and comparing the value found with the value published by the KGC in the registration phase,  $H(S)$ .

$$S' = \sum_{j=1}^n LC_{MTCDi-j} \text{ mod } p \quad (4.10)$$

If  $H(S') = H(S)$ , all the devices are validated and considered legit. If the verification fail, the group has one or more intruders and the process of authentication fails. Then, the process only continues if all devices are legitimate and verified.

3.  $MTCD_{leader} \xrightarrow{(AUTH_{Gi}, TID_{MTCDi-1}, \dots, TID_{MTCDi-j})}$  MME

Then, the  $MTCD_{leader}$  generates the group's  $MAC_{Gi}$  and  $AUTH_{Gi}$ :

$$MAC_{Gi} = h_2(GK \parallel ID_{Gi} \parallel LAI \parallel S') \quad (4.11)$$

$$AUTH_{Gi} = (TID_{Gi} \parallel MAC_{Gi}) \quad (4.12)$$

$MAC_{Gi}$  is based on GK and  $ID_{Gi}$ , which are parameters only known by valid members of the group. It is also based on the group secret  $S'$ , proving the group's legitimacy if  $S'$  is equal to the original secret  $S$ , generated by the KGC in the registration phase. Additionally, it is based on LAI that is an identifier related to the group's legit base station. Then the  $MTCD_{leader}$  sends  $(AUTH_{Gi} \parallel TID_{MTCDi-1} \parallel \dots \parallel TID_{MTCDi-n})$  to MME.

4. MME  $\xrightarrow{(AUTH_{Gi}, LAI)}$  HSS

The MME knows which is the LAI' associated to the group's and adds it to the message, so the HSS can verify if the LAI provided by the group leader is legit. MME storages each device's  $TID_{MTCDi-j}$  for future use. Then MME sends  $AUTH_{Gi} \parallel LAI'$  to HSS.

5. HSS  $\xrightarrow{(f(ID_{MME}), r_{HSS}, GTK)}$  MME

After receiving the message from MME, the HSS associates the group temporary identity,  $TID_{Gi}$  to its permanent identity,  $ID_{Gi}$  and group key GK. Then, it uses GK,  $ID_{Gi}$ , with LAI and  $S'$  received from MME to calculate  $MAC_{Gi}'$ :

$$MAC_{Gi}' = h_2(GK \parallel ID_{Gi} \parallel LAI \parallel S) \quad (4.13)$$

If  $MAC_{Gi}'$  calculated is equal to  $MAC_{Gi}$  received from MME, the MTCD group is authenticated by the HSS. If not, a failure message is sent to the  $MTCD_{leader}$ .

Then, HSS chooses a random number  $r_{HSS}$  and generates the temporary group key GTK:

$$GTK_{Gi} = h_3(GK \parallel r_{HSS}) \quad (4.14)$$

Next, HSS calculates a token to MME,  $f(ID_{MME})$ , using MME's identity,  $ID_{MME}$ . This token will enable the devices to authenticate the MME in the future. In the end, HSS sends  $f(ID_{MME}) \parallel GTK_{Gi} \parallel r_{HSS}$  to MME.

6. MME  $\xrightarrow{(AUTH_{MME})}$   $MTCD_{i-j}$

After receiving the message from HSS, the MME generates a random number,  $r_{MME}$ , and execute an XOR's operation of  $r_{MME}$  and GTK. Thus, only someone who know GTK will be

capable to recover  $r_{MME}$ , that is, just a legitimate device can recover  $r_{MME}$ . Next, MME calculates its own Lagrange component and  $AUTH_{MME}$ :

$$LC_{MME} = f(ID_{MME}) \prod_{q=1}^n \frac{-TID_{MTCDi-q}}{ID_{MME} - TID_{MTCDi-q}} \text{ mod } p \quad (4.15)$$

$$AUTH_{MME} = (LC_{MME} \| r_{MME} \oplus GTK \| r_{HSS} \| ID_{MME}) \quad (4.16)$$

Then, MME broadcasts  $AUTH_{MME}$  to all the  $MTCD_{i-j}$ .

7.  $MTCD_{i-j} \xrightarrow{(LC_{MTCDi-j}, r_{MTCDi-j} \oplus GTK)} MME/MTCD_{i-j}$

When each device receives the message from MME, the devices first need to update its Lagrange component with the MME's identity,  $ID_{MME}$ , in following way:

$$LC_{new_{MTCDi-j}} = LC_{MTCDi-j} * \left( \frac{-ID_{MME}}{TID_{MTCDi-j} - ID_{MME}} \right) \quad (4.17)$$

Next, each device get  $r_{HSS}$  and calculate GTK:

$$GTK_{Gi} = h_3(GK \| r_{HSS}) \quad (4.18)$$

A new group temporary key is generated at each session. After updating the Lagrange component and calculating GTK, each MTCD recovers  $r_{MME}$  executing an XOR operation with GTK. Then, chooses a random number  $r_{MTCDi-j}$  and performs an XOR operation with GTK to keep this value secret.

Finally, the devices broadcast the new Lagrange component and the random number,  $LC_{MTCDi-j} \| r_{MTCDi-j} \oplus GTK$ , to all the group members and to MME.

8.  $MTCD_{i-j} \xrightarrow{\text{Success/Failure}} MME$

When each device receives all the new Lagrange components from other group members, it can authenticate the MME, recalculating the secret  $S$  with the Lagrange component of MME:

$$S'' = \left( \sum_{j=1}^n LC_{MTCDi-j} + LC_{MME} \right) \text{ mod } p \quad (4.19)$$

If  $S''$  calculated is equal to  $S'$  calculated before, the MME is authenticated by the devices and each of them sends it a success message. If the verification fails, each device that detects an authentication failure sends it a failure message.

9.  $MME \xrightarrow{\text{Success/Failure}} MTCD_{i-j}$

When MME receives from each  $MTCD_{i-j}$  their Lagrange components,  $LC_{i-j}$ , it verifies them calculating the secret  $S'$ :

$$S' = \left( \sum_{j=1}^n LC_{i-j} + LC_{MME} \right) \text{ mod } p \quad (4.20)$$

If  $H(S')$  is equal to  $H(S)$  published by KGC, the devices are authenticated by MME and it sends a success message to the group of  $MTCD_{i-j}$ . If the verification fails, it sends them a failure message. Finally, the authentication procedure is over.

If the mutual authentication procedure is successful, MME will integrate the binary tree as a new element, as presented in Figure 8. Then, each  $MTCD_{i-j}$  calculates a session key shared among itself and MME. The MME also calculates a session key shared among itself and each  $MTCD_{i-j}$ . The session key,  $SK_{i-j-MME}$ , is calculated as follows:

$$SK_{i-j-MME} = h_4(SEC_a \oplus SEC_b \oplus \dots \oplus SEC_z \| r_{MTCDi-j} \| r_{MME} \| S) \quad (4.21)$$



Where  $SEC_a, SEC_b \dots SEC_z$  are the secrets of the nodes that each  $MTCD_{i-j}$  and MME have in common, grey circles in Figure 35. This model of session key is based on the binary tree presented by Choi et al.[11] and also can be used to device-to-device communication (D2D) between all the  $MTCD_{i-j}$ . A different session key is generated at the end of each session performed by the group. The session key and all the other keys generated must be refreshed based on the security policy applied by the responsible company or carrier.

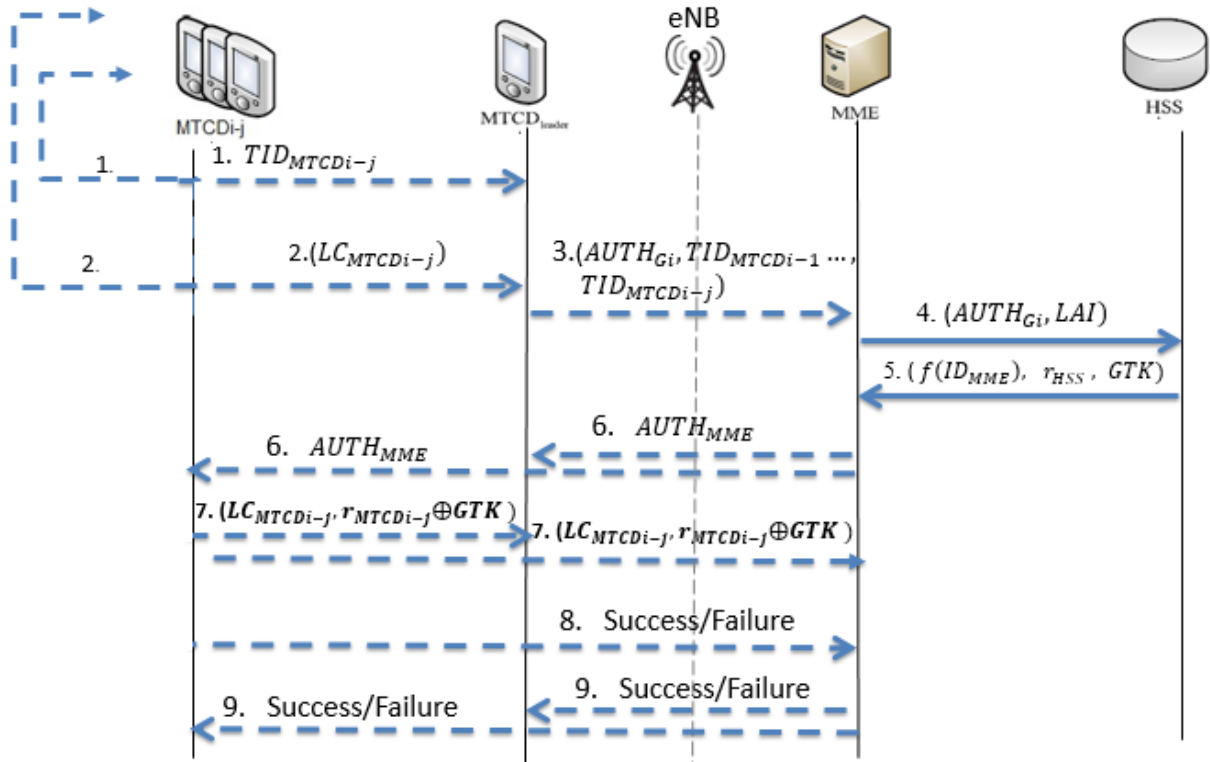


Figure 34 – The authentication and key agreement phase of the proposed protocol.

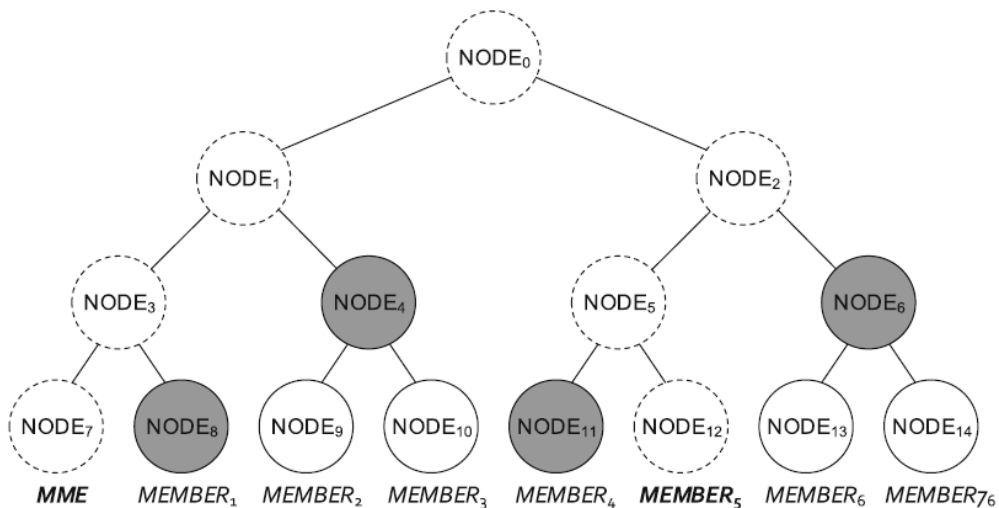


Figure 35 – The binary tree after the entrance of MME (source [11]).

Additionally, the group key needs an update with MME node's secret value, because it is now part of the tree, and considered a group member. The procedure is the same as described in session 4.3.3.

### 4.3.3 GROUP KEY UPDATE

The group key must be updated every time a member enters or leaves the group of devices. In our protocol, we adopt Choi's binary tree scheme [11] and the update of GK is based on it, as the following sections explain.

#### 4.3.3.1 MEMBERS JOINING THE GROUP

As in chapter 3, when a new member joins the group, the current  $GK_i$  needs an update so the new member does not have access to the messages exchanged before its entrance. This procedure is shown on the equation below and is necessary to guarantee the backward secrecy.

$$GK'_i = h_3(GK_i \oplus SEC_{i-y}) \quad (4.22)$$

Where  $GK'_i$  is the new group key and  $SEC_{i-y}$  is the secret value of the node where the new member is located. The new member is not able to discover the old  $GK_i$  because it does not know its own  $SEC_{i-y}$  and cannot revert the hash function.

#### 4.3.3.2 MEMBERS LEAVING THE GROUP

As in chapter 3, there is also the possibility that a device needs to leave the group. This may be justified by many reasons, as the loss of the common characteristics that are the requisites to be part of the group or simple when finished their tasks. This includes the MME that can be changed at any time, as the group interest changes. The procedure is shown on the equation below and is necessary to guarantee the forward secrecy.

$$GK''_i = GK_i \oplus SEC_{i-y} \quad (4.23)$$

In this case, the hash function is not necessary because of the hash function executed in the joining procedure, which is enough to prevent the reversal to old group keys. The member leaving the group cannot discover the group key because it does not know its own  $SEC_{i-y}$ .

### 4.3.4 GROUP SECRET UPDATE

In our protocol, the secret  $S$  is an important parameter, because the group authentication depends on it. The group may have the right  $S$  or not, so this parameter must remain secret to devices that do not integrate the current group. This scheme of secret update was taken from Li et al. [15].

#### 4.3.4.1 MEMBERS JOINING/LEAVING THE GROUP

When a MTCN joins or leaves the group, the secret  $S$  must be updated to avoid the old member to continue knowing the secret and to avoid new members to discover the last secret values  $S$ . Then, when a MTCN leaves or join a group, the HSS generates a new secret as follows:

$$S_{new} = S + \Delta S \quad (4.24)$$

Where  $\Delta S$  is a random value generated every time the secret  $S$  is updated. Then, the HSS sends the new term,  $\Delta S$ , secretly to MME. The MME encrypts  $\Delta S$  with  $SK_{i-j-MME}$  and sends it to each device of the group.

$$En_{SK_{i-j-MME}}[\Delta S] \quad (4.25)$$

When all devices (including MME) receive this new secret and decrypt it with the session key, they update their tokens to:

$$f_{new}(TID) = f(TID) + \Delta S \quad (4.26)$$

The token that each device received from KGC is the result of a polynomial function  $f(x)$ :

$$f(x) = \sum_{i=0}^{t-1} a_i x^i \text{ mod } p \quad (4.27)$$

Where the secret is a constant in polynomial,  $f(0) = a_0 = S$ , so, all tokens have the secret  $S$  as a constant in its composition. Then, when each member updates its own token with  $\Delta S$ , they are updating the secret  $S$ , present in its token, to a new secret  $S_{new}$ , as showed below:

$$f_{new}(x) = \sum_{i=0}^{t-1} a_i x^i \text{ mod } p + \Delta S = (S + \Delta S) + \sum_{i=1}^{t-1} a_i x^i \text{ mod } p \quad (4.28)$$

So, every member has the new secret and when the secret is recovered the result will be  $S_{new}$ .

## 4.4 SECURITY ANALYSIS

In this section, the protocol's security properties accomplishments and resistance to attacks is evaluated and explained.

### Mutual Authentication

- $MTCD_{i-j} \rightarrow HSS$   
In the proposed protocol, the HSS authenticates the  $MTCD_{leader}$  and all  $MTCD_{i-j}$  simultaneously, just verifying  $MAC_{Gi}$ , which authenticates the group because just a legitimate group has a valid GK and a valid  $ID_{Gi}$ .  $MAC_{Gi}$  also authenticates each  $MTCD_{i-j}$  because just legitimate and registered devices can find the original secret  $S$ , produced by KGC in registration phase.
- $MTCD_{i-j} \rightarrow MME$   
The MME authenticates all  $MTCD_{i-j}$  calculating the secret  $S'$ , using their Lagrange component, received in message 7 and comparing  $H(S')$  with  $H(S)$ , provided by the KGC in the registration phase. Just legitimate MTCDs can generate valid Lagrange components and only with valid Lagrange components, it is possible to recover the secret  $S$ .
- $MME \rightarrow MTCD_{i-j}$   
Each  $MTCD_{i-j}$  authenticate the MME verifying its Lagrange component. All devices calculate the secret  $S$  using the Lagrange component of MME and comparing with  $H(S)$  published by KGC. This verification authenticates MME because it just generates valid Lagrange's component if it received a legit token from HSS.
- $MTCD_{i-j} \rightarrow MTCD_{i-k}$   
In our protocol, before the authentication procedure arrive in the core network the  $MTCD_{i-j}$  authenticate themselves. Each device sends its Lagrange component to all members in the group. Then, every member uses these components to calculate the secret  $S$  and compare the value found with the value published by KGC. The Lagrange interpolating formula guarantees that the original secret only is recovered if all devices are legitimate.

### Man-In-The-Middle (MITM) Attack

- The channel between HSS and MME is secure. Then only entity that an attacker may act is between  $MTCD_{i-j}$  and MME. The mutual authentication phase is protected of MITM because:
  - Use of Shamir's secret and Lagrange interpolating formula, because with this formula is possible to construct a Lagrange component based on the secret token. It is quite complex to recover the secret token from Lagrange component and only with valid Lagrange components the secret will be recovered.
  - The group's ID is secret, just the TID is public and only who knows the ID can generate or verify  $MAC_{Gi}$ .

- Use of GK and GTK, only  $MTCD_{i-j}$  of the same group and the HSS know GK, so just them can generate the GTK.
- Use of session key in the communication between device and MME, where just legitimate devices can obtain a session key.

### Replay Attack

- Every process of authentication is different of the previous ones, because in each process new random values are generated to compound the messages, making almost impossible the repetition of this messages
- The parameters that perform this protection are:
  - Random values  $r_{MME}$ ,  $r_{HSS}$  and  $r_{MTCD_{i-j}}$ , present in session key and GTK.
  - Use of temporary identities:  $TID_{MTCD}$  and  $TID_{Gi}$ , where in every new authentication process these temporary identities are updated to a never used value and are never repeated.

### Privacy (Anonymity)

- The privacy of the devices is protected using temporary identities (TID), which perform security against targeted attacks, so, an attacker does not know the real device's identity.

### Redirection Attack

- Each MTCD includes the base station LAI in  $MAC_{Gi}$  and the MME, that also knows the devices base station LAI send it to the HSS in secure channel. If an attacker tries to forge LAI, the verification of  $MAC_{Gi}$  fails and the redirection attack is avoided.

### Personification Attack

- This attack occurs when an attacker tries to pass as legitimate MTCD or MME.
- MTCD → HSS  
An attacker can't forge valid tokens  $f(TID_{MTCD_{i-j}})$  because just KGC can build these tokens, which are based on the secret  $S$ , in a way that the right secret is recovered. Then, the attackers can't produce a valid Lagrange component, so, when the secret  $S$  is calculated the value found is different than the one published by KGC. When HSS verify  $MAC_{Gi}$  using the secret  $S$ , it can easily detect that is an attacker in group.
- MTCD → MME  
An attacker can't forge valid tokens  $f(TID_{MTCD_{i-j}})$ , then, they can't produce a valid Lagrange component. When MME receive all Lagrange components of a group, it tries to recover the secret  $S$  and realize that it is not the same published by KGC.
- MME → MTCD  
The same way, an attacker can't forge a valid Lagrange component, so when the MTCDs verify the  $LC_{MME}$  they realize that is an attacker, because the secret founded is not the same published by KGC.
- MTCGroup → HSS  
Is possible that a set of attackers try to pass themselves like a registered MTC group in network, so this attack will not succeed because only legitimate groups know a valid GK and can produce a valid  $S$ , so HSS will recognize the attack by verifying  $MAC_{Gi}$ .
- $MTCD_{i-j} \rightarrow MTCD_{i-k}$   
Even being from the same group, a MTCD can't pretend to be another MTCD of its group, because a device doesn't know the secret tokens,  $f(TID_{MTCD_{i-j}})$ , of each other and the attacker can't forge a valid Lagrange component of another member. Before sending a message to the network, the MTC group authenticates themselves calculating the secret  $S$ , at this moment all members realize that have at least one attacker in the group and the process fails.

Finally, one device can't generate a valid session key,  $SK_{i-j-MME}$ , of another device, because it doesn't know its own secret value in the tree.

## DoS Attack

This attack occurs when an attacker tries to drop the server or network sending a large amount of authentication messages, until it stops working perfectly.

- In our protocol, the HSS just receive the first message when the members of group already authenticate each other, so all devices can detect the presence of attackers in the group and stop the procedure, before the beginning involving the HSS in the authentication procedure.
- An attacker might create many fake messages to interrupt HSS service. The first message that HSS receives in our scheme contains  $MAC_{Gi}$  and HSS can quickly verify if it is valid or not, just calculating  $MAC'_{Gi}$  with which the received from the group. This verification happens at the beginning of the process, so the rest of the authentication procedure does not suffer if an attack is discovered in this stage.

## Backward Secrecy (BS) and Forward Secrecy (FS)

- The keys that are necessary to guarantee BS and FS are GK, the session key  $SK_{i-j-MME}$  and the Secret  $S$ .
- In our protocol, when a device enters or leaves the group GK is updated to perform BS and FS. In other words, if a device goes out, it cannot discover the future GKs and if a device enters the group, it cannot discover the past GKs.  
When a device is added to group, HSS broadcasts to all other devices its secret node and the new GK is:

$$GK'_i = h_3(GK_i \oplus SEC_{i-y}) \quad (4.29)$$

- When a device goes out, every device updates its GK as following:

$$GK''_i = GK_i \oplus SEC_{i-y} \quad (4.30)$$

- So, our protocol guarantee strong backward secrecy (sBS) and strong forward secrecy (sFS) to GK, because even though an attacker discovers the current GK it can't discover past and future GKs because he doesn't know the secret value used in the formula. Even if it occasionally discovers the current GK and the secret values used to generate GK, it will not compromise past or future GK, because the values used in the calculation are freshly renewed in each update. The same occur to  $SK_{i-j-MME}$ , because it is calculated as follows:

$$SK_{i-j-MME} = h_3(SEC_{i-a} \oplus SEC_{i-b} \oplus \dots \oplus SEC_{i-z} || r_{MME} || r_{MTCDi-j} || S) \quad (4.31)$$

If an attacker discovers the current value of the session key, he can't associate it to discover past or future keys, because he doesn't know the secret values (even if he was a member's group, he doesn't know his own secret value), neither the secret  $S$  and  $r_{MME}$  (if the attacker is not a group member). Even though the attacker eventually discovers all the secret values, the secret  $S$  and the currents  $r_{MME}$  and  $r_{MTCDi-j}$ , he won't be capable to calculates all past keys or future keys, because in each authentication process this random values are randomly generated. Also, if the attacker is not part of the group, he will not know the secret  $S$  and  $r_{MME}$ . Because of the described reasons, our session key has strong Backward Secrecy and strong Forward Secrecy.

The secret  $S$  must guarantee BS and FS because, if not, every new or old member will know the secret of the group and can try to perform attacks with this information. Then, any modification in the group formation needs an update in the secret  $S$ . The new secret is defined as:

$$S_{new} = S + \Delta S \quad (4.32)$$

Where  $\Delta S$  is a random term, which is defined every time an update in  $S$  is necessary. Then, even if an attacker discovers the current or the last secret  $S$ , he will not be able to discovery the next or the other past secrets, because  $S$  is defined by  $\Delta S$ . Even if  $\Delta S$  was discovered, this value does not have any correlation with future values or past values, so  $S$  is not compromised. Then, the secret  $S$  has strong Forward Secrecy and strong Backward Secrecy.

Table 22 – Comparison of security objectives between protocols

Security Objectives	EPS-AKA[1]	GLARM [13]	CHOI [11]	GR-AKA [15]	Proposed Protocol
<b>Mutual authentication and Key Agreement</b>	Yes	Yes	Yes	Yes	Yes
<b>Confidentiality</b>	No	Yes	Yes	Yes	Yes
<b>Integrity</b>	No	Yes	Yes	Yes	Yes
<b>Privacy (Anonymity)</b>	No	No	No	No	Yes
<b>Perfect FS/BS</b>	No	Yes	Yes	Yes	Yes
<b>Resistant to replay attack</b>	No	Yes	Yes	Yes	Yes
<b>Resistant to DoS attack</b>	No	Yes	Yes	No	Yes
<b>Resistant to Man-in-the-Middle attack</b>	No	Yes	Yes	Yes	Yes
<b>Resistant to redirection attack</b>	No	Yes	Yes	Yes	Yes
<b>Resistant to impersonation attack</b>	No	Yes	No	No	Yes

## 4.5 PERFORMANCE EVALUATION AND COMPARISONS

This section presents the performance evaluation of the proposed protocol and a comparison to the performance of some other protocols, [1], [11], [13] and [15]. All these protocols consider an architecture with a group of MTCs, MME, HSS, eNB and MTC Server, as showed in Figure 32. They also consider a safe channel between HSS and MME. Additionally, they have a registration/initialization phase, to define all the important parameters used in the protocol and an authentication and key agreement phase among the group of devices and a MME.

### 4.5.1 COMPUTATIONAL COST

The comparison of the computational cost of the proposed protocol with the other schemes analyzed is presented in two different approaches. The first analysis is based on the detailing of each arithmetic operation involved in the Lagrange component calculations. The second analysis is based on the Lagrange component time proposed by Li et al. [15], where a smaller computation time is presented, without explicit reasons.

The first protocols' analysis and comparison is presented on Table 24 and Figure 36. The time cost of each operation is showed on Table 23 and the values adopted were carefully chosen, based on the values used by [9], [11], [15] and [20]. The time to perform addition and XOR operations were omitted, since they are negligible if compared to the other operations. The following operations are considered:

Table 23 – Time cost of each operation considered in *ms*.

Notation	Cost (ms)	Description
$T_M$	0,013 ms[20]	Cost of a normal multiplication operation
<b>Thash</b>	0,06 ms [11]	Cost of a one-way hash operation
<b>Tmul</b> (MTC/Core)	1.537/0.475 ms [9]	Cost of a multiplication operation over elliptical curve
<b>Tmod</b>	0,12 ms [11]	Cost of a modular operation
<b>Taes</b>	0,16 ms [11]	Cost of AES encryption operation.
$TL_{MTC}$	0.0572 ms [15]	Cost of a Lagrange component creation in the MTCs
$TL_{Core}$	0.0351 ms [15]	Cost of a Lagrange component creation in the Core Network.

It is considered an environment with  $n$  devices, divided into  $m$  groups, where all groups have  $n/m$  members. In the proposed protocol, each MTCD perform 3 hash operations ( $GTK_{Gi}$ ,  $H(S)$ ,  $SK_{i-j-MME}$ ), 2 modular operations (mod  $p$ ) and  $\frac{n^2}{m}$  multiplications (Lagrange component). The group leader just perform a hash operation ( $MAC_{Gi}$ ), in a total of  $(\frac{n^2}{m} + n)T_M + (3n+m)Thash + 2nT_{mod} = 0,43n + 0,06m + 0,013\frac{n^2}{m}$  milliseconds in operations performed by devices.

The core network (MME and HSS) performs 1 hash operation to each MTCD ( $SK_{i-j-MME}$ ) and 3 hash operations ( $GTK_{Gi}$ ,  $H(S)$ ,  $MAC_{Gi}$ ), 2 modulus operation (mod  $p$ ), and 1 multiplication (Lagrange component) to each group, in a total of  $nT_M + (n+3m)Thash + 2mT_{mod} = 0,073n + 0,42m$  milliseconds. Consequently, the protocol takes  $0.5n + 0.48m + 0.013\frac{n^2}{m}$  milliseconds to successful complete the authentication and key agreement procedure.

Table 24 – Computation cost comparison between protocols (First method)

	MTCDs (ms)	Core Network (ms)	Total (ms)
<b>EPS-AKA[1]</b>	$6nThash + nT_{aes} = 0,52n$	$6nThash + nT_{aes} = 0,52n$	$1,04n$
<b>CHOI[11]</b>	$(7n+3m)Thash+nT_{mod}+mT_{aes} = 0,54n + 0,34m$	$(3n+6m)Thash + nT_{mod} + mT_{aes} = 0,3n + 0,52m$	$0,84n + 0,86m$
<b>GLARM[13]</b>	$8nThash + mThash = 0,48n + 0,06m$	$5nThash + 4mThash = 0,3n + 0,24m$	$0,78n + 0,3m$
<b>GR-AKA[15]</b>	$2nT_{mul} + (3n+4m)Thash + \frac{n^2}{m}T_M + 2mT_{mod} = 3,25n + 0,013\frac{n^2}{m} + 0,48m$	$(n+2m)Thash + (n+m)T_{mul} + mT_M = 0,54n + 0,61m$	$3,79n + 1,09m + 0,013\frac{n^2}{m}$
<b>Proposed Protocol</b>	$(\frac{n^2}{m}+n)T_M + (3n+m)Thash + 2nT_{mod} = 0,43n + 0,06m + 0,013\frac{n^2}{m}$	$nT_M + (n+3m)Thash + 2mT_{mod} = 0,073n + 0,42m$	$0,5n + 0,48m + 0,013\frac{n^2}{m}$

Table 24 shows that the proposed protocol communication costs are higher than [1], [11] and [13], just because of the Lagrange component generations, which generates a cost of order  $\frac{n^2}{m}$ . However, the 3GPP standard EPS-AKA [1] is not a group authentication protocol and the proposed protocol offers lower communication costs than [1], [11] and [13]. It also has a better security, as presented in Table 22. From Figure 36, it is possible to conclude that the computational cost of the proposed protocol is reduced when the number of groups ( $m$ ) rises. It has better performance when each group has less than 22 devices.

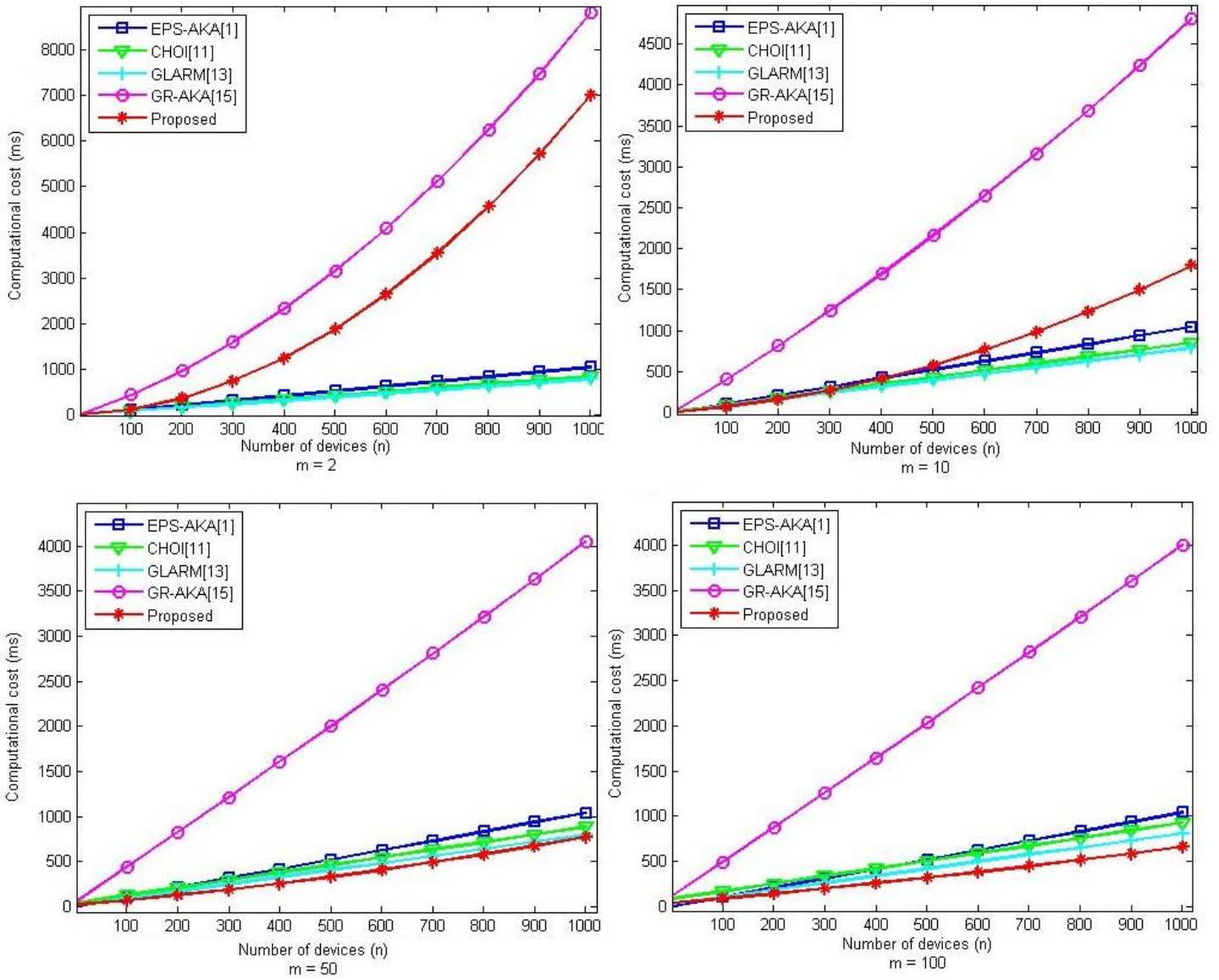


Figure 36 – Comparison of computational cost based on first analysis.

The second analysis and comparison method is presented on Table 25 and Figure 37. The time costs considered are presented in Table 23. The time to perform an XOR operation was omitted, since it is negligible if compared to the other operations.

It is considered an environment with  $n$  devices, divided into  $m$  groups, where all groups have  $n/m$  members. In the proposed protocol, each MTC performs 3 hash operations ( $GTK_{Gi}$ ,  $H(S)$ ,  $SK_{i-j-MME}$ ), 1 modular operation ( $\text{mod } p$ ) and 1 Lagrange component generation ( $LC_{i-j}$ ). The group leader just performs a hash operation ( $MAC_{Gi}$ ), in a total of  $nT_{MTC} + (3n+m)T_{hash} + 2nT_{mod} = 0,49n + 0,06m$  milliseconds in operations performed by devices.

The core network (MME and HSS) performs 1 hash operation to each MTC ( $SK_{i-j-MME}$ ) and 3 hash operations ( $GTK_{Gi}$ ,  $H(S)$ ,  $MAC_{Gi}$ ), 1 modulus operation ( $\text{mod } p$ ), and 1 Lagrange component generation ( $LC_{MME}$ ) to each group, in a total of  $mT_{LCore} + (n+3m)T_{hash} + 2mT_{mod} = 0,06n + 0,48m$  milliseconds. Consequently, the protocol takes  $0.55n + 0.54m$  milliseconds to successfully complete the authentication and key agreement procedure.



Table 25 – Computation cost comparison between protocols (Second method)

	MTCDs (ms)	Core Network (ms)	Total (ms)
<b>EPS-AKA[1]</b>	$6n\text{Thash} + n\text{Taes} = 0,52n$	$6n\text{Thash} + n\text{Taes} = 0,52n$	1,04n
<b>CHOI[11]</b>	$(7n+3m)\text{Thash} + n\text{Tmod} + m\text{Taes} = 0,54n + 0,34m$	$(3n+6m)\text{Thash} + n\text{Tmod} + m\text{Taes} = 0,3n + 0,52m$	$0,84n + 0,86m$
<b>GLARM[13]</b>	$8n\text{Thash} + m\text{Thash} = 0,48n + 0,06m$	$5n\text{Thash} + 4m\text{Thash} = 0,3n + 0,24m$	$0,78n + 0,3m$
<b>GR-AKA[15]</b>	$2n\text{Tmul} + 3n\text{Thash} + n\text{TL}_{\text{MTCD}} + 2m\text{Tmod} + 4m\text{Thash} = 3,31n + 0,48m$	$n\text{Thash} + n\text{Tmul} + m\text{TL}_{\text{Core}} + 2m\text{Thash} + m\text{Tmul} = 0,53n + 0,63m$	$3,84n + 1,11m$
<b>Proposed Protocol</b>	$n\text{TL}_{\text{MTCD}} + n\text{T}_M + (3n+m)\text{Thash} + 2n\text{Tmod} = 0,49n + 0,06m$	$m\text{TL}_{\text{Core}} + (n+3m)\text{Thash} + 2m\text{Tmod} = 0,06n + 0,48m$	$0,55n + 0,54m$

Table 25 shows that the proposed protocol has the lowest computational cost, if the second method of analysis is adopted. Figure 37 confirms its higher performance from small to large groups. By this method, the proposed protocol has the best performance among the studied protocols.

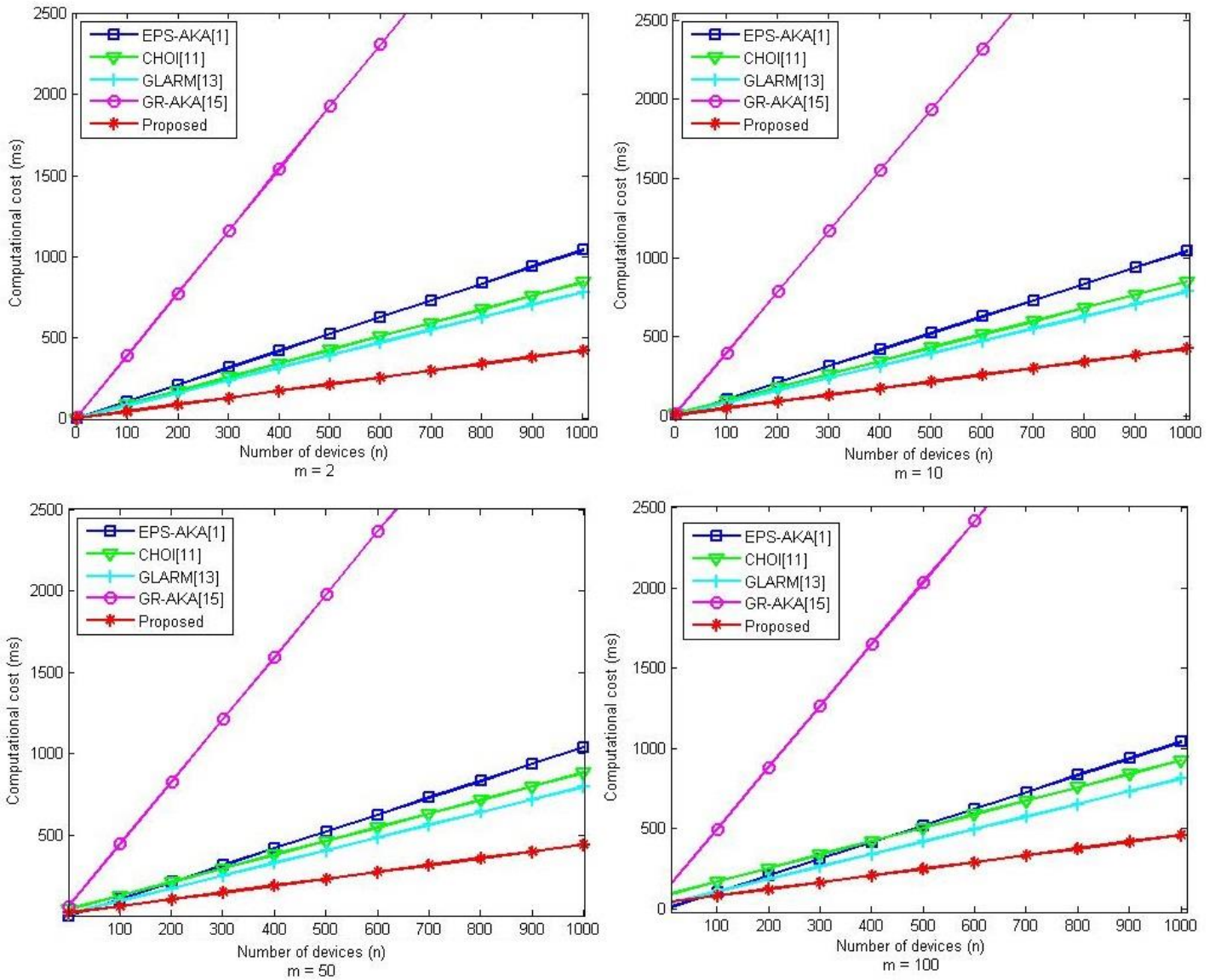


Figure 37 – Comparison of computational cost based on second analysis.

## 4.5.2 COMMUNICATION COST

The communication cost was measured in bits, by message exchanged. The values adopted to each parameter transmitted are presented on Table 26. They were carefully chosen, based on the values used by [9], [11], [14] and [15].

Table 26 – Communication cost of each parameter transmitted

Parameter	Size (bits)	Parameter	Size (bits)
<b>ID/TID</b>	128	<b>KDF</b>	128
<b>ECDH</b>	192	<b>AMF</b>	48
<b>MAC</b>	64	<b>Rand</b>	128
<b>Hash</b>	128	<b>LAI</b>	40
<b>PN, SQN</b>	128	<b>KL</b>	256
<b>Ts</b>	20	<b>AES</b>	256
<b>LC</b>	128		

It is considered an environment with  $n$  devices, divided into  $m$  groups, where in each group have  $\frac{n}{m}$  members. The calculations were based on the quantity of parameters exchanged in each message, in other words, it was considered every parameter that is sent through the channel. Taking message 5 as an example, the HSS send  $LC_{MME}$ ,  $GTK_{Gi} = h_2(GK||r_{HSS})$  and  $r_{HSS}$  to the MME. Hence, the message has two hash functions with 128 bits each and a random number with 128 bits, in a total of 384m bits. Table 27 compares the communication cost of the proposed protocol with the other protocols analyzed.

Table 27 – Communication cost in bits by message and in total.

	<b>M1</b>	<b>M2</b>	<b>M3</b>	<b>M4</b>	<b>M5</b>	<b>M6</b>	<b>M7</b>	<b>M8</b>	<b>TOTAL</b>
<b>EPS- AKA[1]</b>	-	128 bits	256 bits	704 bits	576 bits	128 bits	-	-	1792n bits
<b>CHOI[11]</b>	128n + 128m bits	128n + 256m bits	960m bits	576m bits	128m bits	128m bits	448n - 448 bits	256 m bits	704n + 1984m bits
<b>GLARM[ 13]</b>	448n - 448m bits	384n + 64m bits	384n + 104m bits	880m bits	560m bits	560m bits	128(n -m) bits	128 m bits	1344n + 1720m bits
<b>GR- AKA[15]</b>	340(n-m) bits	340m bits	380m bits	212m bits	404m bits	768n bits	-	-	1108n + 996m bits
<b>Proposed Protocol</b>	128n bits	128n bits	128n + 192m bits	232m bits	384m bits	512m bits	256n bits		640n + 1320m bits

From Table 27 it is possible to conclude that the proposed protocol has the lowest communication cost when compared to the other protocols analyzed. Figure 38 also shows this comparison and its good performance.

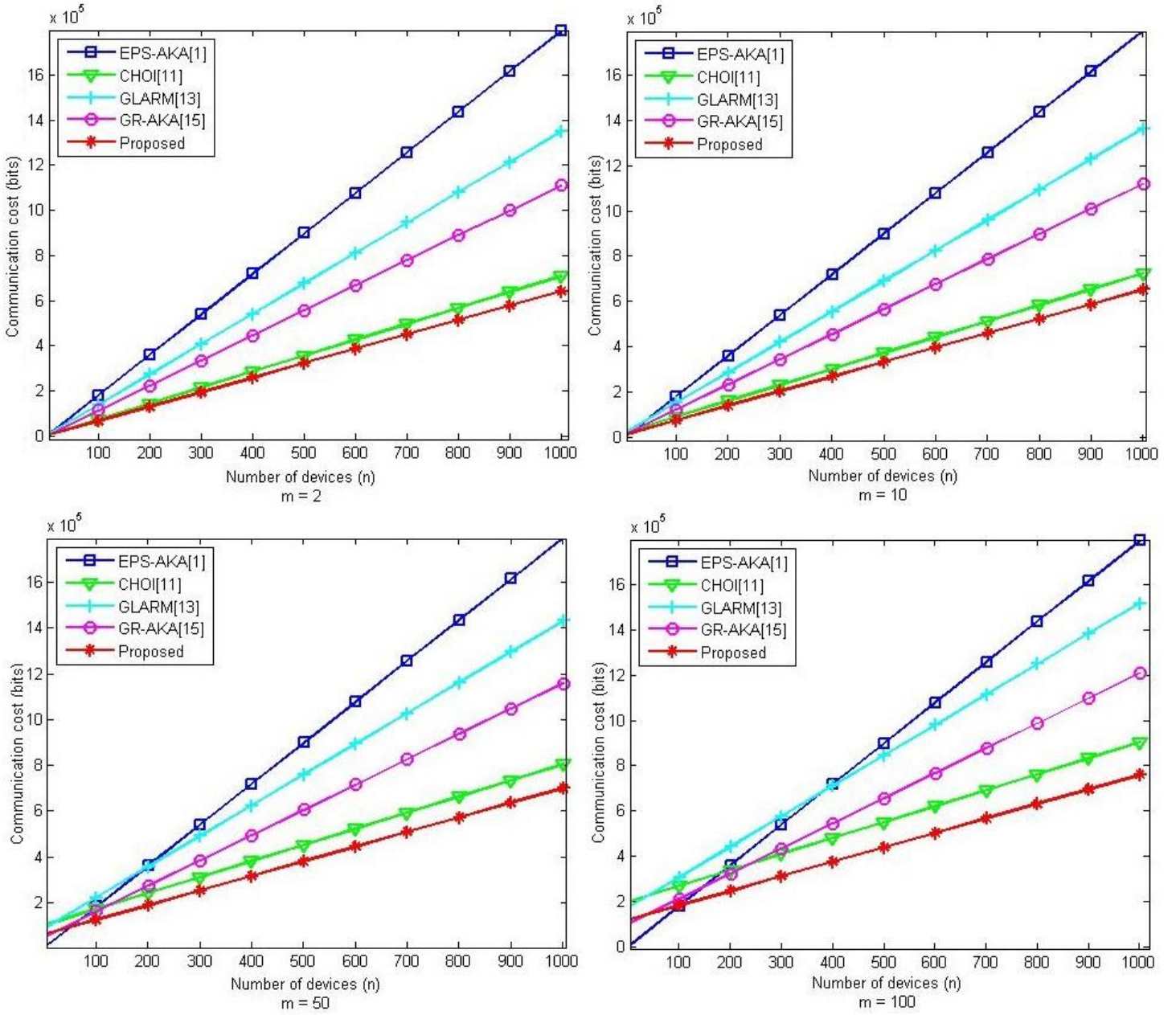


Figure 38 – Comparison of communication cost.

To better visualize the communication costs enhancements accomplished by our protocol when compared to 3GPP EPS-AKA[1], we also present an improvement rate ( $IR$ ), as described in [18]. The  $IR$  equation can be seen below:

$$IR = \frac{EPS|AKA - Proposed}{EPS|AKA} \quad (4.33)$$

After replacing the respective values in the equation, we obtained the following:

$$IR = 0.64 - 0.67 \frac{m}{n} \quad (4.34)$$

From the equation above, it can be deduced that the maximum improvement the proposed protocol can accomplish in relation to EPS-AKA is 64%. Hence, the proposed protocol stabilizes its enhancements in 0.64. This stabilization is shown in Figure 39, which shows that the proposed protocol also has a better IR than the other protocols studied.

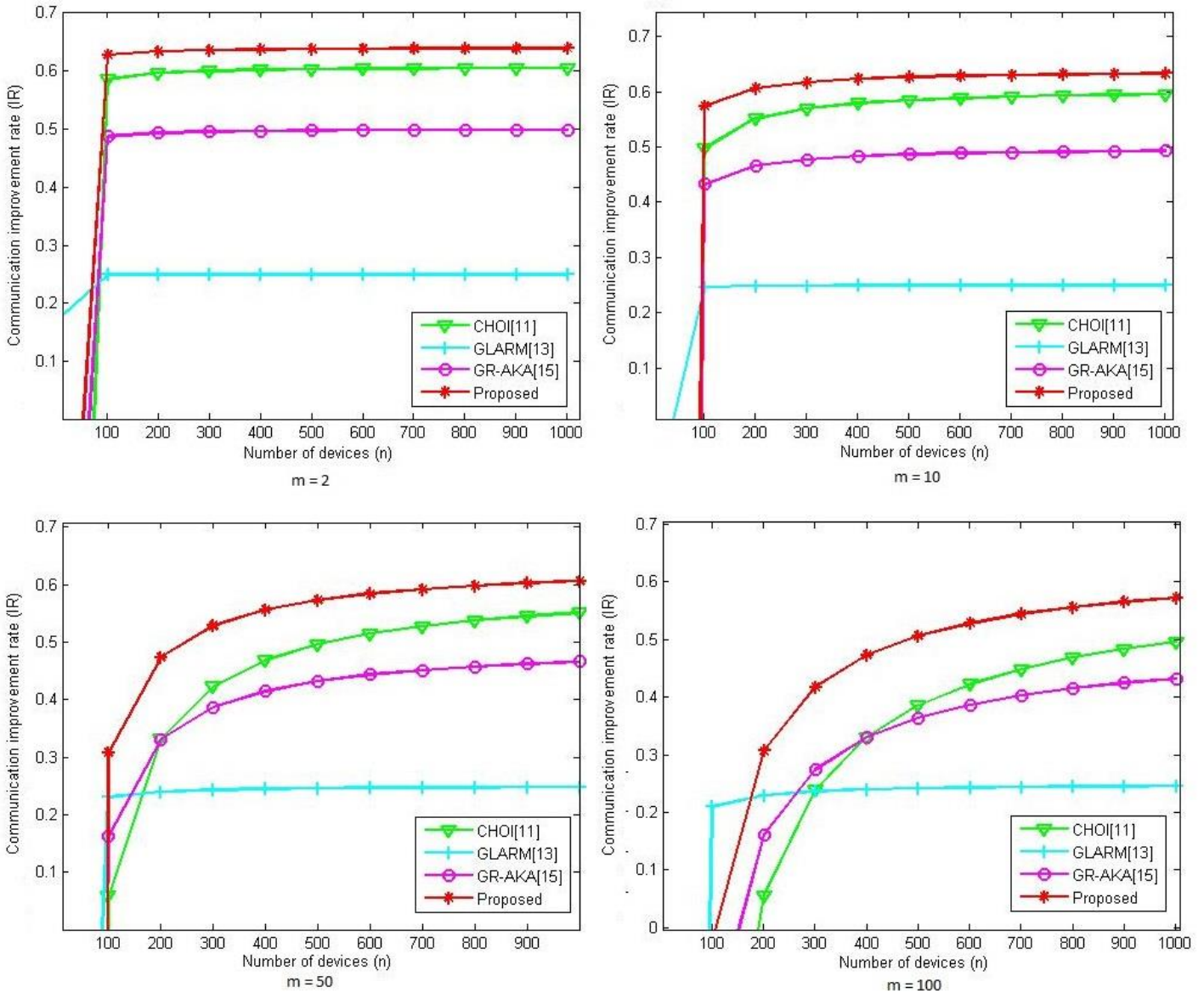


Figure 39 – Comparison of improvement rate for communication cost

### 4.5.3 STORAGE COST

In this section, we show the storage cost of the proposed protocol and compare it with the other protocols studied. Only the parameters derived from the authentication and key agreement procedure are considered. Each entity need to store some parameters to verify the devices. The storage cost comparison with the analyzed protocols can be seen on Table 28.

Table 28 – Storage cost in bits by entity.

Protocol	$MTCD_{i-j}$	$MTCD_{leader}$	MME
<b>CHOI[11]</b>	128	-	$128n + 512$
<b>GLARM[13]</b>	-	-	$256(n+1)$
<b>GR-AKA[15]</b>	-	320n	192
<b>Proposed Protocol</b>	$128n + 256$	-	$128n + 256$

The storage cost of the proposed protocol MME is lower than [11] and [13] and higher than Li et al. [15] protocol. Apparently, Li's protocol only uses freshly received information to calculate its parameters and this is the reason of its low storage cost. Additionally, each MTCD has higher storage cost than the others because it stores information from all the devices in the group to generate the right shared secret  $S$ . Although having high storage cost, the proposed protocol has better communication and computational (when the second method is adopted) costs than the other protocols studied. It also has an excellent security. We observe that an energy cost could be derived, considering the number of bits processed and transmitted for each component of the architecture and each coverage flow.

## 4.6 PROPOSED PROTOCOL FORMAL VERIFICATION

This section presents the formal verification of the proposed protocol, using AVISPA tool.

### 4.6.1 PROTOCOL SIMULATION

The AVISPA tool is based on the HLPSL language (High Level Protocol Specification Language). The protocols in HLPSL have a designated structure, as follows: roles, defining the agents that will participate in the process and their parameters; transitions, which is the action that each agent perform and the condition to this action to occur; session role, which defines the channel used in protocol; and the environment, which define all constants in protocol and the intruder's knowledge. The AVISPA tool analyses all these components in protocol and the knowledge that the intruder obtains during the protocol. Next, we will show our protocol written in HLPSL and in the end, we bring the results related with the security properties of our protocol, based on the analysis from AVISPA.

```

role
role_MTCDisj(MTCDisj:agent,MTCDI:agent,MME:agent,HSS:agent,IDg:text,TIDg:text,IDm:text,
TIDm:text,Tkm:text,GK:text,LCmtcd:text,SND,RCV:channel(dy))
played_by MTCDisj
def=
local
State:nat,
TIDI:text,
Hash:function
init
State := 0
transition
1. State=0  $\wedge$  RCV(start)  $\Rightarrow$  State':=1
 $\wedge$  TIDm':= Hash(IDm')
 $\wedge$  SND(TIDm)
 $\wedge$  secret(IDm', id_mtcd, {MTCDisj})

2. State=1  $\wedge$  RCV(TIDI')  $\Rightarrow$  State':=2
 $\wedge$  Tkm':=new()
 $\wedge$  LCmtcd':= Hash(TIDm',Tkm')
 $\wedge$  SND(LCmtcd)
 $\wedge$  secret(Tkm', token_mtcd, {MTCDisj})
end role

```

Figure 40 - Role of each MTCD in HLSPL.

```

role
role_MTCdI(MTCdIj:agent,MTCDI:agent,MME:agent,HSS:agent,IDg:text,TIDg:text,TIDl:text,IDL:
text,Tkl:text,GK:text,LAI:text,S:text,LCmtcd:text,Rmtcd:text,MACg:text,SND,RCV:channel(dy))
played_by MTCdI
def=
local
State:nat,TIDm:text,H2:function,IDmme:text,LCmme:text,Rmme:text,Rhss:text
init
State := 0
transition
1. State=0  $\wedge$  RCV(TIDm')  $\Rightarrow$  State':=1
 $\wedge$  TIDl':= H2(IDl')
 $\wedge$  SND(TIDl)
 $\wedge$  secret(IDl', id_leader, {MTCDI})

3. State=1  $\wedge$  RCV(LCmtcd')  $\Rightarrow$  State':=2
 $\wedge$  S':= H2(LCmtcd',Tkl)
 $\wedge$  MACg':= H2(GK.IDg.LAI.S)
 $\wedge$  SND(MACg'.TIDm'.TIDl')
 $\wedge$  secret(S', secret, {MTCDI,HSS})
 $\wedge$  secret(GK, group_key, {MTCDI,HSS})
 $\wedge$  secret(IDg, id_group, {MTCDI,HSS})
 $\wedge$  witness(MTCdI, HSS, group_auth, S')

7. State=2  $\wedge$  RCV(Rmme'.LCmme'.Rhss'.IDmme')  $\Rightarrow$  State':=3
 $\wedge$  Rmtcd':= new()
 $\wedge$  SND(LCmtcd.Rmtcd')
 $\wedge$  request(MTCdI, MME, mme_auth, LCmme')
 $\wedge$  witness(MTCdI,MME, group_auth_mme, LCmtcd')
end role

```

Figure 41 - Role of MTCdI<sub>leader</sub> in HLSPL

```

role
role_MME(MTCDij:agent,MTCDI:agent,MME:agent,HSS:agent,LAI:text,Rmme:text,LCmme:text,IDmme
:text,Key_set_MME_HSS:(symmetric_key) set,Key_set_HSS_MME:(symmetric_key)
set,SND,RCV:channel(dy))
played_by MME
def=
local
State:nat,TIDm:text,TIDI:text,S:text,IDg:text,GK:text,H2:function,TIDg:text,GTK:text,Tkmme:text,Rhss:te
xt,Rmtcd:text,MACg:text,LCmtcd:text,Key_2:symmetric_key,Key_1:symmetric_key
init
State := 0
transition
4. State=0  $\wedge$  RCV(MACg'.TIDm'.TIDI'.TIDg')  $\Rightarrow$  State':=1
 $\wedge$  Key_1':=new()
 $\wedge$  Key_set_MME_HSS':=cons(Key_1',Key_set_MME_HSS)
 $\wedge$  SND({TIDg'.LAI.MACg'}_Key_1')

6. State=1  $\wedge$  in(Key_2',Key_set_HSS_MME)
 $\wedge$  RCV({Tkmme'.Rhss'.GTK'}_Key_2')  $\Rightarrow$  State':=2
 $\wedge$  LCmme':= H2(Tkmme',TIDm',TIDI',IDmme)
 $\wedge$  Key_set_HSS_MME':=delete(Key_2',Key_set_HSS_MME)
 $\wedge$  SND(Rmme.LCmme.Rhss'.IDmme)
 $\wedge$  secret(Tkmme', token_mme, {MME,HSS})
 $\wedge$  witness( MME, MTCDI, mme_auth, LCmme')

8. State=2  $\wedge$  RCV(LCmtcd'.Rmtcd')  $\Rightarrow$  State':=3
 $\wedge$  request(MME, MTCDI, group_auth_mme, LCmtcd')
end role

```

Figure 42 – Role of the MME in HLSPL

```

role
role_HSS(MTCDij:agent,MTCDI:agent,MME:agent,HSS:agent,IDm:text,IDg:text,IDL:text,S:text,
TIDg:text,GK:text,Tkmme:text,Rhss:text,GTK:text,MACg:text,Key_set_MME_HSS:(symmetric_
key) set,Key_set_HSS_MME:(symmetric_key) set,SND,RCV:channel(dy))
played_by HSS
def=
local
State:nat,LAI:text,H2:function,Key_2:symmetric_key,Key_1:symmetric_key
init
State := 0
transition
5. State=0  $\wedge$  in(Key_1',Key_set_MME_HSS)
 $\wedge$  RCV({LAI'.MACg'}_Key_1')  $\Rightarrow$  State':=1
 $\wedge$  Key_set_MME_HSS':=delete(Key_1',Key_set_MME_HSS)
 $\wedge$  Key_2':=new()
 $\wedge$  Key_set_HSS_MME':=cons(Key_2',Key_set_HSS_MME)
 $\wedge$  Tkmme':= new()
 $\wedge$  Rhss':= new()
 $\wedge$  GTK':= H2(GK, Rhss')
 $\wedge$  SND({Tkmme'.Rhss'.GTK'}_Key_2')
 $\wedge$  secret(GTK', temp_group_key, {HSS,MME,MTCDI,MTCDij})
 $\wedge$  request( HSS, MTCDI, group_auth, S')
end role

```

Figure 43 – Role of the HSS in HLSPL



```

role
session1(IDmme:text,LCmme:text,Rmme:text,LAI:text,Tkm:text,TIDm:text,TIDl:text,Tkl:text,LC
mtcd:text,Rmtcd:text,MTCDij:agent,MTCDl:agent,MME:agent,HSS:agent,IDm:text,IDg:text,IDL:t
ext,S:text,TIDg:text,GK:text,Tkmme:text,Rhss:text,GTK:text,MACg:text,Key_set_MME_HSS:(s
ymmetric_key) set,Key_set_HSS_MME:(symmetric_key) set)
def=
local
SND4,RCV4,SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
composition
role_HSS(MTCDij,MTCDl,MME,HSS,IDm,IDg,IDL,S,TIDg,GK,Tkmme,Rhss,GTK,MACg,Key_
set_MME_HSS,Key_set_HSS_MME,SND4,RCV4)
^
role_MME(MTCDij,MTCDl,MME,HSS,LAI,Rmme,LCmme,IDmme,Key_set_MME_HSS,Key_s
et_HSS_MME,SND3,RCV3)
^
role_MTCDl(MTCDij,MTCDl,MME,HSS,IDg,TIDg,TIDl,IDL,Tkl,GK,LAI,S,LCmtcd,Rmtcd,MA
Cg,SND2,RCV2)
^
role_MTCDij(MTCDij,MTCDl,MME,HSS,IDg,TIDg,IDm,TIDm,Tkm,GK,LCmtcd,SND1,RCV1)
end role

role environment()
def=
const
id_mtcd, token_mtcd,id_leader,secret, group_key,id_group, token_mme, temp_group_key :
protocol_id,%% Secrecy
group_auth, mme_auth, group_auth_mme : protocol_id,%% Authentication
hash_0:function,rhss:text,gk:text,s:text,idg:text,hss:agent,leader:agent,rmtcd:text,tkl:text,tidm:text,
lai:text,lcmme:text,idmme:text,rmme:text,tkm:text,tidl:text,lcm:text,device:agent,mme:agent,idm:t
ext,idl:text,tidg:text,tkmme:text,rhss:text,gtk:text,macg:text
intruder_knowledge = { device,leader,mme,hss,tidl,tidm }

composition
session1(idmme,lcmme,rmme,lai,tkm,tidm,tidl,tkl,lcm,rmtcd,device,leader,mme,hss,idm,idg,idl,s,t
idg,gk,tkmme,rhss,gtk,macg,{},{})
end role

```

Figure 44 – Role specification for the session and environment in HLSP.

```

goal
secrecy_of id_mtcld
secrecy_of token_mtcld
secrecy_of id_leader
secrecy_of secret
secrecy_of group_key
secrecy_of id_group
secrecy_of token_mme
secrecy_of temp_group_key
authentication_on group_auth
authentication_on mme_auth
authentication_on group_auth_mme

end goal

environment()

```

Figure 45 – Security goals intended in HLSP.

Figure 45 shows all the goals evaluated. This section necessary to enable AVISPA to analyze the protocol according with the security properties defined.

#### 4.6.2 SECURITY VERIFICATION RESULTS

The AVISPA tool have some simulators to test some attacks in protocol, in this work we focus on two simulations: OFMC and CL-AtSe. The simulation results showed that the proposed protocol is safe to both. These results can be seen in Figure 46 and 47.

```

% OFMC
% Version of 2006/02/13
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/span/span/testsuite/results/protc2.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.01s
visitedNodes: 8 nodes
depth: 3 plies

```

Figure 46 - Security simulation results for OFMC

**SUMMARY**

**SAFE**

DETAILS

BOUNDED\_NUMBER\_OF\_SESSIONS

TYPED\_MODEL

PROTOCOL

/home/span/span/testsuite/results/protc2.if

GOAL

As Specified

BACKEND

CL-AtSe

STATISTICS

**Analysed : 4 states**

Reachable : 1 states

Translation: 0.01 seconds

Computation: 0.00 seconds

Figure 47 - Security simulation results for CL-AtSe.

Figures 46 and 47 show the results that AVISPA found when analyzing the security properties of our protocol. Figure 46 is an analysis of OFMC simulation and the protocol is considered safe to the goals specified. Figure 47 shows the CL-AtSe simulation and it considered our protocol safe too.

The AVISPA have also a graphic simulation tool, SPAN (Security Protocol Animator for AVISPA), which permit a better visualization of the messages exchanged and the participation of the intruder during the protocol. The graphical animations of our protocol are shown in Figure 48 and Figure 49.

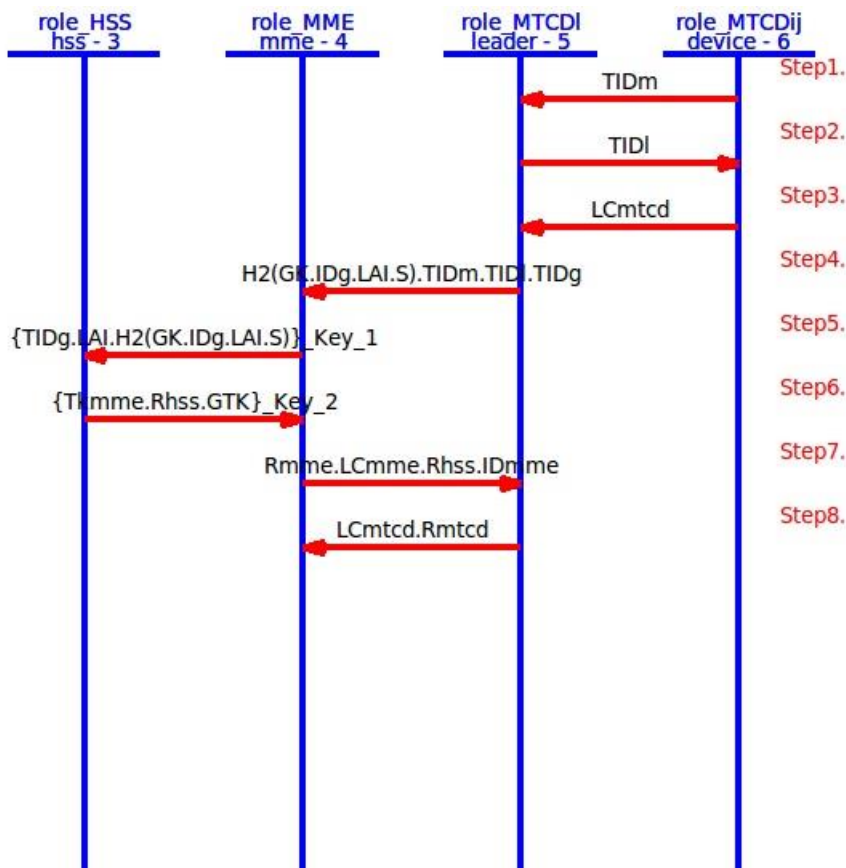


Figure 48 – Protocol’s message exchange in SPAN

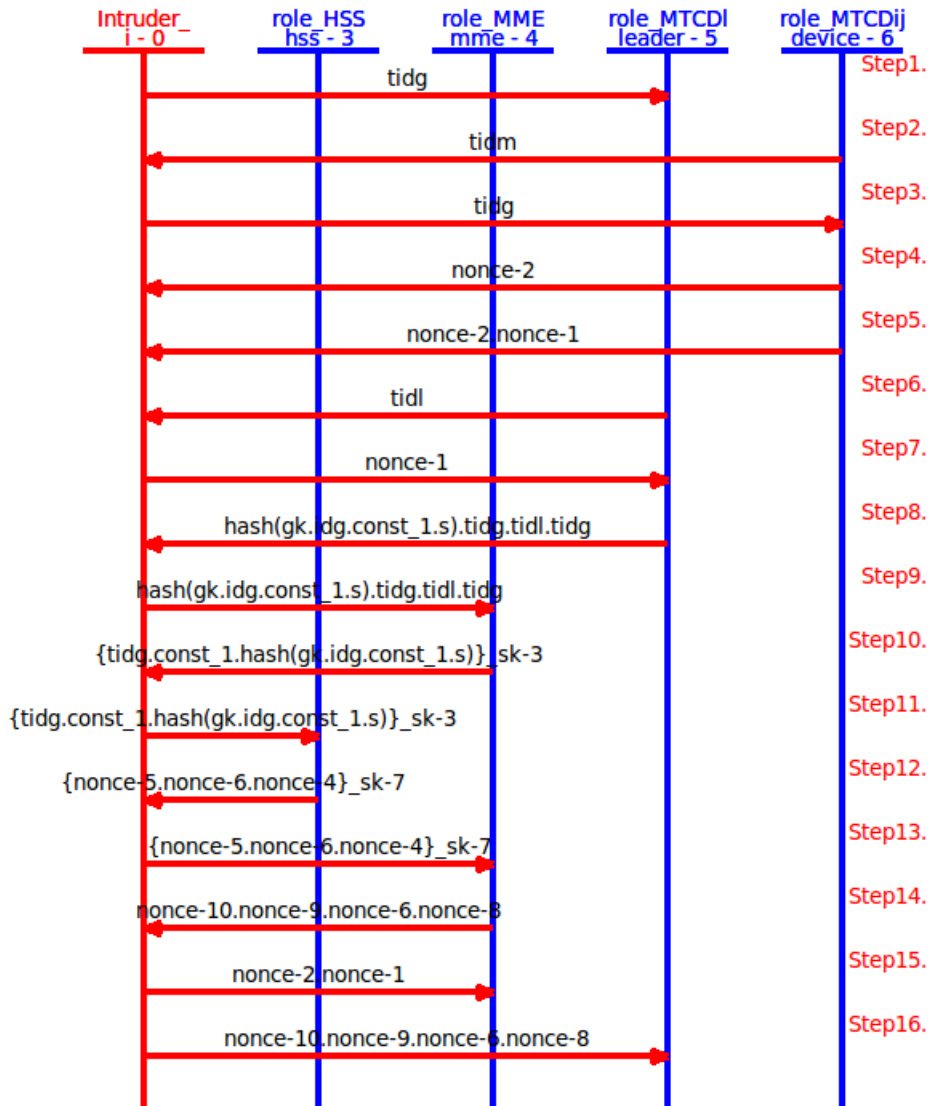


Figure 49 - Intruder's simulation in SPAN

## 4.7 CONCLUSION

The chapter had the purpose of presenting a proposal of group authentication and key agreement protocol, based Shamir's secret and using symmetric cryptography. First, it introduced some of the group authentication protocols already proposed in the literature by [11], [13] and [15]. It also described the Asynchronous (t; m; n) Group Authentication Scheme proposed by Harn [16] and used as the base to our protocol.

Then, two of the referenced protocols were detailed, Lai et al. [13] and Li et al. [15], to enable an immersion in group authentication scenario, before the presentation of the proposed protocol. Next, the proposed protocol was presented, with a detailed description of its basic assumptions and of registration and authentication and key agreement phase. After, its security objectives and performance were evaluated and compared with the other protocols described in this chapter. The performance evaluation comprised the computation, communication and storage costs.

The proposed protocol computational performance was evaluated in two different ways. The first one, detailing the operations executed to construct the Shamir secret, this is, to generate the Lagrange's component. The second evaluation was performed using the Lagrange's component execution time proposed by Li et al. [15]. In the first evaluation model, had the second higher cost, losing only to [15]. However, its performance improves as the number of groups grows and it presents the best performance

to groups with less than 22 devices. As expected, our protocol had a better performance when evaluated with the second method, presenting lower computational cost than all the other protocols compared.

The proposed protocol also has the lowest communication cost of the analyzed protocols, proving its efficiency. Finally, a formal verification using AVISPA simulation tool was executed proving that the proposed protocol accomplishes the security objectives necessary to successful group authentication.

First, it is necessary to mention the importance of this work, which is interdisciplinary and involves information security, wireless networks, Internet of Things (IoT), Quality of Service (QoS) and formal verification of protocols. This work introduced some of the main group authentication and key agreement protocols currently available in the literature and also two new protocols were proposed, based on the best characteristics of previous proposals ([9],[12],[13], [14], [15] and [16]). Both protocols had their phases detailed and had their security, performances evaluated and compared with the other protocols studied. Additionally, the protocols were formally verified by AVISPA tool, to guarantee that accomplishment of the necessary security goals to be considered safe. Both protocols guarantee that all the devices that arrived together in a server network are authenticated at the same time.

The first protocol proposed, presented in chapter 3, is based on asymmetric cryptography, ECDH and bilinear pairing. It uses the binary tree proposed by Choi et al. [11] as the devices management mechanism. The protocol has initialization and authentication and key agreement phases. The initialization phase defines and distribute important parameters that will be used in the next phase. The authentication and key agreement phase performs mutual authentication between a group of devices and the MME, establishing a session key among them if the procedure were successful. The session key secrecy lays on the binary tree configuration and on the discrete logarithm problem, provided using ECDH.

The security objectives accomplished by the protocol described in chapter 3 are mutual authentication, confidentiality, integrity, perfect forward and backward secrecy, anonymity of the devices and protection against replay, DoS, man-in-the-middle, redirection and impersonation attacks. It presents better security than the other compared protocols, [9],[12],[14], as presented in Table 8.

The performance evaluation of the first protocol comprised computational, communication, membership verification and storage costs. When compared to some of the protocols studied, [9], [12] and [14], the protocol proposed in chapter 3 presented better computational cost and a communication cost a little lower than the protocol proposed by Cao et al. [9]. However, Cao's protocol does not guarantee protection against redirection attack and does not preserve the anonymity of the devices during the procedure. In addition, it has a higher computational cost than our proposed protocol. The first proposed protocol also has the best verification cost

The second protocol proposed in this work, presented in chapter 4, is based on symmetric cryptography, Shamir's secret and in the Asynchronous (t,m,n) GAS proposed by Harn [16]. It also uses the binary tree proposed by [11] as the devices management mechanism. The protocol has registration and authentication and key agreement phases. The initialization is responsible to distribute the parameters used in the authentication and key agreement phase, as the temporary identity TID a token associated to it. The authentication and key agreement phase has the same purpose of the first protocol. However, the session key generated by the protocol proposed in chapter 4, beyond being based on the binary tree secret values, is based on the secrecy of the Shamir's secret generated among the group of devices and the MME.

The second protocol accomplishes the following security objectives: Mutual authentication, anonymity, backward and forward secrecy and protection against man-in-the-middle, replay, redirection, impersonation and DoS attacks. Table 22 confirms that when compared to [11],[13] and [15], the protocol proposed in chapter 4 has better security performance.

The performance evaluation of the second protocol comprised computational, communication and storage costs. The computational cost was evaluated using two different methods, the first method considered all the operations performed to calculate a Lagrange component and the second method used the time proposed by Li et al. [15] to represent the respective component. By the first method, the proposed protocol presented high costs to a scenario with low number of groups and lower costs as the number of groups rise. But even with high costs, the protocol presented better performance than Li et al., maybe because [15] is based on asymmetric cryptography. When evaluated by the second method, the protocol proposed in chapter 4 has the best computational performance when compared to [11], [13] and [15]. The proposed protocol communication cost also is better than all the protocols studied.

The formal verification using AVISPA tool returned that both protocols are safe, completing their proof of efficiency and safety. By the safety and performance analysis described in the work, it is possible to assure that both protocols are good choices to be implemented in the authentication of groups of devices for Internet of Things.

## REFERENCES

- [1] 3GPP (2009) 3GPP TS 33.401 V8.2.1, 3GPP System Architecture Evolution (SAE). Security Architecture.
- [2] Stallings, William. *Cryptography and network security: principles and practice*. sixth ed. Boston: Pearson, 2014.
- [3] Menezes, Alfred. An introduction to pairing-based cryptography. *Recent trends in cryptography*, v. 477, p. 47-65, 2005.
- [4] Kahn, J., Abbas, H., Al-Muhtadi, J. Survey on Mobile User's Data Privacy Threats and Defense Mechanisms. *International Workshop on Cyber Security and Digital Investigation (CSDI 2015)*, p. 376-383, 2015.
- [5] Zhang, M., Fang, Y. Security Analysis and Enhancements of 3GPP Authentication and Key Agreement Protocol. *IEEE Trans. Wireless Communication*, v. 4, n. 2, p. 734-742, Mar. 2005.
- [6] The AVISPA Project: European Union in the Future and Emerging Technologies (FET Open). Retrieved Nov 26, 2016, from <http://www.avispa-project.org>.
- [7] Armando, A. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications, Proc. CAV 2005, v. LNCS 3576, p. 281-285, 2005.
- [8] Cao, J., Ma M., and Li, H. ABAAM: Access Authentication of Mass Device Connections for MTC in LTE Networks. *Smart Computing. Rev.*, v. 4, n. 4, p. 262–277, 2014.
- [9] Cao, J., Ma, M. and Li, H. GBAAM: Group-based Access Authentication for MTC in LTE Networks. *Security and Communication Networks*, v. 8, n. 17, p.3282-3299, 2015.
- [10] Chen, Y.W., Wang, J.T., Chi, K.H. and Tseng, C.C. Group-based authentication and key agreement. *Wireless Personal Communications*, v. 62, n. 4, p. 965–979, 2012.
- [11] Choi, D., Hong, S. and Choi, H.K. A group-based security protocol for Machine Type Communications in LTE-Advanced. *Wireless Network*, v. 21, n. 2, p.405-419, 2015.
- [12] Fu, A., Song, J., Li, S., Zhang, G. and Zhang. Y. A privacy-preserving group authentication protocol for machine-type communication in LTE/LTE-A networks. *Security and Communication Networks*, v. 9, n. 13, p. 2002–2014, 2016.
- [13] Lai, C., Lu, R., Zheng, D., Li, H. and Sherman, X. GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications. *Comput. Networks*, v. 99, p. 66–81, 2016.
- [14] Lai, C., Lu, R., and Shen, X. SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks. *Computer Networks*, v. 57, n. 17, p. 3492-3510, 2013.
- [15] Li, J., Wen, M., Zhang, T. Group-based authentication and key agreement with dynamic policy updating for MTC in LTE-A Networks. *IEEE Internet of Things Journal*, v. 99, p. 1-9, 2016.
- [16] Harn, L. Group Authentication. *IEEE Transactions on Computers*, v. 62, n. 9, p. 1893-1898, 2012.
- [17] Cao, J., Ma, M.D., Li, H. Handover authentication between different types of eNBs in LTE networks. *The Journal of China Universities of Posts and Telecommunications*, Elsevier, v. 20, n. 2, p 106–112, 2013.
- [18] Jiang, R., Lai, C., Luo, J., Wang, X. and Wang, H. EAP-based group authentication and key agreement protocol for machine-type communications. *International Journal of Distributed Sensor Networks (IJDSN)*, p. 1-14, 2013.
- [19] Giustolisi, R. and Gehrmann, C. Threats to 5G Group-Based Authentication. *13th International Conference on Security and Cryptography (SECRYPT 2016)*, p. 26-28, 2016.
- [20] Cao, J., Li, H., Ma, M., Zhang, Y., and Lai, C. A simple and robust handover authentication between HeNB and eNB in LTE networks. *Computer Networks*, v. 56, n. 8, p. 2119–2131, 2012.



- [21] Jeffreys, H. and Jeffreys, B. S. Lagrange's Interpolation Formula. *Methods of Mathematical Physics*, 3rd ed. Cambridge, England: Cambridge University Press, p. 260, 1988.
- [22] Shamir, A. How to share a secret. *Communications*, p. 612-613, 1979.
- [23] Cremers, C. J. F. and Feltz, M. Beyond eCK: perfect forward secrecy under actor compromise and ephemeral-key reveal. *Designs, Codes and Cryptography*, v. 74, n. 1, p. 183–218, 2015.
- [24] Cisco “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020.” Retrieved Nov 26, from <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>.
- [25] Holma, H. and Toskala, A. LTE-Advanced: 3GPP Solution for IMT-Advanced, Wiley, 2012.
- [26] Basin, D., Moedersheim, S., and Vigano, L. OFMC: A symbolic model checker for security protocols. *International Journal of Information Security*, v.4, n.3, p. 181-208, 2005.
- [27] Dutta, R., Barua, R. and Sarkar., P. Pairing-based cryptography: A survey. *Cryptology Research Group, Stat-Math and Applied Statistics Unit*, v.203, 2004.
- [28] Abbasi, A. A. and Younis, M. A survey on clustering algorithms for wireless sensor networks. *Computer communications*, v.30, n.14, p.2826-2841, 2007.
- [29] Chatterjee, M., Das, S. K. and Turgut, D. WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks. *Clustering Computing*, v.5, p. 193–204, 2002.