AMERICAN UNIVERSITY OF BEIRUT

An Identity Based Security Framework for Advanced Metering Infrastructure

by
VAHE SARKIS SEFERIAN

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
April 2016

AMERICAN UNIVERSITY OF BEIRUT

An Identity Based Security Framework for Advanced Metering Infrastructure

By
VAHE SARKIS SEFERIAN

Approved by:

.....

Dr. Rouwaida Kanj, Assistant Professor — Advisor
Electrical and Computer Engineering

.....

Dr. Ali Chehab, Professor — Member of Committee
Electrical and Computer Engineering

.....

Dr. Ayman Kayssi, Professor — Member of Committee
Electrical and Computer Engineering

Date of thesis defense: April 22, 2016

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: _____
Last First Middle

⬭ Master's Thesis ⬭ Master's Project ⬭ Doctoral Dissertation

☐ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, **three years after the date of submitting my thesis, dissertation, or project,** to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

_____

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Rouwaida Kanj for her time, efforts and patience. She always had an open door whenever I needed guidance or help about my research or writing. I would also like to thank the committee members Dr. Ali Chehab and Dr. Ayman Kayssi for their continuous help and support throughout my research.

I would like to thank the University Research Board (URB) at the American University of Beirut and the Lebanese National Council for Scientific Research (LNCSR) for funding this research.

I would also like to thank my friends and colleagues for being there and understanding me at times where I had to be away from them. A special thanks goes to Nanor who stood next to me throughout this work. I thank her for her encouragement and infinite understanding.

Finally, I would like to express my gratitude to my mom and dad for supporting me by all means. Without you, I wouldn't be where I am now. Thank you!

AN ABSTRACT OF THE THESIS OF

Vahe Sarkis Seferian for Master of Engineering
Major: Electrical and Computer Engineering

Title: An Identity Based Security Framework for Advanced Metering Infrastructure

The smart grid represents the next generation power grid and will have a substantial role in increasing the reliability and sustainability of energy production and distribution of power delivery networks. Particularly, the integration of information and communication technology is considered to be an evolution in the context of existing power grids. The establishment of two-way communication channels between different components of the smart grid and the utility offers several benefits ranging from early detection of blackouts to time-based rates of energy and demand management programs. Compromising the communication and networking data systems of the smart grid will lead to the risk of compromising reliable and secure power system operations which is the ultimate objective of the smart grid. Therefore, cyber security of the smart grid is of high importance and is considered as one of the highest priorities of the smart grid design. This thesis will focus on the security of the advanced metering infrastructure (AMI) network, which is the last mile of the smart grid, connecting smart meters to the utility. Specifically, the thesis covers topics related to efficient and scalable key management by relying on identity based cryptographic primitives. The first part of the thesis proposes an identity based non-interactive and scalable key distribution system for generating pairwise symmetric keys to be used for link-layer security. In addition, to prevent the discovery and access of the cryptographic keys stored on accessible memory locations physical unclonable functions are adopted. In the second part of the thesis, we present a new lightweight key update and delivery method to securely update the private keys utilized by the identity based cryptosystem. Finally, we propose a new method for deriving keys for the purpose of ensuring a forward secure communication between the meters and the utility at the application level. The proposed method is based on merging identity based cryptography and Diffie-Hellman key exchange for the purpose of deriving forward secure keys. The proposed ideas and algorithms are assessed on hardware platforms that mimic processors used by smart meter vendors and on event driven network simulators in order to simulate the proposed systems in the context of AMI wireless mesh networks.

Table of Contents

Acknowledgments vi
Abstract vi
List of Figures ix
List of Tables x
Chapter 1 – Introduction 1
1.1 Overview of Smart Grids 1
1.2 Wireless mesh based AMI Architecture 2
1.3 Overview of Smart Grid Security Requirements 3
1.4 Scope of Thesis 4
Chapter 2 – Background Review 5
2.1 Literature Review 5
2.2 Cryptographic Primitives 14
2.2.1 Hash Functions & HMAC 14
2.2.2 Symmetric Key Cryptography 15
2.2.3 Asymmetric Key Cryptography 16
2.2.4 Discrete Log Problem (DLP) 18
2.2.5 Diffie–Hellman key exchange 19
2.2.6 Elliptic Curves in Finite Fields 20
2.2.7 Elliptic Curve Discrete Log Problem (ECDLP) 21
2.2.8 Elliptic Curve Diffie-Hellman Key Exchange (ECDH) 22
2.2.9 A Notion of Bilinear Pairings on Elliptic Curves 22
2.2.10 Bilinear Maps 23
2.2.11 A notion of Identity-based cryptosystems 24
Chapter 3 – Identity Based Scalable Key Distribution for AMI networks 25
3.1 Introduction and Motivation 25
3.2 Physical Unclonable Functions 26
3.3 Proposed Methodology 28
3.3.1 Attacker model 29
3.3.2 Methodology overview 29
3.4 Pre-deployment phase 30
3.5 ID-NIKD and pairwise symmetric key generation 32
3.6 Field phase and packet overview 33
3.7 Simulations & Results 35
3.7.1 Simulation Setup 36
3.7.2 Simulation Results 37

Chapter 4 - Lightweight Key Update & Delivery Mechanism for ID-based Cryptosystem 41
4.1 Introduction 41
4.2 Private key update process 41
4.3 Key Delivery Mechanism and Shuffled-ID based Security Scheme 42
4.3.1 Precursor 43
4.3.2 Key delivery and Authentication Mechanism 43
4.4 Blocks Ciphers, Pseudo Random Permutation & AES 46
4.5 Security of Key distribution mechanism 47
4.6 Simulation Analysis and Results 50
4.6.1 Hardware Testing 50
4.6.2 Network Simulations 51
Chapter 5 – Efficient Forward Secure Meter-to-Utility Communication 57
5.1 Introduction & Motivation 57
5.2 Diffie Hellman & Forward Secrecy 58
5.3 Elliptic Curve Diffie Hellman 59
5.4 Identity Based Non-Interactive Key Exchange 60
5.5 Proposed Methodology 60
5.6 Simulations & Results 62
5.6.1 Hardware Testing 62
5.6.2 Network Simulations 63
Chapter 6 – Conclusion 70
References 71

List of Figures

Figure 1 - Smart grid network architecture....1
Figure 2 – Wireless mesh based AMI architecture....2
Figure 3 - AMI attack tree in [5]....7
Figure 4 - Initialization and Authentication in [14]....8
Figure 5 - Meter reading collection process in [14]....9
Figure 6 - Lightweight authentication scheme in [16]....11
Figure 7 – RO based PUF in [17]....12
Figure 8 - Key generation in [17]....13
Figure 9 – PUF system on chip design in [18]....14
Figure 10 - Representation of one-way trapdoor function....17
Figure 11 - High level overview of a 1-bit ring oscillator PUF....27
Figure 12 - PUF based SoC....28
Figure 13 - End-to-end and hop-by-hop security mechanism overview....29
Figure 14 - Packet overview illustrating security mechansims....34
Figure 15 – Hop-by-hop authentication mechanism in action....35
Figure 16 - Topology of the simulated network....36
Figure 17 - Average packet latency for Exp #1....39
Figure 18 - Average packet latency for Exp #2....39
Figure 19 - Average packet latency for Exp #3....39
Figure 20 - Unicast key delivery (left) vs Mutlicast (right).....44
Figure 21 - Proposed key delivery packet at the application layer....45
Figure 22 - Number of stored keys vs. key size....45
Figure 23 – Utilization of AES within proposed system....48
Figure 24 - Probability of collision for different ID sizes....49
Figure 25 - Chain based network used in simulation....51
Figure 26 - Tree based network used in simulation.....51
Figure 27 - Delay comparison for the chain based network....54
Figure 28 - Delay comparison for the tree based network....55
Figure 29 - Bytes received at each node for a tree based network....56
Figure 30 - Power usage to personal activity mapping....57
Figure 31 – ECDH key exchange illustration....59
Figure 32 – ID-NIKE with ECDH....61
Figure 33 - Square network used in simulations....64
Figure 34 - Packet form at the application layer for ECDH-ID-NIKE....65
Figure 35 - Normal metering data in 7x7 network....66
Figure 36 - ECDH-RSA key exchange in 7x7 network....66
Figure 37 - Proposed key exchange in 7x7 network....67
Figure 38 - Normal metering data in 11x11 network....67
Figure 39 - ECDH-RSA key exchange in 11x11 network....68
Figure 40 - Proposed key exchange in 11x11 network....68

List of Tables

Table 1 - Security requirements: SG vs Internet 4
Table 2 - Requirements for cryptographic hash functions 15
Table 3 - Double-and-Add algorithm 21
Table 4 - Experimental setups with different network parameters. 37
Table 5 - Average packet latency for variable packet size corresponding to Figure 18 38
Table 6 - Smart meter packet drops in Scenario III 40
Table 7 - NIST recommended key sizes 43
Table 8 - `ID_KEY` Generation algorithm at the utility 48
Table 9 - Real time debugger data. 51
Table 10 - Average reduction in transmitted data in chain based network 54
Table 11 - Average reduction in received data in tree based network. 54
Table 12 - Average reduction in transmitted data in tree based network. 55
Table 13 - Average reduction in received data in tree based network. 55
Table 14 – Real time debugger data 63
Table 15 - Average number of Rx bytes 69
Table 16 - Average number of Tx bytes 69

Chapter 1 – Introduction

1.1 Overview of Smart Grids

The smart grid (SG) can be seen as the next generation electricity grid and is considered to be a revolution in the regime of existing power grids. The integration of Information and Communication Technology (ICT) with power grids has a substantial role in increasing the reliability and sustainability of energy production and distribution of power delivery networks. The establishment of two-way communication channels between different components of the smart grid and the utility offers several benefits ranging from early detection of blackouts to time-based rates of energy and demand management programs. It combines with the advanced sensor measurement technology, computer technology, information technology, control technology, communication technology and physical power. Figure 1 presents the communication network of the smart grid. [1]

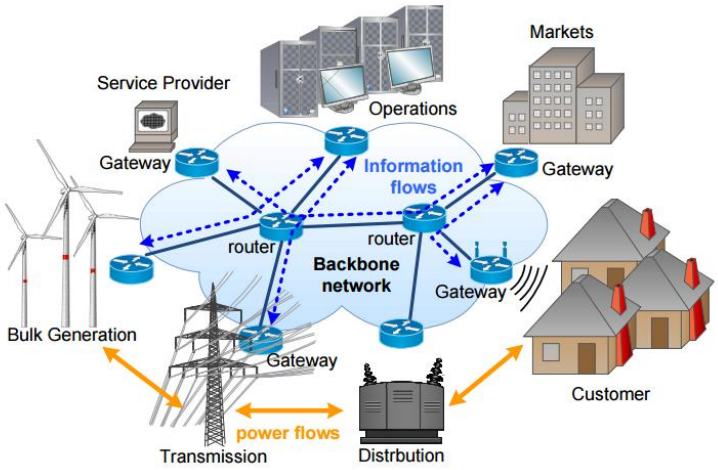

Figure 1 - Smart grid network architecture

According to NIST's smart grid model, the smart grid is composed of 7 logical domains: (1) Bulk generation, (2) Transmission, (3) Distribution, (4) Customer, (5) Markets, (6) Service provider, (7) Operations. The last 3 are related to information collection and power

management whereas the first 4 feature 2-way power and information flows (details covered in [2], [3], [4]).

1.2 Wireless mesh based AMI Architecture

The AMI network is considered the last mile of the smart grid. It connects the network of smart meters found in the Neighbourhood Area Network (NAN) to the utility's network via the NAN gateway. The AMI network carries various types of data, such as on demand and periodic electricity usage readings, updates of electricity pricing information, alerts about outages, meter firmware upgrades etc. Figure 2 depicts a wireless mesh based AMI network. [5]

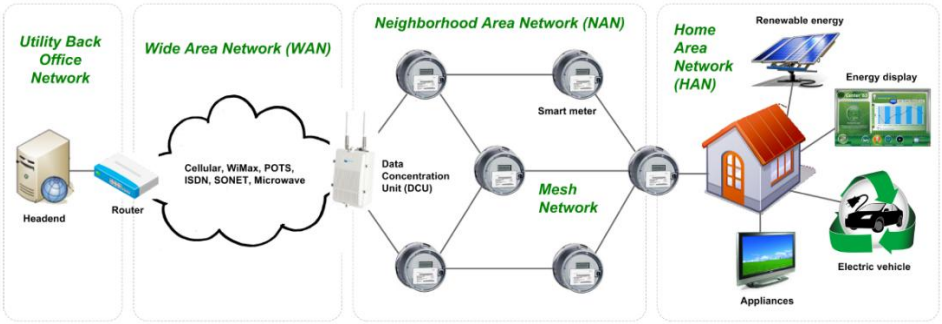

Figure 2 – Wireless mesh based AMI architecture

The Wide Area Network (WAN) connects the utility's data network to the DCUs installed in neighbourhoods after which the NAN connects the DCUs to the smart meters which are gateways to the Home Area Network (HAN). High bandwidth and long range communication channels are used within the WAN such as cellular (3G, 4G etc.), wireless (WiMAX), satellite etc. High bandwidth is links are required because of the high amount of traffic flowing through this network. Note that some messages require real-time packet delivery, such as control packets, where others can be delayed without adverse concerns.
NANs would require links with shorter range of communications and can be deployed either using wireless technologies (IEEE 802.15.4g, IEEE 802.11s etc.) or based on Power Line Communication. Other than transmitting its own data, each meter in the NAN acts as a relay and forwards data from its neighbouring meters in a multi-hop (hop-by-hop) manner.

The HAN enables the users to manage power consumption and on-demand requirements. As can be seen in Figure 2, it connects smart appliances to the smart meter, which can be seen as the HAN gateway.

In our work, a wireless mesh based topology is chosen for the NAN. The wireless mesh network has many advantages, the most important is that brings robustness to the network, since communication routes can automatically adapt when failures occur A transmitted packet traverses many smart meters before making its way to the DCU.

1.3 Overview of Smart Grid Security Requirements

Cyber security of the smart grid is of high importance and is considered as one of the highest priorities of the smart grid design. Compromising the communication and networking data systems of the smart grid will lead to the risk of compromising reliable and secure power system operations which is the ultimate objective of the smart grid. According to NIST and from a high level overview, there are 3 main objectives [6]: (1) Availability, (2) Integrity, (3) Confidentiality. For reliability purposes, (1) and (2) are the most important for the operation of the smart grid where (3) is the least critical for its reliability. However, (3) is important in systems that involve private user data and in system which deal with users such as in AMI networks.

Other than the specified objectives, NIST [6] specifies cyber security requirements to ensure the safe operations of the smart grid network which can be summarized by: (1) Attack detection and resilience operations, (2) Identification, authentication and access control, (3) Secure and efficient communication protocols (detailed in [1], [2]).
Compared to the cyber security requirements of the Internet, the smart grid imposes stricter security requirements for the purpose of obtaining secure and efficient information delivery for the purpose of keeping the power infrastructure safe. Table 1 summarizes this comparison [1].

Table 1 - Security requirements: SG vs Internet [1]

Security Functions	Smart Grid Communication Network	The Internet
Authentication and access control	Strictly enforced for all communication flows throughout the system	Mostly free end-to-end without access control
Attack detection and countermeasures	Essential and widely-deployed everywhere	Mainly for critical routers and servers
Every node	Basic cryptographic functions	No specification
Security for network protocols	From MAC-layer to application-layer security	From network-layer to application-layer security

1.4 Scope of Thesis

This thesis will focus on the security of the AMI network. Specifically, it will cover topics related to efficient and scalable key management and exchange by relying on identity based cryptographic primitives. To assess the performance of our proposed systems, we run simulations both on hardware and in software.

Chapter 2 starts by reviewing what's in the literature of smart grid security and focuses on the security of the AMI, the second part of the chapter reviews cryptographic primitives and concepts used throughout the thesis. Briefly, chapter 3 focuses on an identity based non-interactive efficient and scalable key distribution system targeted for generating pairwise symmetric keys to be used for link-layer authentication. In addition, a hardware based solution is presented to prevent the discovery and the leakage of the cryptographic keys stored on accessible memory locations. Chapter 4 continues the work of chapter 3 whereby a lightweight key update and delivery methods are proposed in order to secure update the private keys utilized by the identity based cryptosystem in an efficient fashion. The proposed method securely delivers the keys over the multi-hop NAN network with minimal overhead by exploiting the fact that there is room for key aggregation. Chapter 5 focuses on deriving keys for the purpose of ensuring a forward secure communication between the meters and the utility. The idea is based on merging identity based cryptography and Diffie-Hellman key exchange in order to derive forward secure keys.

Chapter 2 – Background Review

2.1 Literature Review

In this section papers related to smart grid security will reviewed. Papers and articles that are not directly related to our work will be passively reviewed, however papers and works that are relevant and directly related to ours will be actively discussed.

In [2], [3] authors present the smart grid infrastructure along with an overview of the motivations, requirements, challenges and solutions facing the smart grid. [7] presents the security requirements of a smart grid network particularly focusing on the communications challenges for achieving a futureproof and an interoperable smart grid network where security and privacy are emphasized. [4] presents a survey on cyber security requirements for smart grid communications where the authors summarize the security requirements and the vulnerabilities in smart grid communications. Focusing more about the security issues in the AMI, [8] discusses the general cyber security issues in AMIs from a high level. The authors of [9] discuss security attacks related to energy theft in AMI. Motivated by the analysis in [9], the authors continue their discussion in [10] to present a new methodology for penetration testing in AMIs. [11] discusses the seriousness of the remotely commanded off switch and the remote software upgrades introduced in smart meter. If an attacker hacks the control facility or hacks the meters might cause widespread blackouts. The authors suggest several strategies such as shared rate-limiting mechanisms to restrain the scale of an attack and local-override mechanisms to mitigate the attack's effects. [12] describes an efficient authentication mechanism based on Merkle hash trees that secures the communication between the HAN gateway and the NAN gateway. According to the authors, their proposed mechanism is able to defend against replay, message injection, message analysis and message modification attacks.

Authors of [13] propose a mechanism to efficiently handle the certificate revocation lists (CRLs) for the smart grid's AMI networks based on grouping of smart meters. In a few

words, their idea is to form CRLs for each group and send a single CRL within a group to reduce the delay and storage overhead. The group formation exploits the fact that AMI networks are based on the communications patterns among the smart meters where smart meters follow a path to the gateway, thereby having the need to talk to each of these nodes within its path. The authors implement their proposed methods in NS-3 to verify that their approach reduces the size of the CRLs and reduces the transmission delays.

In [5] the authors survey various threats facing AMI networks, particularly wireless mesh based AMIs, where they analyze several attack methods with the goal of carving the requirements of IDSs for the AMI network where. They depict a detailed attack tree (Figure 3) that highlights the attacker's objectives and the individual steps required involved in a particular attack. The orange nodes represent the objectives and the child nodes represent the individual attack steps.

They also performed the following 3 case studies based on the attack tree: (1) Distributed Denial of Service (DDoS) Attack Against the Data Collection Unit, (2) Stealing Customer Information, (3) Sending Remote Disconnect Commands Through the Data Collection Unit. After analysing the attack tree and understanding the information required for the attack detection, the authors propose a hybrid IDS mechanism that relies on both a centralized intrusion detection system and embedded sensors within meters.

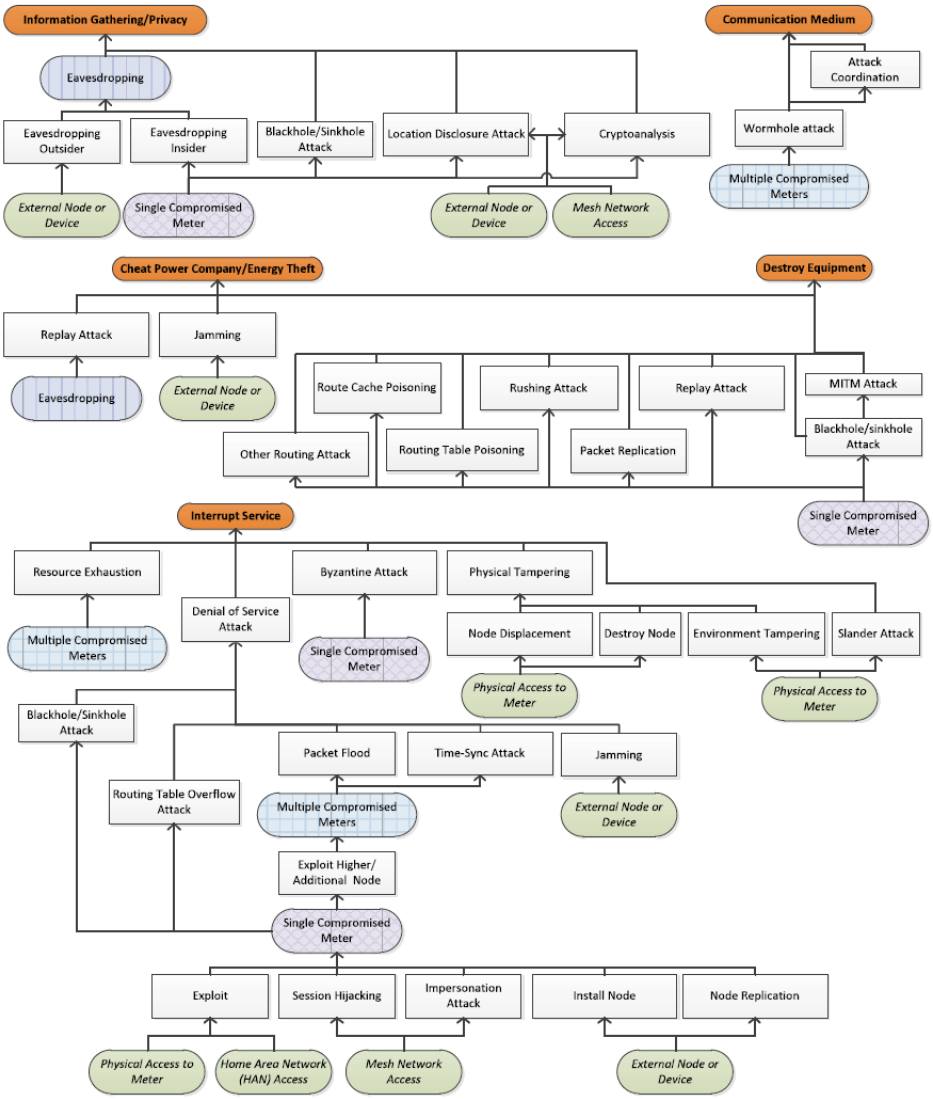

Figure 3 - AMI attack tree in [5]

The authors of [14] present a security protocol for the AMI network which they call Integrated Authentication and Confidentiality (IAC). Their protocol provides data aggregation, trust services, data privacy and data integrity by relying on end-to-end along with hop-by-hop security mechanisms. They assume the AMI network is a wireless mesh based network that connects the meters to a feeder. The proposed scheme can be seen dissected into 3 parts: (1) Initialization process, (2) Meter reading collection process, (3) Control message distribution process. (1) is brought forward when a new meter asks to join a network which is summarized in figure 2. Before a meter joins the network, it must be

verified by the remote authentication server where neighboring authenticating meters' aid in this process by relaying the authentication process messages. After completing this process, the meter would be authenticated at the authentication server and receive mutual secret key to be used in the message authentication code (MAC) generation/verification with its neighboring meter.

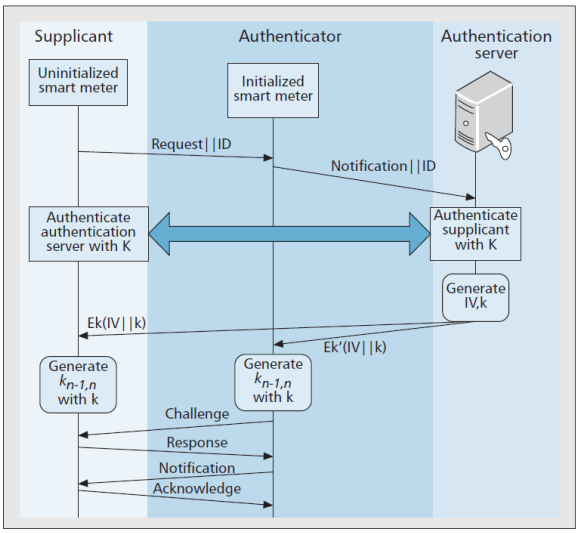

Figure 4 - Initialization and Authentication in [14]

In (2) the meters work in a collaborated fashion to forward the data collected from the meters to the feeder in an aggregated manner. Neighbouring meters verify the authenticity of the data by generating and verifying message authentication codes (keys were obtained in (1)) whereby the data on which MAC is generated is encrypted using a secret key shared between the meter and the utility. This is depicted in Figure 5where K_i is a secret shared between utility and meter, k_i is a mutual shared key between neighbouring meters, Ri is the generated data from the meter and C the MAC generating block and M_i the encryption of Ri. At the last hop, M_1… M_n denote the aggregated data to be sent to the feeder. Process (3) is when the utility needs to send control messages to the n meters and it can be seen as the reverse process of (2). The simulations show that the proposed protocol compared to the basic security scheme adds little overhead and is more efficient.

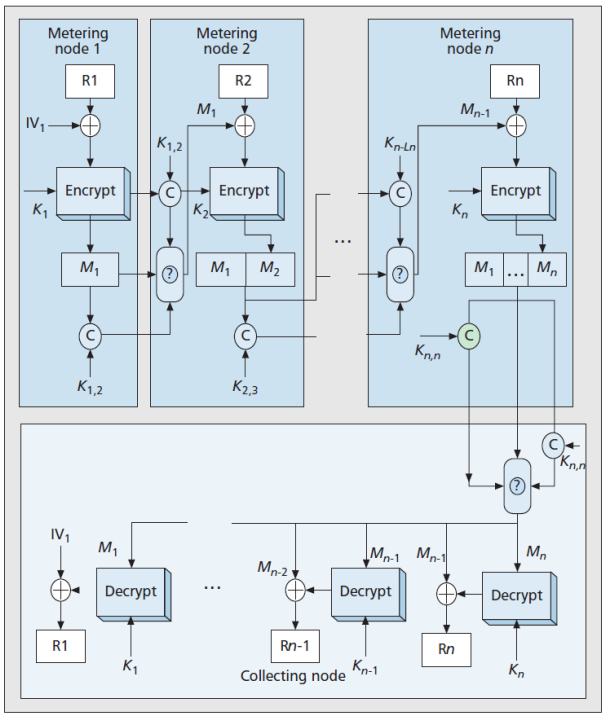

Figure 5 - Meter reading collection process in [14]

The authors of [15] propose the use of an identity based signcryption system to provide a zero-configuration encryption and authentication solution for end-to-end secure communications in a smart grid network. Compared to other public-key systems, the proposed system is scalable and ensures secure communications without the need for complex setup from the users, thus called a zero-configuration signcryption scheme. The system relies on Boneh-Franklin identity-based encryption scheme for encryption and identity-based signature scheme of [5] for authentication where the unique manufacturing serial number of the meter is used as the ID for all cryptographic functions. During the manufacturing phase, public parameters are inserted into the node, however before initiating any communication with any party in the network, a node should communicate securely with the key generation system (KGS) in order to obtain its private key. For the encryption, the Tate pairing is used on an elliptic curve s.t the pairing is used to generate a shared secret key between the sender and the receiver. As for the authentication, the same pairing is used to generate and verify signatures. The authors admit that the most time consuming process is the calculation of the Tate pairing and propose key caching schemes

to reduce the number of Tate calculations. The authors also briefly discuss about key escrow, key revocation and key update in their proposed scheme.

Smart grid communication requires a secure authentication framework. This framework should minimally increase the overhead placed on the messages exchanged amongst the smart meters. A lightweight message authentication scheme is proposed in [16]. The proposed scheme is based on mutual authentication and the establishment of shared session keys using Diffie-Hellman (DH) key exchange protocol. The shared session key derived from DH is later used for authenticating the data exchanged based on a hash-based authentication technique. The communicational network devised in the paper is divided into several hierarchical networks: (1) the neighbourhood area network (NAN), (2) the building area network (BAN) and (3) the home area network (HAN). The smart meters are assumed to be HAN gateways with limited resources in terms of memory and computational power. The meters connect the HAN to the BAN gateway which are thought to have more resources (10x) than smart meters. The BAN gateways connect the BAN to the NAN gateways which can be seen as portals to the utility. The authors believe that a meter might transmit each message within an interval of one second which leads to an increase in the amount of data packets that need to be authenticated by a BAN gateway. In addition, there exists processing delays at the meters for decrypting the received encrypted packets thereby increasing the communication latency. The authors presume that conventional public key infrastructure schemes aren't adequate for the rigid timing requirements of smart grid communications, thereby requiring a lightweight verification algorithm for smart grid communications. The proposed scheme achieves source authentication and message integrity, low communication overhead with fast verification, privacy preservation and forward secrecy. By forward secrecy it is meant that a session key derived from a set of long-term keys will not be revealed if one of the long-term secret keys is compromised at a later time.

Assuming that the each of the HAN gateway and BAN gateway have private and public key pairs, an initial handshake based on the DH key exchange takes place between the 2 gateways. After the handshake, a symmetric shared session key is computed which is used as the key for a

hash based message authentication. The scheme is depicted in Figure 6. $\{a, b\}$ and $\{g^a, g^b\}$ are volatile private and public keys generated by each HAN gateway and BAN GW respectively and used in the DH key exchange. Pub_{BAN_GW} and Pub_{HAN_GW} are the public keys used for encrypting the packets transmitted. After the exchange of the 1st 3 packets, both parties are able to compute K_i the symmetric shared key.

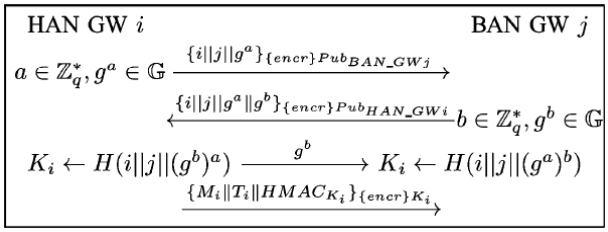

Figure 6 - Lightweight authentication scheme in [16]

Physical Unclonable Functions (PUFs) evolved from the fact that different instances of the same hardware result in different behavioural characteristics. This is due to the random and uncontrolled discrepancies that result at the circuit production stage.

Because of that, they are viewed as primitives to derive secrets from complex physical characteristics of integrated circuits which are extremely difficult to predict or extract. In [17], authors discuss the usage of PUFs for low-cost integrated circuit authentication and generation of volatile secret keys used in crypto. The authors also introduce ring oscillators (RO) based PUF circuit designs. Compared to other designs, this type of PUF design has advantages in the ease of implementation and reliability over previously proposed designs. An RO PUF (Figure 7) is made of several identical ring oscillators each of which oscillates with a specific frequency. However, due to fabrication variations, the frequency of oscillations varies. Fixed order of oscillator pairs are selected for the purpose of generating a constant number of bits where their frequencies are compared in order to produce a single output bit. The authors discuss on how the reliability of the PUF circuit can be enhanced, whereby errors or bit-flips result from environmental changes. By picking oscillators that have base frequencies far apart from each other, outputs that are less likely to flip than the

case where the frequencies are closer. The authors then discuss about several ways to create challenge-response pairs based on the ring oscillator PUF design.

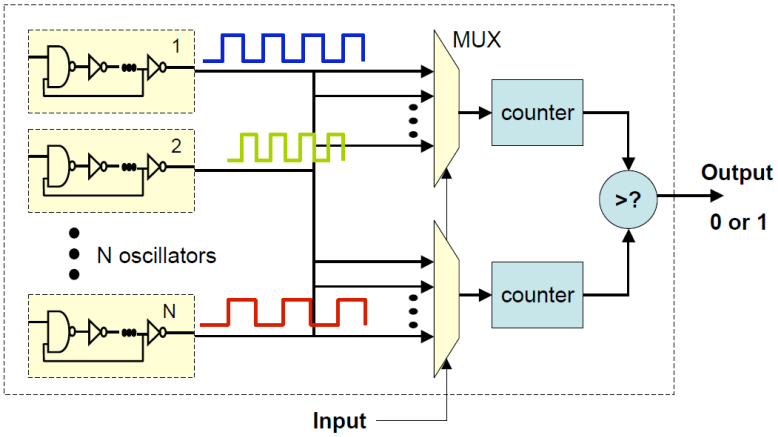

Figure 7 – RO based PUF in [17]

As examples, the paper covers 2 applications of ring oscillator based PUF. The 1st is using it for low-cost authentication of integrated circuits without the usage of crypto. This application can be seen useful in resource constrained environments such as in Radio Frequency Identification (RFID). The 2nd application that is discussed is the usage of the ring oscillator PUF in cryptographic key generation. Because of noise on output, the output is likely to be marginally different on each run even when the same challenge is used on the same integrated circuit. However, the key used in cryptosystems require every bit to be constant. The authors solve the issue by relying on error correction processes whereby a 2 component system is devised to perform this task. In the 1st component (Figure 8), the error correcting blocks ensures the output is consistent even if there are significant environmental changes (voltage and temperature changes). During the initialization stage, the output of the PUF is used by the error correcting codes (ECC) to generate an error correct syndrome which is saved and used later. The syndrome contains information that allows for correcting bit-flips in the output.

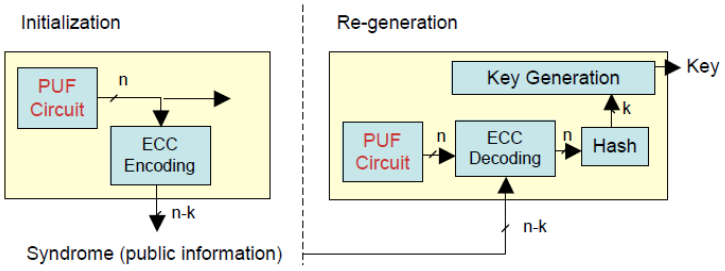

Figure 8 - Key generation in [17]

The 2nd step is to use the output of the ECC and input it into a hash block in order to obtain a fixed desired length to be used as a crypto key. As for crypto systems that require keys with special properties (RSA cryptosystem) the key generated from the hash can be used as a seed for the RSA key generation.

The authors evaluate their work on an FPGA to show that ring oscillator based PUFs can be used for authentication and for secret key generation. They show that if BCH (127,64,21) code is to be used, the probability of failing to regenerate the same key is less than 5×10^{-11} for 10 errors in a 127 bit PUF.

The idea of integrating Physical Unclonable Functions (PUFs) into smart meters was introduced in [18]. The paper addresses the problem of crafting a key management scheme that is able to attain secure end-to-end communication in the AMI. Their solution provides an efficient approach to manage keys and a strong authentication mechanism. It is based on inexpensive PUF technology that provides a hardware based strong authentication mechanism. In their work PUF devices are utilized to generate symmetric keys and access level passwords for smart meters, providing protection against key leakage at the hardware level as the master secret key is never stored in memory. The PUF device is used in a feedback loop in a system on chip design as seen in Figure 9. C_i and R_i are the given challenge and the obtained response from the PUF; ECU is the error correcting unit; Reg is a register that stores the initial challenge and is overridden by the new response and CC is the cryptographic block. An ECU is used to perform error correction on a noisy PUF response s.t during operation every time the same challenge is given to the PUF the same response is obtained.

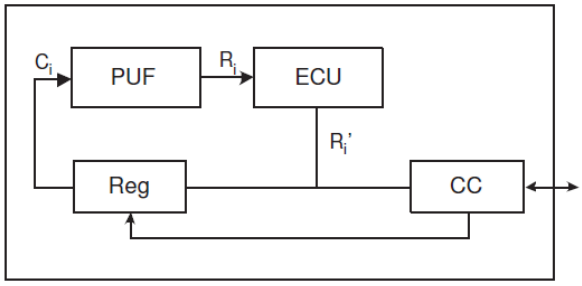

Figure 9 – PUF system on chip design in [18]

Their scheme involves the following procedures to provide end-to-end meter-to-utility security: (1) Initialization, (2) Smart meter registration, (3) Refreshing PUF secrets, (4) Smart meter authentication, (5) Smart meter key/password generation and re-key, (6) Adding/revoking smart meters. From a high level, (1) and (2) can be seen as processes performed by the utility before placing the meters on the field. They involve defining cryptographic hash functions, defining challenges and storing them within the meter along with computing the Pedersen commitment to be used in the authentication scheme etc. (3-6) involves process that happen when the meters are at the field. (3) involves updating the PUF secrets by running the PUF feedback loop and reporting back the outputs to the utility by encrypting them with the old PUF response. (4) involves authenticating the meter to the utility before proceeding with any further communication. It based on the PUF response along with zero knowledge proofs.

2.2 Cryptographic Primitives

The discussions, formulas and figures of this subsection is based on the works of [19], [20], [21], [22].

2.2.1 Hash Functions & HMAC

Hash functions are functions that take an input of some length and return a shorter output with a fixed length. Let $h = H(x)$ denote a hash function with input x, where x is considered the preimage of h. Table 2 lists the requirements for Cryptographic hash functions.

Table 2 - Requirements for cryptographic hash functions

Requirement	Description
Output size is fixed	H produces output with fixed lengths
Input size is variable	H accepts data blocks of arbitrary sizes
Collision resistance	Finding $\{x, y\}$ s.t $H(x) = H(y)$ is computationally infeasible
Preimage resistance	Given h it is computationally infeasible to find y s.t $H(y) = h$
Second preimage resistance	Given x, it is computationally infeasible to find $x \neq y$ s.t $H(x) = H(y)$
Pseudorandomness	Output of H meets standard tests for pseudorandomness

One of the most important applications of cryptographic hash functions is in message authentication where message authentication is a mechanism to verify the integrity of a message. This is achieved using message authentication codes (MACs) which are seen as cryptographic checksums. This assumes that both communicating parties (generator and verifier) share a common secret K s.t

$MAC = C(K, M)$ where M is the input message and C is the MAC generating function. A popular MAC algorithm is the HMAC (Hashed MAC) where a cryptographic hash function (such as SHA-1) is at the heart of the MAC generation. HMAC can be expressed as

$$HMAC(K, M) = H[(K^{+} \oplus opad) \,||H[K^{+} \oplus ipad)||M]]$$

Where K^{+} is the padded version of K with zeros on the left s.t the padding results in b bits length, where b is the number of bits in a block; ipad = 0x36 repeated b/8 times; opad = 0x5C repeated b/8 times; RFC 2104 describes HMAC in a much more detailed manner.

2.2.2 Symmetric Key Cryptography

The most basic form of cryptography is the one in which all parties use the same key to perform a cryptographic operation on the message. Both sender and receiver use the same key, which is why this system is known as symmetric key encryption. There is one big disadvantage with this system; the key has to be distributed without being intercepted along the way by a third party, as the security of this encryption is largely dependent on this.

Mathematically, a symmetric cipher uses a key k chosen from a space (i.e., a set) of possible keys K to encrypt a plaintext message m chosen from a space of possible messages M, and the result of the encryption process is a ciphertext c belonging to a space of possible ciphertexts C. Encryption may be viewed as a function

$$e: K \times M \to C$$

whose domain $K \times M$ is the set of pairs (k, m) consisting of a key k and a plaintext m and whose range is the space of ciphertexts C. The decryption function "undoes" the result of the encryption function and can be stated as the function

$$d: K \times C \to M$$

Such that $d(k, e(k, m)) = m$ for all $k \in K$ and all $m \in M$. For convenience, let the dependency on k be denoted by a subscript. In other words, for every key k, the function d_k is the inverse function of the function e_k. In particular, this means that e_k must be one-to-one, since if $e_k(m) = e_k(m')$, then

$$m = d_k\big(e_k(m)\big) = d_k\big(e_k(m')\big) = m'$$

Symmetric crypto is mostly categorised as stream cipher or block cipher. A stream cipher system will encrypt each bit of data at a time in serial, and there'll probably be a system that changes the key each time a bit is encrypted. Block cipher systems work on set blocks of plaintext, encrypting each one individually with the key.

2.2.3 Asymmetric Key Cryptography

If Alice and Bob want to exchange messages using a symmetric cipher, they must first mutually agree on a secret key k. This is fine if they have the opportunity to meet in secret or if they are able to communicate once over a secure channel. But what if they do not have this opportunity and if every communication between them is monitored by their adversary Eve? This problem gave the birth to what is called public key (or asymmetric) crypto. In this system, one key is distributed openly and used to encrypt data, and the other key is kept secret and used to decrypt the same data. This is made possible with one-way mathematical functions; calculations that are almost impossible to reverse. So while the two keys are both mathematically related, and one (the public key) can be distributed to anyone, it will be

extremely difficult to determine the secret key from this. To formulate it mathematically, assume that there are spaces of keys K, plaintexts M, and ciphertexts C. Unlike the symmetric key crypto's case, an element k of the key space is really a pair of keys $k = (k_{priv}, k_{pub})$ termed private key and the public key. For each public key k_{pub} there is a corresponding encryption function

$$e_{k_{pub}}: M \to C$$

and for each private key k_{priv} there is a corresponding decryption function

$$d_{k_{priv}}: C \to M$$

such that $d_{k_{priv}}\left(e_{k_{pub}}(m)\right) = m$ for all $m \in M$ where $(k_{priv}, k_{pub}) \in K$.

The fact that the encryption and decryption keys k_{pub} and k_{priv} are different makes the cipher asymmetric. If an asymmetric cipher is to be secure, it must be difficult for Eve to compute the decryption function $d_{k_{priv}}(c)$ even if he knows k_{pub}. Using the above formulation, Bob sends k_{pub_B} to Alice over an insecure channel monitored by Eve. Alice generates $c = e_{k_{pub\_B}}(m)$ and sends to Bob without distressing that Eve would be able to decrypt the message.

Whitfield Diffie and Martin Hellman formulated the concept of a public key encryption system and made several innovative contributions to this new field. Their main contribution was the description of a secure public key cryptosystem and its associated components: one-way functions and trapdoor information.

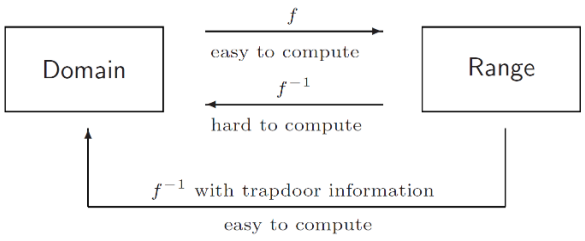

Figure 10 - Representation of one-way trapdoor function

A one-way trapdoor function f is invertible and easy to compute, however its inverse f^{-1} is "hard" to compute unless a trapdoor information is present, where a trapdoor is a piece of auxiliary information that allows the inverse to be easily computed. In section 1.2, k_{priv} is a

trapdoor function for the function $e_{k_{pub}}$ since without the presence of this piece of information it is hard to compute the inverse of $e_{k_{pub}}$. Throughout this text, a function is said to be "hard" to compute if its runtime takes unreasonable amount of time.

2.2.4 Discrete Log Problem (DLP)

The DLP is a mathematical problem that arises in many settings, like the modulus p version defined this section and the elliptic curve version that is treated in upcoming sections. The first published public key construction in [36] is based on DLP in a finite field $\mathbb{F}_p$, with p being a large prime number.

Primitive Root Theorem: Let p be a prime number. Then there exists an element $\mathrm{g} \in \mathbb{F}_{\mathrm{p}}^{*}$ whose powers give every element of $\mathbb{F}_{\mathrm{p}}^{*}$:

$$\mathbb{F}_{\mathrm{p}}^{*} = \{1, \mathrm{g}, \mathrm{g}^2, \dots, \mathrm{g}^{\mathrm{p}-2}\}$$

g is called the primitive root of generator of $\mathbb{F}_{\mathrm{p}}^{*}$.

Fermat's Little Theorem: Let p be a prime number and let a be an integer. Therefore,

$$\mathrm{a}^{\mathrm{p}-1} \equiv \begin{cases} 1 \pmod{\mathrm{p}} & \text{if } \mathrm{p} \nmid \mathrm{a} \\ 0 \pmod{\mathrm{p}} & \text{if } \mathrm{p} \mid \mathrm{a} \end{cases}$$

Definition. Let g be a primitive root for $\mathbb{F}_{\mathrm{p}}$, and let h be a nonzero element of $\mathbb{F}_{\mathrm{p}}$. The DLP is the problem of finding an exponent x such that

$$\mathrm{g}^{\mathrm{x}} \equiv \mathrm{h} \pmod{\mathrm{p}}$$

where x is the discrete logarithm of h to the base g, symbolized by $\log_{\mathrm{g}}(\mathrm{h})$.

DLP is a well-posed problem, i.e. finding an exponent x such that $g^x = h$. However, if there is one solution, then based on Fermat's little theorem there are infinitely many solutions. Henceforth, if x is a solution to $g^x = h$, then $x + k(p-1)$ is also a solution for all values of k, because

$$g^{x+k(p-1)} = g^x (g^{p-1})^k \equiv h.1^k \equiv h \ (mod\ p)$$

The fastest algorithm known to solve the DLP in $\mathbb{F}_p^*$ is the index calculus has an order of $\mathcal{O}\left(e^{c\sqrt{(\log p)(\log\log p)}}\right)$, thus it takes **sub-exponential** time to solve it.

2.2.5 Diffie–Hellman key exchange

Alice and Bob want to share a secret key for use in a symmetric cipher over an insecure. This insecure channel is monitored by an adversary named Eve. How is it possible for Alice and Bob to share a key without making it available to Eve? The Diffie–Hellman key exchange algorithm solves this problem by relying on the difficulty of solving the discrete logarithm problem in $\mathbb{F}_p^*$.

First, Alice and Bob agree on a large prime p and a nonzero integer g modulo p. Alice and Bob make the values of p and g public knowledge where Eve might gain access to this public info. Next, each Alice and Bob pick a secret integer a and b respectively. Alice and Bob use their secret integers to compute respectively

$$A \equiv g^a \bmod p;\ B \equiv g^b \bmod p$$

They next exchange these computed values where Alice sends A to Bob and Bob sends B to Alice. Eve gets to see the of A and B since they are sent over an insecure channel. Lastly, Alive and Bob use their secret integers to compute respectively the shared key

$$A' \equiv B^a \bmod p;\ B' \equiv A^b \bmod p$$

A' and B' end up being the same values since

$$A \equiv B^a \equiv (g^b)^a \equiv\ (g^a)^b \equiv\ A^b \equiv B'\ (\bmod p)$$

If Eve, knowing the values of $(g^a,\ g^b,\ a,\ b)$, can solve the DLP, she can find a and b, after which it is easy to compute the share key g^{ab}. But DLP is a very hard problem which makes the key exchange safe. However, solving DLP is not the only means of computing g^{ab}. The security of the shared key reposes on the difficulty of solving the Diffie-Hellman Problem (DHP) which is no harder than DLP.

Definition. *Let p be a prime number and g an integer. The DHP is the problem of computing the value of* $g^{ab} \pmod p$ *from the known values of* $g^a \pmod p$ *and* $g^b \pmod p$.

2.2.6 Elliptic Curves in Finite Fields

In order to apply the theory of elliptic curves to cryptography, there is the need for elliptic curves whose points have coordinates in a finite field $\mathbb{F}_p$. Such an elliptic curve over $\mathbb{F}_p$ can be defined as the set of solutions to a Weierstrass equation

$$E: Y^2 = X^3 + AX + B \quad \text{s.t } A, B \in \mathbb{F}_p \text{ satisfying } \Delta E = 4A^3 + 27B^2 \neq 0$$

The points on E with coordinates in $\mathbb{F}_p$ are denoted by

$$E(\mathbb{F}_p) = \{(x, y) : x, y \in \mathbb{F}_p \text{ satisfy } y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}$$

$\mathcal{O}$ is a point located at infinity. Elliptic curves over $\mathbb{F}_2$ are actually quite important in cryptography.

Assume $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are points in $E(\mathbb{F}_p)$. We define the sum $P + Q$ to be the point (x_3, y_3) in $E(\mathbb{F}_p)$ obtained by applying the elliptic curve addition algorithm.

Elliptic Curve Addition Algorithm: Let P_1 and P_2 be points on $E(\mathbb{F}_p)$.

1. *If* $P_1 = \mathcal{O}$ *then* $P_1 + P_2 = P_2$
2. *Otherwise, if* $P_2 = \mathcal{O}$ *then* $P_1 + P_2 = P_1$
3. *Otherwise, write* $P_1 = (x_1, y_1)$ *and* $P_2 = (x_2, y_2)$
4. *If* $x_1 = x_2$ *and* $y_1 = -y_2$, *then* $P_1 + P_2 = \mathcal{O}$
5. *Otherwise, define* λ *by*

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } P_1 \neq P_2 \\ \dfrac{3x_1^2 + A}{2y_1} & \text{if } P_1 = P_2 \end{cases}$$

and let $x_3 = \lambda^2 - x_1 - x_2$; $y_3 = \lambda(x_1 - x_3) - y_1$, *then* $P_1 + P_2 = (x_3, y_3)$

Repeated addition is represented as multiplication of a point by an integer s.t. $nP = P + P + \cdots + P$ (n copies of P).

The set of points $E(\mathbb{F}_p)$ is a finite set, since there are only finitely many possibilities for the X-and Y-coordinates. Namely there are p possibilities for X and for each X, there are at most 2 solutions. Along with the point $\mathcal{O}$ it can be seen that $\#E(\mathbb{F}_p)$ is at most $2p + 1$. This number is considerably larger than its true size. According to Hasse's theorem

$$\left| \#E(\mathbb{F}_p) - (p + 1) \right| \leq 2\sqrt{p}$$

2.2.7 Elliptic Curve Discrete Log Problem (ECDLP)

Similar to the DLP in $\mathbb{F}_p^*$, a discrete log problem can be formalized with an elliptic curve E over $\mathbb{F}_p^*$.

Definition. *Let E be an elliptic curve over the finite field $\mathbb{F}_p^*$ and let P and Q be points in $E(\mathbb{F}_p)$. The ECDLP is the problem of finding an integer n such that $Q = nP$. By analogy with the DLP for $\mathbb{F}_p^*$, the integer n denoted by*

$$n = log_P(Q)$$

is called the elliptic curve discrete logarithm of Q w.r.t to P.

Although the recovery of n given the points Q and P in $E(\mathbb{F}_p)$ is very hard, there are efficient methods to calculate Q given n and P. One of the most efficient ways is to use the Double-and-Add algorithm. The algorithm is summarized in Table 3.

Table 3 - Double-and-Add algorithm

	Input. Point $P \in E(\mathbb{F}_p)$ and integer $n \geq 1$.
1.	Set $Q = P$ and $R = \mathcal{O}$.
2.	Loop while $n > 0$.
3.	If $n \equiv 1 \pmod 2$, set $R = R + Q$.
4.	Set $Q = 2Q$ and $n = \lfloor n/2 \rfloor$.
5.	If $n > 0$, continue with loop at Step **2**.
6.	Return the point R, which equals nP.

The main reason that elliptic curves are used in cryptography is the fact that there are no index calculus algorithms known for the ECDLP and because there are no general algorithms that solve the ECDLP in less than $\boldsymbol{O(\sqrt{p})}$ steps thus it is considered to have an **exponential** runtime.

2.2.8 Elliptic Curve Diffie-Hellman Key Exchange (ECDH)

Alice and Bob agree to use a particular elliptic curve $E(\mathbb{F}_p)$ and a point $P \in E(\mathbb{F}_p)$. Alice and Bob choose secret integers n_A and n_B respectively. They also calculate the public values $Q_A = n_A P$ and $Q_B = n_B P$ and exchange the calculated values. Note that Q_A and Q_B are not single numbers. They are tuples of the form $Q = \{x,y\}$. After the exchange, both parties can calculate a shared symmetric secret key which can be utilized by any symmetric cipher. This shared key is calculated as follows:

$$n_B Q_A = (n_A n_B)P = n_A Q_B$$

A possible means for Eve to discover the shared secret is by solving the ECDLP since solving this problem reveals n_A which can be used to compute $n_A Q_B$. However, the precise problem that Eve needs to solve is the elliptic analogue of the Diffie–Hellman problem, which is the elliptic curve Diffie-Hellman problem (ECDHP) in this case.

Definition. *Let $E(\mathbb{F}_p)$ be an elliptic curve over a finite field and let $P \in E(\mathbb{F}_p)$. The ECDHP is the problem of computing the value of $n_1 n_2 P$ from the known values of $n_1 P$ and $n_2 P$.*

2.2.9 A Notion of Bilinear Pairings on Elliptic Curves

In linear algebra the dot product is a bilinear pairing on the vector space $\mathbb{R}^n$ where this can be formulated as $\beta(\boldsymbol{v},\boldsymbol{w}) = \boldsymbol{v}.\boldsymbol{w} = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n$.
This can be seen as a pairing in the sense that it takes a pair of vectors and returns a number, and it is bilinear in the sense that it is a linear transformation in each
of its variables. That is, for any vectors $\boldsymbol{v_1}, \boldsymbol{v_2}, \boldsymbol{w_1}, \boldsymbol{w_2}$ and real numbers a_1, a_2, b_1, b_2 it is found that

$$\beta(a_1\boldsymbol{v_1} + a_2\boldsymbol{v_2}, \boldsymbol{w}) = a_1\beta(\boldsymbol{v_1}, \boldsymbol{w}) + a_2\beta(\boldsymbol{v_2}, \boldsymbol{w}),$$

$$\beta(\boldsymbol{v}, b_1\boldsymbol{w_1} + b_2\boldsymbol{w_2}) = b_1\beta(\boldsymbol{v}, \boldsymbol{w_1}) + b_2\beta(\boldsymbol{v}, \boldsymbol{w_w})$$

The bilinear pairings in cryptography are similar in that they take as input two points on an elliptic curve and give as output a number. However, the bilinearity condition is slightly

different, because the output value is a nonzero element of a finite field. For most of these applications it is necessary to work with finite fields $\mathbb{F}_{p^k}$ of prime power order.

2.2.10 Bilinear Maps

Consider two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q where we let $\mathbb{G}_1$ to be an additive group and $\mathbb{G}_2$ a multiplicative. $\mathbb{G}_1$ will be an elliptic curve group on a finite field and $\mathbb{G}_2$ will be a subgroup of the multiplicative group of a related finite field. A bilinear map $\hat{\boldsymbol{e}}$ can be considered as follows

$$\hat{\boldsymbol{e}}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$$

The Weil and Tate pairings are used to implement such pairings. They involve fairly complex mathematics and their details will not be covered. However, they can be dealt with abstractly, that is by only using the group structure and mapping properties.

Useful bilinear maps have 3 properties:

1. **Bilinearity**: $\forall P, Q \in \mathbb{G}_{1,} \forall a, b \in \mathbb{Z}_q^*$

   $\hat{\boldsymbol{e}}\,(aP, bQ) = \hat{\boldsymbol{e}}(P, Q)^{ab}$

2. **Non-Degeneracy**: $\forall P \in \mathbb{G}_1\ s.t\ P \neq 0 \Rightarrow \hat{\boldsymbol{e}}(P, P) = \mathbb{G}_2$, that is for

   $P \neq 0 \Rightarrow \hat{\boldsymbol{e}}(P, P) \neq 1$

3. **Computability**: $\hat{\boldsymbol{e}}$ is efficiently computable

The realization of bilinear map results with 2 important complexity implications:

1. **MOV reduction:** DLP in $\mathbb{G}_1$ is no harder than the DLP in $\mathbb{G}_2$. To prove this, assume Q to be an element of $\mathbb{G}_1$ s.t $Q = xP$. It can be seen that $\hat{\boldsymbol{e}}(P, Q) = \hat{\boldsymbol{e}}(P, xP) = \hat{\boldsymbol{e}}(P, Q)^x$, therefore computing the log of $\hat{\boldsymbol{e}}(P, Q)$ in $\mathbb{G}_2$ to the base $\hat{\boldsymbol{e}}(P, P)$ yields x.
2. **Decisional Diffie Hellman (DDH) is easy in $\mathbb{G}_1$:** The DDH problem involves distinguishing $\langle P, aP, bP, cP\rangle$ and $\langle P, aP, bP, abP\rangle$ where $a, b, c \in \mathbb{Z}_q^*$ and $P \in \mathbb{G}_1$. To prove this, providing $P, A, B\ and\ C$ to the distinguisher, she computes $x_1 = \hat{\boldsymbol{e}}(A, B)$ and $x_2 = \hat{\boldsymbol{e}}(P, C)$. If $x_1 = x_2$, then the tuple is $\langle P, aP, bP, abP\rangle$ for $C = abP$, else it is $\langle P, aP, bP, cP\rangle$. This is because for $C = abP$,

   $\hat{\boldsymbol{e}}(A, B) = \hat{\boldsymbol{e}}(aP, bP) = \hat{\boldsymbol{e}}(P, P)^{ab} = \hat{\boldsymbol{e}}(P, C)$

Since DDH is easy in $\mathbb{G}_1$ it cannot be used to build a secure cryptosystem. A variant of DDH has to be used called Bilinear Diffie Hellman (BDH).

BDH Problem: $\mathbb{G}_1$ and $\mathbb{G}_2$ are groups of prime order q. $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a bilinear map and P is a generator of $\mathbb{G}_1$. The BDH problem states that given $\langle P, aP, bP, cP \rangle$ with $a, b, c \in \mathbb{Z}_q^*$ compute $X = \hat{e}(P,P)^{abc} \in \mathbb{G}_2$. An algorithm $\mathcal{A}$ has advantage ϵ in solving BDH in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ if

$$\Pr[\mathcal{A}(P, aP, bP, Cp) = \hat{e}(P,P)^{abc}] \geq \epsilon(k)$$

2.2.11 A notion of Identity-based cryptosystems

One would like a public key cryptosystem in which the user's public key can be chosen by the user. For example, Bob might use his phone number as his identity-based public key and then anyone who knows his phone number automatically knows his public key. This would remove the need to exchange data in order to obtain a party's public key and simplify life. This system would also need a trusted authority (TA) to generate the private keys associated with the public key. The idea of ID-based cryptography was initially described by Shamir in 1984 and a practical ID-based system was devised by Boneh and Franklin in 2001.

Unlike X.509 and many other traditional public key cryptosystems, it is the Certificate Authority's (CA) responsibility to supply both public and private keys. Although public keys are not sensitive information, the CA does so in order to ensure that Alice acquires the genuine public key of Bob instead of bogus ones provided by Eve. For such purposes, prior to communicating with Bob, Alice has to acquire a certificate from CA containing the public key of Bob.

Chapter 3 – Identity Based Scalable Key Distribution for AMI networks

3.1 Introduction and Motivation

Data exchanged between smart meters and the utility travel along a certain path and traverse several hops before reaching the final destination. If an adversary is present along this path, she can eavesdrop or tamper with the data on transit. Eavesdropping on transmitted data from the meters may breach users' privacy. Tampering with data might lead to serious consequences due to falsified meter readings. This can lead to potential company revenue loss and possibly blackouts due to supply/demand mismatch if carried on a large scale. An adversary may also flood the network with bogus packets to create a Denial of Service attack (DoS) preventing meter or utility packets from reaching the other end. She may also maliciously craft packets to exploit certain meter functionalities such as the 'remote disconnect' functionality.

For such reasons, end-to-end data exchanged between meters and the utility should be (a) encrypted when sensitive information is present to ensure confidentiality of the transit data and (b) authenticated to ensure integrity and to verify the source of data. However, end-to-end security mechanisms alone do not guard against all the attack vectors.

The data-link layer security is an essential element in the security of the AMI. By having a weak link-layer security, a network can be vulnerable to many attack vectors such as unauthorized access to the network, spoofing and tampering attacks, Denial of Service attacks (DoS) and different types of Layer 2 routing attacks.

Several security frameworks have been established in the context of wireless sensor networks to ensure the security of the link layer some of which were presented in the section on literature review. The security of all the frameworks lies on the strength of the

cryptographic primitives, which by themselves are dependent on the strength of the keys used.

This indeed raises the need for a strong and scalable key exchange system in the context of AMIs due to the huge amount of meters present on the network. The usage of PKIs along with certificates and CAs are discouraged in the context of the smart grids due to challenges and issues in scalability and complexity [23], [24].

Identity based non-interactive key distribution (ID-NIKD) [25] can be seen as a prominent solution to the problem faced. It provides secure symmetric keys to neighbouring smart meters without having the need to exchange extra packets for key negotiation. It also removes the need to communicate to a central server every time a node or a packet needs to be authenticated.

In addition, cryptosystems implemented on accessible hardware require careful management of secret keys to avoid key leakages at the hardware level.

In this chapter, we propose a security framework for wireless mesh-based AMI. It relies on merging the benefits of hardware-based symmetric-key cryptography and identity-based cryptography to enable an efficient and scalable framework with the following features:

1. Secure and efficient end-to-end and meter-to-utility communication based on symmetric keys generated from Physical Unclonable Functions (PUFs).
2. Scalable ID-based non-interactive key distribution mechanism for the security of the link layer. Keys negotiated between the neighbouring meters are used for the ID-based authentication of the transit packets.
3. Secure and leakage-free storage of private keys based on symmetric volatile keys generated by PUFs.

3.2 Physical Unclonable Functions

The idea of Physical Unclonable Functions (PUFs) evolved from the fact that different instances of the same hardware result in different behavioural characteristics. This is due to the random and uncontrolled discrepancies that result at the circuit production stage [17], [26].

Mathematically speaking, PUFs are one-way functions such that for an input challenge C_i, $PUF(C_i) \rightarrow R_i$, where R_i is the unique response for challenge C_i. From a high level, R_i is considered to be the volatile secret key. Since PUFs are determined by variations in the hardware, the response R_i is only derived upon circuit execution and cannot be derived based on assumptions of the function itself. PUFs are neither linear nor injective nor surjective. Figure 11 presents an example of a circuit-based PUF built upon a 1-bit ring oscillator. Ring oscillators are designed to be identical. However, as a result of the manufacturing process, one oscillator will be oscillating faster than the other thereby outputting slightly different results.

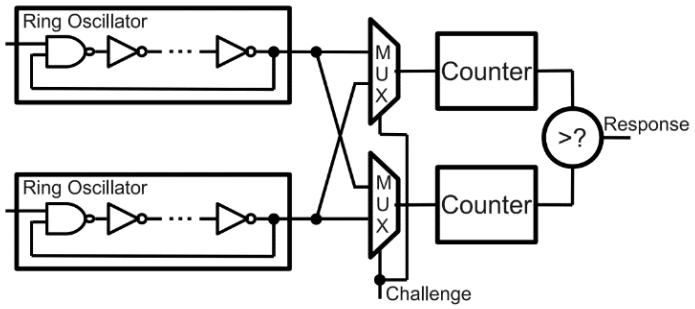

Figure 11 - High level overview of a 1-bit ring oscillator PUF

In our work, the PUF circuitry will be used for 2 purposes: The 1st is to provide secure cryptographic storage to private and secret keys stored on publicly accessible memory i.e. to protect the secret key K, one could store $X = K \oplus R_i$ on the accessible memory using the encrypted version of key K instead of storing K in plaintext. Later, when K needs to be used, R_i and X will be used to retrieve back K. The 2nd purpose is to provide volatile keys generated from the PUF and which are not stored on any addressable memory. The generated keys will only depend on C_i. The purpose of each will be discussed in the upcoming sections.

One method to realize the concepts discussed is to have a system-on-chip (SoC) design as shown in Figure 12. **Addressable Memory** is used to fetch the encrypted stored in

accessible memory. **Crypto_in** contains the data that needs to be passed to the Cryptographic Core. The Cryptographic Core (CC) is a hardware component that provides symmetric key cryptographic services such as AES, CMAC and HMAC etc. Learning the secret key from the bus is nearly impossible since keys are never transmitted across the bus. To utilize the 1st feature discussed, the chain $PUF(C_i) \rightarrow ECU(R_i) \oplus Data \rightarrow CC_K(Crypto\_in)$ is utilized. To utilize the 2nd feature the chain $PUF(C_i) \rightarrow ECU(R_i) \rightarrow CC_{Key}(Crypto\_in)$ is utilized.

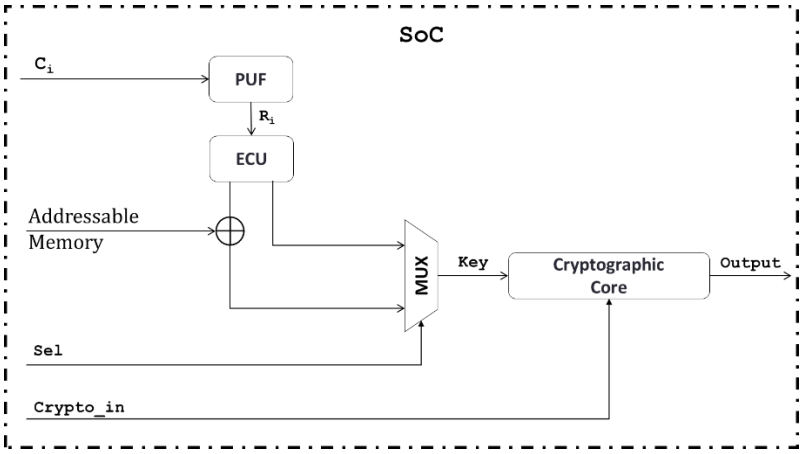

Figure 12 - PUF based SoC

In order to maintain the same output every time, the same challenge is given, an error-correcting unit (ECU), such as Reed-Solomon, is placed after the output of the PUF to perform error correction on the noisy response.

Another method to realize the concepts discussed above is to have a single-chip secure processor as in [27] where all components outside the processor chip, including external memory and peripherals, are assumed to be insecure, that is, they may be observed and tampered by an adversary.

3.3 Proposed Methodology

In this section the attacker model and the proposed security framework are presented along with solutions to the problems discussed in the introduction.

3.3.1 Attacker model

The attacker is assumed to be present in the network after the installation of the smart meters in the field. She is assumed to have a wireless device that has the same communication protocols used by regular smart meters. As stated in the introduction, the attacker can:

- Sniff packets passing through the wireless medium
- Inject new packets into the network to send information or commands to the meters or to the utility
- Inject new bogus packets into the network for the purpose of flooding the network and creating a DoS attack.
- Tamper with the hardware of the smart meter in order to retrieve secret keys used by the cryptographic functions.

3.3.2 Methodology overview

To guard against such an attacker model, we propose to rely on an end-to-end and hop-by-hop security mechanism as shown in Figure 13. The security roles for each entity can be described as follows.

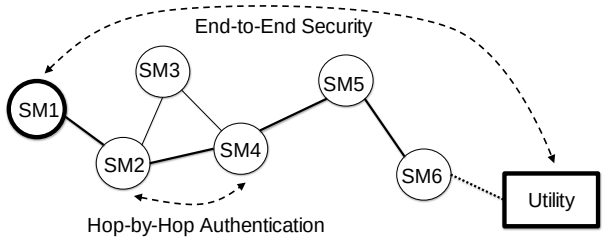

Figure 13 - End-to-end and hop-by-hop security mechanism overview

1. **Utility**: It represents the communication end-point for all the smart meters and the Private Key Generator(PKG) of the ID-based cryptographic system. It is the trusted entity responsible for:
   a. Storing challenge/response pairs of the smart meters' devices.
   b. Generating and transporting encrypted private keys for the smart meters during the setup/update phase. The encryption is performed by relying on the volatile keys generated from the PUF devices

2. **Smart meter**:
   a. Sending/receiving messages to/from the utility with optional end-to-end security that utilizes the responses generated from PUFs as keys for encryption and authentication at the application layer.
   b. Acting as a relay to forward messages of its direct neighbours. Implementing link layer authentication that requires the generation of pairwise symmetric keys between the meter and each of its neighbours.

All generated and stored keys will be encrypted using PUF responses of the meter to guard against key leakage at the hardware level. Furthermore, the unique keys used on the application layer for the meter-to-utility along with the pair-wise symmetric key generation used for the link layer security helps to circumvent total network compromise that might arise when a single shared key is used for the whole system. In what follows, we describe the details of the implementation.

3.4 Pre-deployment phase

The pre-deployment phase is when the meters are still available at the utility's vicinity prior to deploying them in the field. At this stage, it is assumed that the utility has physical access to the meters. The first step of this phase is inputting a challenge C_i (i is device index) to the PUF of the smart meter and obtaining a response R_i. The utility then stores the value of the response R_i in its database. This operation is repeated for all meters that will be deployed in the fields.

The second step involves setting up the ID-based cryptosystem and generating the private keys for the meters, referred to as the **Setup** and the **Keygen** algorithms respectively [22], [28].

Let $k \in \mathbb{Z}^+$ be the security parameter given to the setup algorithm. Let $\mathcal{G}$ be a BDH parameter generator having a polynomial runtime in k and satisfying the BDH assumption.

Definition. The BDH parameter generator $\mathcal{G}$ can be formulated as follows:

$$\mathcal{G}(1^k) = \{q, \langle\hat{e}\rangle, \langle\mathbb{G}\rangle, \langle\mathbb{G}_T\rangle\}$$

$\mathcal{G}$ takes k and outputs a random k-bit prime number q, description of the groups $\mathbb{G}$ and $\mathbb{G}_T$ and the description of a bilinear map $\hat{e}: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. The descriptions $\langle\mathbb{G}\rangle$ and $\langle\mathbb{G}_T\rangle$ contain algorithms for computing the group action in $\mathbb{G}/\mathbb{G}_T$ and contain a generator of $\mathbb{G}/\mathbb{G}_T$. On the other hand, the description $\langle\hat{e}\rangle$ contains a polynomial time algorithm for computing $\hat{e}$.

Definition. The BDH assumption formulates that an algorithm $\mathcal{A}$ has an advantage $\epsilon(k)$ in solving the BDH problem for $\mathcal{G}$ if for a large value of k

$$\mathrm{Adv}_{\mathcal{G},A}(k) = \Pr\left[\mathcal{A}(q, \mathbb{G}, \mathbb{G}_T, \hat{e}, P, aP, bP, cP) = \hat{e}(P,P)^{abc} \,\middle|\, \begin{matrix} \mathbb{Z}_q^* \rightarrow a, b, c;\ \mathbb{G} \rightarrow P \\ \mathcal{G}(1^k) \rightarrow \{q, \langle\hat{e}\rangle, \langle\mathbb{G}\rangle, \langle\mathbb{G}_T\rangle\} \end{matrix}\right] \geq \epsilon(k)$$

That is $\mathcal{G}$ satisfies the BDH assumption if for a randomized polynomial time algorithm $\mathcal{A}$, $\mathrm{Adv}_{\mathcal{G},A}(k)$ is a negligible function. When this is satisfied, $\mathcal{G}$ generates BDH hard groups.

Setup	
(1)	Execute $\mathcal{G}(1^k)$ to obtain $\{q, \langle\hat{e}\rangle, \langle\mathbb{G}\rangle, \langle\mathbb{G}_T\rangle\}$.
(2)	Let $P \in \mathbb{G}_1$ be a random generator. Randomly select $s \in \mathbb{Z}_q^*$ the PKG's secret key; set $P_{pub} = sP$ as the PKG's public key.
(3)	Choose a cryptographic hash function $H: \{0,1\}^* \rightarrow \mathbb{G}$
(4)	Set $\{P_{pub}, P, H, q, \langle\hat{e}\rangle, \langle\mathbb{G}\rangle, \langle\mathbb{G}_T\rangle\}$ as the public parameters of the cryptosystems.

$\langle\hat{e}\rangle$ generated from the execution of $\mathcal{G}(1^k)$ exhibits 3 useful bilinear properties:

1. **Bilinearity**: $\forall P, Q \in \mathbb{G}_1, \forall a, b \in \mathbb{Z}_q^*$
   $\hat{\boldsymbol{e}}(aP, bQ) = \hat{\boldsymbol{e}}(P, Q)^{ab}$
2. **Non-Degeneracy**: $\forall P \in \mathbb{G}_1\ s.t\ P \neq 0 \Rightarrow \hat{\boldsymbol{e}}(P,P) = \mathbb{G}_2$, that is for $P \neq 0 \Rightarrow \hat{\boldsymbol{e}}(P,P) \neq 1$
3. **Computability**: $\hat{\boldsymbol{e}}$ is efficiently computable

Keygen	
(1)	For every meter x having an identity denoted by ID_x, calculate $P_x = H\,(ID_x)$
(2)	Generate smart meter x's private key $Pvt_x = s.P_x$ where is the PKG's secret key

ID_x is an identity that defines the smart meter. ID_x can be any publicly known information about the meter. In our case we assume it is the meter's wireless network interface's physical address.

After executing the **Setup** and **Keygen** algorithms, the utility stores $e_{R_i}(Pvt_x)$ that is the encrypted version of Pvt_x in the meter's memory using the response obtained from the PUF as the symmetric key for the encryption. Other than storing Pvt_x, the utility stores the challenge C_i and the public parameters $\{P_{pub}, P, H, q, \langle\hat{e}\rangle, \langle\mathbb{G}\rangle, \langle\mathbb{G}_T\rangle\}$ of the cryptosystems generated by $\mathcal{G}$ during the **Setup** phase. At a later stage, when the meter wants to retrieve Pvt_x, it executes $PUF(C_i) \rightarrow R_i$ and uses its outcome to decrypt the encrypted private key stored within its memory in the following manner $d_{R_i}\left(e_{R_i}(Pvt_x)\right) = Pvt_x$.

3.5 ID-NIKD and pairwise symmetric key generation

Consider two neighbouring smart meters A and B with unique IDs, ID_A and ID_B respectively. Before communicating with the utility, meter A should know ID_B and meter B should know ID_A so that they can derive the pairwise symmetric key used for the link layer security mechanism. Our scheme assumes that a neighbour discovery protocol will provide the IDs to the ID-NIKD mechanism. The two meters also have their private keys Pvt_A and Pvt_B generated in the **Keygen** phase and securely stored in their memories during

the Pre-deployment phase. By only knowing the ID of the other meter, both meters are able to derive the pairwise symmetric key without exchanging any additional information. This is done using ID-based non-interactive key distribution scheme.

Meter A derives the symmetric key $SK_{A,B}$ using Pvt_A and ID_B as follows:

$$\hat{e}\left(pvt_A, H(ID_B)\right) = \hat{e}(s.H(ID_A), H(ID_B)) = \hat{e}(H(ID_A), H(ID_B))^s = SK_{A,B}$$

Meter B derives the symmetric key $SK_{B,A}$ using Pvt_B and ID_A as follows:

$$\hat{e}\left(pvt_B, H(ID_A)\right) = \hat{e}(s.H(ID_B), H(ID_A)) = \hat{e}(H(ID_B), H(ID_A))^s = SK_{B,A}$$

It is clear from the above equations that $SK_{A,B} = SK_{B,A}$. The resultant symmetric key is then post-processed with a key derivation function to make it compatible with the used cryptographic functions [28]. Although pairings are computationally expensive, this step is only performed when a new neighbour is discovered or a new key needs to be negotiated due to key updates. Once the pairwise symmetric key is derived, it is stored securely in the device memory for further usage. Hence, it is encrypted by the PUF response and is not stored in plain text.

3.6 Field phase and packet overview

Figure 14 illustrates the general packet format for the proposed methodology. We assume at this point that at the application layer, meter-to-utility communication is based on the ANSI C12.22 protocol [29]. ANSI C12.22 messages consist of various nested elements: Association Control Service Element (ACSE), user-information element and one or more Extended Protocol Specifications for Electric Metering (EPSEM) that carry data and ANSI C12.19 tables. EPSEM data can be sent authenticated, encrypted or both authenticated and encrypted. If authentication is enabled, a MAC is included for validity checking and authentication of the message. If encryption is enabled, a portion of user-information element is encrypted. The Authenticated Encryption with Associated Data (AEAD) mode adopted by ANSI C12.22 is the EAX' mode which is based on AES-128 in CBC and CTR modes and uses standard Message Authentication Mode, CMAC. In our

scheme, the response of the PUF is used as the key for the EAX' to provide authentication and encryption.

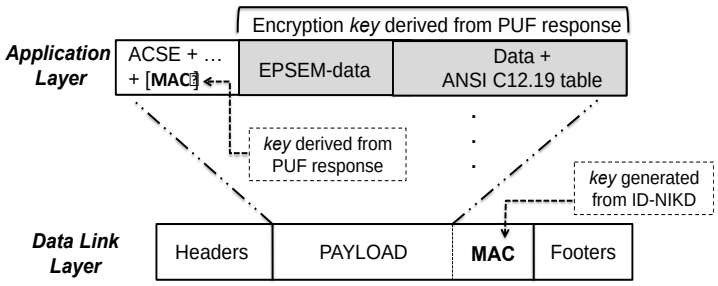

Figure 14 - Packet overview illustrating security mechansims

At the data-link layer, used in 802.15.4g [30] or the alternative wireless mesh network protocol 802.11s [31], the pairwise keys generated are supplied to the security mechanism in order to generate a MAC as illustrated in Figure 15. Our framework relies on the freshness checks of 802.15.4g, which are based on frame sequence numbers to avoid replay attacks.

As packets are transported through the network, the proposed link layer authentication mechanism mitigates the effects of DoS attacks that are caused by flooding the network with bogus packets. This is achieved by discarding bogus packets at the point of detection and preventing them from further traversing the mesh network. Such flooding attacks would otherwise consume the energy and computational resources of all on-transit meters that carry the packets to the utility. In addition, they consume network bandwidth and congest the queues of the on-transit meters. This can lead to increased packet latency and potential packet loss thereby leading to dire consequences. It is worth mentioning that our scheme is not specific only to 802.15.4g. It can be applied to any link layer protocol. As for the Message Authentication Code (MAC), any MAC generating algorithm can be used such as HMAC, CMAC etc.

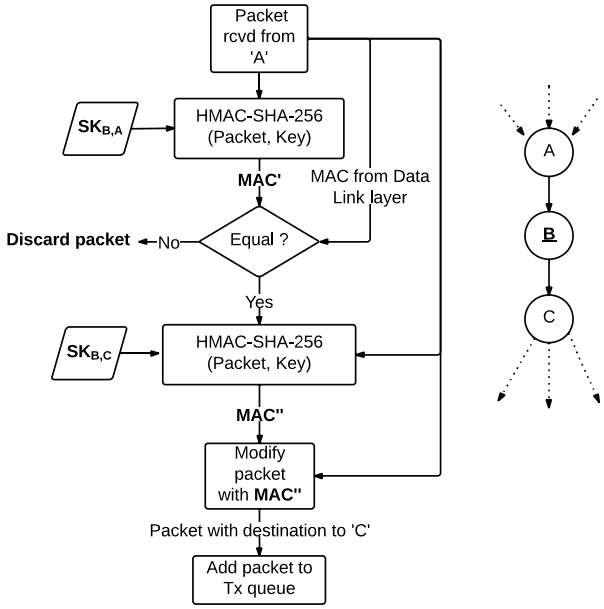

Figure 15 – Hop-by-hop authentication mechanism in action

The authentication scheme utilizes the pairwise symmetric keys generated by the ID-NIKD scheme. In Figure 15, Meter 'B' receiving a packet from 'A' will forward it to 'C' only if the computed message authentication code, MAC' = HMAC-SHA-256 (Packet, $\mathbf{SK_{B,A}}$), matches the one received in the Aux security header of the data link layer. If the two MACs match, meter 'B' replaces the existing MAC with MAC'' = HMAC-SHA-256 (Packet, $\mathbf{SK_{B,C}}$) and forwards the updated packet to meter 'C'. HMAC-SHA-256 is merely an example, and as stated can be replaced by any other MAC generating algorithm.

3.7 Simulations & Results

We simulated the proposed methodology for a wireless mesh based AMI at the NAN level in the event of a DoS attack. Our goal was to collect packet loss and latency data from the perspective of the aggregator. We relied on a discrete event-driven network simulator NS-3 to simulate the network for the following scenarios. The simulations do not take into consideration the existence of the PUF circuitry.

1. *Scenario I (NoAuth_NoAttack):* No authentication and no attack. This is the golden reference scenario.
2. *Scenario II (Auth_NoAttack):* Authentication and no attack. This scenario evaluates the overhead of the proposed methodology in the event of no attacks.
3. *Scenario III (NoAuth_Attack):* No authentication and attack. This scenario evaluates the impact of the attacker on an unsecured network.
4. *Scenario IV (Auth_Attack):* Authentication and attack. This scenario measures the benefits of the proposed methodology in the event of an attack.

3.7.1 Simulation Setup

The simulated system is assumed to carry a few hundred smart meters. Each *Node* in the system is connected to tens of smart meters. It is assumed that every smart meter will report its metering data at least every 15 minutes. Other than reporting metering data, meters respond to utility queries and send sync and routing packets to keep links fresh. The simulated topology for an M hop network is presented in Figure 16. For purposes of our simulation, we assumed that each node, N, sends 1 packet every 9 seconds. The attacker is positioned in the middle of the network and is flooding the network at almost full bandwidth capacity.

We simulate the access to the wireless medium using Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) protocol where the wireless channel is taken to be error free in order to obtain clear results without external factors. The link speeds and maximum transmission unit (MTU) used are similar to that of 802.15.4g. The MTU is set to 1500 bytes.

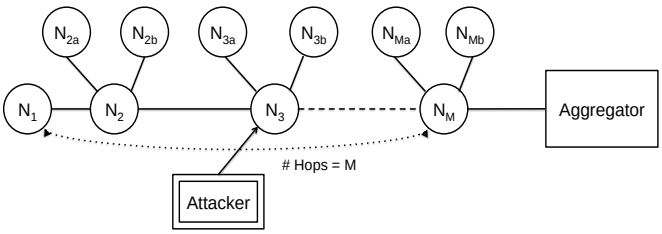

Figure 16 - Topology of the simulated network

Meters communicate using IPv4 at the network layer and use UDP at the transport layer. When smart meters are communicating with the utility head-end, the application layer carries messages of sizes typical to ANSI C12.22 messages. If link layer authentication

is enabled, additional bytes are appended to the payload at the link layer that serve as the MAC. The key used to generate the MAC is based on the non-interactive key distribution mechanism discussed in the previous sections where it is assumed that neighbouring meters have already generated pairwise symmetric keys. Routing tables are generated before the runtime of the simulation and are static during the simulation. The Tx buffers of the meters are assumed to be 150,000 bytes; incoming packets are dropped when the buffers are full and cannot accept any more packets.

Processors inside meters are assumed to be resource constrained with processor frequencies in the 100MHz range. SHA-256 is used to produce the MAC at the link layer and its implementation is based on Crypto++ library that takes 15.8 cycles per byte. The generated MAC has a size of 32 bytes. Genuine smart meter packets require 2 SHA-256 operations: one for checking its authentication and the other to generate a new MAC. However, a bogus packet requires only 1 SHA-256 operation to check its authenticity. The delays inferred by the hashing functions are integrated within the simulation to produce accurate results.

In all scenarios that include an attacker, the attacker will transmit fixed sized packets of 300 bytes at a link rate similar to that of the smart meters.

3.7.2 Simulation Results

For our simulations, we varied different network parameters in terms of packet size, link speed and number of hops. For a typical NAN, the number of hops can vary between 5 and 15 hops [32]. The packet size ranges between 200 and 500 bytes that include the lower layer communication headers and the ANSI C12.22 message sizes. We also varied the link speed between 100 and 400 Kbps. We, thus, simulated the different scenarios using the experimental setups illustrated in Table 4.

Table 4 - Experimental setups with different network parameters.

	Packet Size (Bytes)	**Link Speed (Kbps)**	**# Hops**
Exp #1	[200-500]	200	10
Exp #2	300	[100-400]	10
Exp #3	300	200	[6-14]

Table 5 summarizes the average packet latency for a simulation runtime of 5 minutes for Exp #1.

Table 5 - Average packet latency for variable packet size corresponding to Figure 18

	Exp #1: Average Packet Latency (ms)			
Packet Size (kB)	**Scenario I**	**Scenario II**	**Scenario III**	**Scenario IV**
200	42	44	1599	201
300	61	65	2260	219
400	80	83	2917	237
500	98	102	3578	256

Figures Figure 17, Figure 18 Figure 19 illustrate all the average packet latency simulation results for the different experiments with simulation runtime of 5 minutes. Two key observations are listed below.

1. In the absence of an attacker, the overhead of the proposed methodology is minimal with a maximum of 6.5% increase in the average packet latency recorded for scenario II (Auth_noAttack) compared to scenario I. This is attributed to the fact that the hash function evaluations induce minimal cost.
2. In the presence of an attacker, the authentication scheme is able to thwart the attack and reduce the average packet latency significantly compared to scenario III (noAuth_Attack). Note that scenario III latencies are 8x-14x larger than those of scenario IV.

To study the packet loss ratio, we then simulated the network for a longer time of 30 minutes. As illustrated in Table 6, the corresponding packet loss ratio for scenario III reaches up to 23%. On other hand, in scenario IV the packet loss is eliminated completely due to the fact that bogus packets are discarded before being added to the Tx buffer.

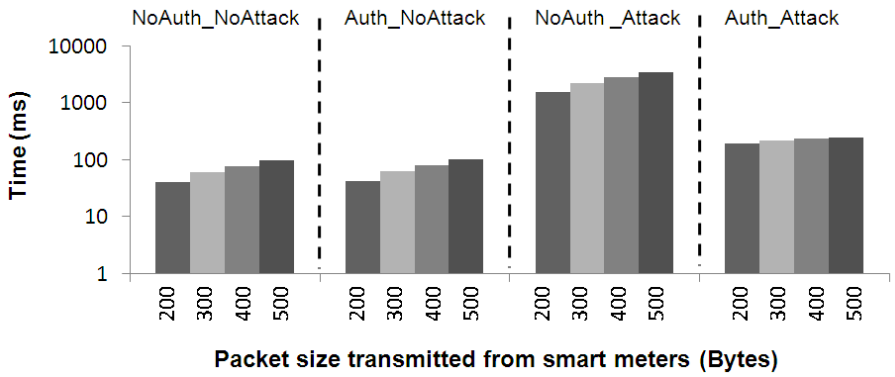

Figure 17 - Average packet latency for Exp #1

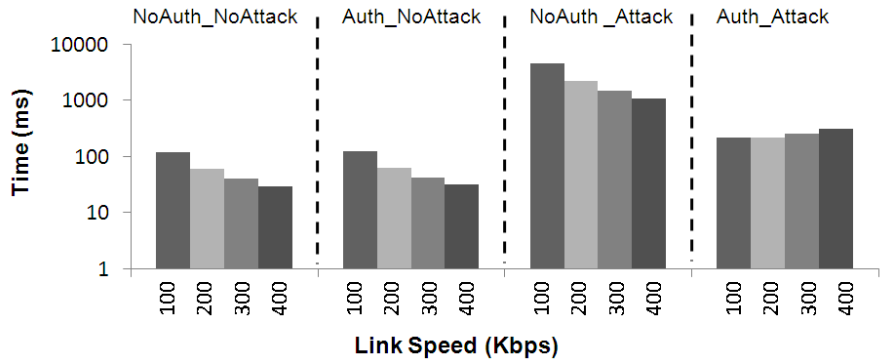

Figure 18 - Average packet latency for Exp #2

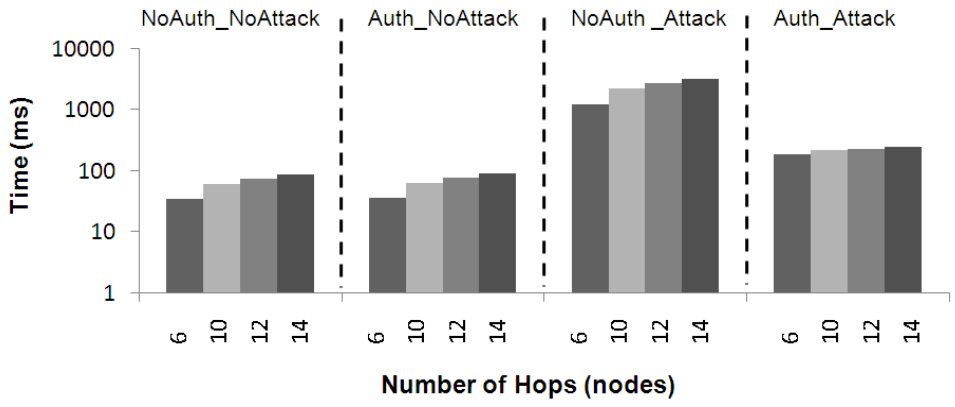

Figure 19 - Average packet latency for Exp #3

Table 6 - Smart meter packet drops in Scenario III

Exp #1	**Link Speed (Kbps)**	**100**	**200**	**300**	**400**
	% Dropped Packets	6.52	6.72	5.48	6.02
Exp #2	**Packet Size (Bytes)**	**200**	**300**	**400**	**500**
	% Dropped Packets	0.36	6.72	16.84	23.22
Exp #3	**# Hops**	**6**	**10**	**12**	**14**
	% Dropped Packets	6.53	6.72	8.13	10.70

Chapter 4 - Lightweight Key Update & Delivery Mechanism for ID-based Cryptosystem

4.1 Introduction

In this chapter, we continue our work on the proposed framework of chapter 3. Key update and delivery mechanisms play a crucial role in the key distribution framework. Once smart meters are deployed and running, they need to obtain new keys for different security schemes. In this chapter we propose a lightweight authenticated key-update and delivery mechanism. The keys provided by the lightweight update and delivery will be utilized by the identity-based non-interactive key distribution framework. The delivery process exploits the multicasting/broadcasting capabilities of the network whereby the authenticity of the updated keys is ensured by utilizing a proposed efficient algorithm.

4.2 Private key update process

The key update process can be seen as a variant of the **Keygen** process described in the previous chapter. By initiating a key update, the utility generates a new set of private keys for the nodes within a selected group of smart meters that it wishes to provide new keys for. In addition to updating keys, the proposed private key update process is intended to achieve identity revocation. Revocation is achieved by merely excluding identities from the group that the utility wishes to grant new keys. The proposed time-correlated key update process is motivated by the work in [22]. The key update procedure described relies on the Real Time Clocks (RTC) embedded within meters' microcontrollers providing precise timing information. This assumption is realistic since all vendors that produce microcontrollers used in smart meter applications embed RTCs. The proposed key update mechanism differs from the **Keygen** process in the definition of P_x, such that:

$$P_x' = H\,(ID_x \,||\, \varphi)$$

The function $H()$ and ID_x were both defined in the previous chapter. The symbol $||$ denotes the concatenation operation. φ is an argument given to the update process that

denotes some timing information. This argument is dependent on how frequent the utility wants to update the keys which has an implication on security and on the effects of a key compromise. A higher frequency of key update indicates better security and reduces the effects of a compromised key. For this scheme to work, smart meters will have to use P_x' as the public key of the neighbouring meters instead of using P_x. φ is generated inside smart meters by relying on 2 values; the first is the time information obtained from the RTC and the second is based on the frequency of the key update dictated by the utility during the **Setup** process where the utility sets $\{\varphi, P_{pub}, P, H, q, \langle\hat{e}\rangle, \langle\mathbb{G}\rangle, \langle\mathbb{G}_T\rangle\}$ as the public parameter of the cryptosystem. By following such a scheme, smart meters that haven't obtained new keys for a certain date span will not be able to communicate with the neighbouring meters, thus automatically having their identities revoked.

4.3 Key Delivery Mechanism and Shuffled-ID based Security Scheme

For our application, the smart meters need to obtain new private keys for the ID-NIKD scheme. The frequency of the key updates has an implication on security and on the effects of a key compromise. A higher frequency of key update indicates better security and reduces the effects of a compromised key. For key delivery mechanisms, there are 2 major security aspects that should be employed: encryption and authentication of the delivered private keys.

- Encryption ensures that the delivered keys are properly revealed only to the destined end parties. That is, no eavesdropper on the delivery path can obtain any data regarding the transmitted private key on route.
- Authentication allows the destined end parties to verify that the received key is truly generated by the utility and is not a bogus key. In the absence of authentication, an attacker might be able to generate a forged encrypted key (without necessarily knowing its content), forcing a smart meter to accept the key as a genuine key, thereby blocking its access to the network.

Typically, authenticated encryption is used to realize the mentioned points which is can be implemented by having an encrypted block of data prepended next to a message

authentication code (MAC). To achieve high levels of security and to ensure very low collision probabilities, 16 bytes of MAC is recommended. In the following subsections, we propose an efficient authenticated encryption methodology with lower packet overhead compared to regular authenticated encryption techniques. For purposes of our proposed key delivery mechanism, we focus on the following aspects:

1. Security and robustness of the key delivery mechanism.
2. Latency and network traffic reduction.

4.3.1 Precursor

The Advanced Encryption Standard (AES) will be used for encrypting the private keys and it will be used as a pseudo random permutation block (PRP) for the authentication of the private keys. Since the ID-NIKD is a mechanism based on Elliptic Curve Crypto (ECC), it will be using one of the ECC key sizes recommended by NIST [33]. An equivalence table between the different cryptographic domains can be observed in Table 7. For purposes of our analysis, we assume an IPv4 network along with UDP at the transport layer having a Maximum Transfer Unit (MTU) of 1500 bytes.

Table 7 - NIST recommended key sizes

Symmetric (bits)	RSA (bits)	ECC (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

4.3.2 Key delivery and Authentication Mechanism

In order to reduce the latency and the overhead placed on the network during the delivery of the newly generated keys, the proposed mechanism exploits the multicasting features Figure 20 of the network and sends the keys in an aggregated fashion without degrading the security of the delivered keys.

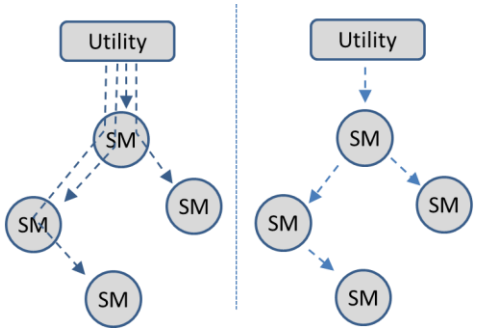

Figure 20 - Unicast key delivery (left) vs Mutlicast (right).

The mechanism is designed in a manner that a smart meter is able to drop its own private key from the aggregated list of keys inside the multicast packet and to forward the rest of the data without affecting the authenticity of the packet. This helps in the reduction of the size of the packet as it traverses the mesh network. The proposed delivery form reduces the traffic generated in the wireless mesh network compared to a regular unicast key delivery form. The headers used in the different OSI layers (link layer, network and transport) will be shared, thus reducing the overhead ratio. In order to be able to gain the capability of removing chunks of bytes from the packet as it traverses through the network, authentication has to be performed on each individual key. A simple approach would be to place an encrypted private key and a message authentication code for each key. However, a secure message authentication code would have sizes not less than 16 bytes. This increases the size of the packet and decreases the space left for the private keys to be stored since we have a fixed MTU.

Figure 21 depicts how the proposed key delivery packet looks like at the application layer. The **`Nonce`** is a random number generated by the utility to add entropy into the system and is used to protect against replay attacks. This **`Nonce`** is employed in the encryption and authentication blocks. ID is a number that presents the identity of a smart meter. **`ID_KEY`** is an encrypted version of both the private key and the shuffled bits of the **`ID`** as illustrated in Figure 21. The Application layer key delivery packet consists of the **`Nonce`** followed by a sequence of **`ID`** and **`ID_KEY`**. **`ID`** bits are shuffled with the private key bits and encrypted to form **`ID_KEY`**. It is utilized also for the authentication of the private keys as will be explained in the following subsection.

Figure 22 shows the average number of keys that can be stored in the application layer (y-axis) using the configuration to be specified in the simulation's section, versus the size of the keys used shown in Table 7. In Figure 22 a nonce of 4 bytes is assumed to be used along with an ID of size 4 bytes as it can cover ˜4 billion smart meters. The average number of keys stored at the application layer was calculated as follows.

$$\frac{MTU - \sum(OSI\_Layers + Nonce\_size)}{Pvt\_Key\_size + 2 * ID\_size}$$

The following subsection discusses the security implications of the ID size on the authentication mechanism.

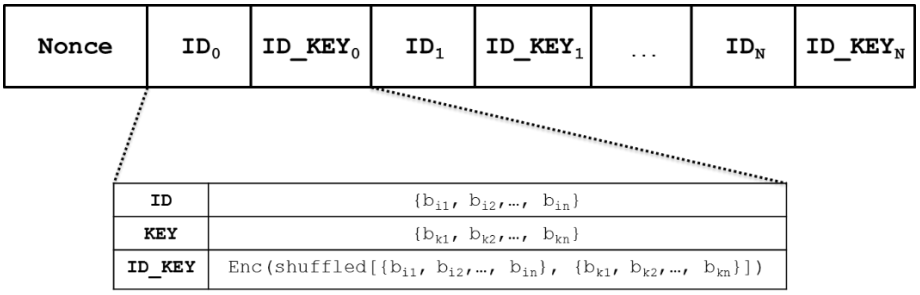

Figure 21 - Proposed key delivery packet at the application layer

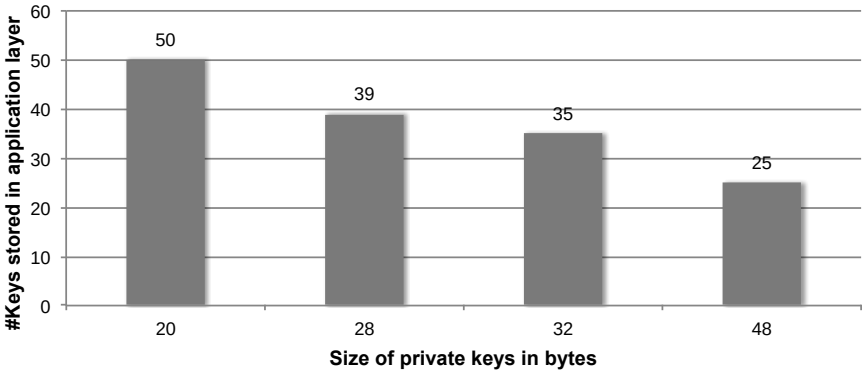

Figure 22 - Number of stored keys vs. key size

4.4 Blocks Ciphers, Pseudo Random Permutation & AES

A block cipher is a deterministic cipher $\varepsilon = (E, D)$ having a key space K where both message and ciphertext space are in the same finite set X [20], [34]. For every $k \in \mathrm{K}$ the function $f_k \coloneqq E(k,.)$ is defined s.t f_k: X $\rightarrow$ X sends every element $x \in \mathrm{X}$ to $E(k,x) \in \mathrm{X}$. Formally speaking, any cipher implies that for every fixed key $k \in \mathrm{K}$:

- $f(k,.)$ is a one-to-one function[1]
- $\exists$ an efficient deterministic algorithm to evaluate $f(k,.)$ which calculates a permutation on X
- $\exists$ an efficient inversion algorithm $f^{-1}(k,x) = D(k,.)$
- A secure block cipher should be computationally indistinguishable from a truly random permutation

To define the security of a pseudo random permutation (PRP), let the set $P_F = \{\, E(k,.) \mid k \in \mathrm{K} \,\} \subseteq \mathrm{Perms}\,[\mathrm{X}]$ denote all the permutations specified by the PRP when the key is specified. A PRP is said to be secure if a truly random permutation in Perms [X] is indistinguishable from a random permutation in P_F; Perms [X] denotes all permutations on X that is $|\mathrm{Perms}\,[\mathrm{X}]| = |\mathrm{X}!\,|$.

We define an experiment denoted by $\mathrm{Exp}(m)$ as a game between a challenger and an adversary $\mathcal{A}$ where $m = \{0,1\}$.

- For $m = 0$, the challenger takes $k \leftarrow \mathrm{K}$ and $p \leftarrow \mathrm{E(k,.)}$
- For $m = 1$, the challenger takes $p \leftarrow \mathrm{Perms}\,[\mathrm{X}]$

The adversary submits queries $x_1, x_2 \ldots, x_i \in \mathrm{X}$ to the challenger. The challenger responds back with $p(x_1), p(x_2) \ldots, p(x_i) \in \mathrm{X}$. After reception, the adversary decides if the output is from a truly random or a pseudorandom permutation.

In order to assume that a certain PRP is secure all efficient[1] $\mathcal{A}$ should have a negligible probability to win the game. That is

$$\mathrm{Adv_{PRP}}[\mathcal{A}, \mathrm{P}] = |\,\Pr[\mathrm{Exp}[0] = 0] - \Pr[\mathrm{Exp}[1] = 0]| < \epsilon$$

[1] By efficient we mean that $\mathcal{A}$ runs in polynomial time or in a specific time period of time.

Which means that $\mathcal{A}$ should not be able to distinguish Exp[0] from Exp[1].

For AES with $|X| = 2^{128}$ the number of permutations is $\approx 2^{2^{135}}$. As for the number of permutations defined by 128-bit AES keys is at most 2^{128}.

Since all 2^{80}-time algorithms have $\mathrm{Adv}_{\mathrm{PRP}}[\mathcal{A}, \mathrm{AES}] < 2^{-40}$, therefore AES is a secure PRP making it a secure block cipher.

4.5 Security of Key distribution mechanism

The proposed algorithm adds 4 bytes and achieves an equivalent level of security as compared to adding 16 bytes of MAC next to each private key. The algorithm attains this by combining shuffled bits of **ID** at certain deterministic positions within the private key **KEY** and generating an encrypted version of the combination denoted by **ID_KEY.** The process relies on shared keys associated between a meter and the utility which can be generated by utilizing the PUF as described in the previous sections or by using ID-NIKD as a block to generate a shared key between the utility and the meter.

Table 8 presents the authenticated encryption algorithm ran at the utility. In step 1, the utility starts by generating a **Nonce**. In step 2, it generates a unique private key to be delivered to each meter in the multicast sub-network as described in the key update section. For step 3, the utility allocates **T** bits of memory for each key in **ID_KEY_BUFFER** where **T** represents the number of bits allocated for one **ID** and one private key. In steps 4 and 5 the utility uses **ID** and the **Nonce** along with a shared secret key between the utility and the meter of identity $\mathbf{ID_i}$ to generate S_1 as a seed for the Pseudo Random Generator(PRG). The utility then proceeds according to steps 6 and 7 and generates the index ID_bit_pos for each bit of $\mathbf{ID_i}$ and places those bits into the $\mathrm{ID\_bit\_pos}^{\mathrm{th}}$ position of **ID_KEY_BUFFER**, respectively. After filling the $\mathbf{ID_i}$ bits in their designated positions, the utility fills the rest of **ID_KEY_BUFFER** with the unencrypted version of the private key $\mathbf{KEY_i}$. The final step is to encrypt **ID_KEY_BUFFER** with the shared key and generate $\mathbf{ID\_KEY_i}$. After running the algorithm of Table 8, **ID_KEY** will contain the authenticated and encrypted version of **KEY**. It

is assumed that AES is ran in a mode that doesn't require padding (such as AES-OFB or AES-CTR) in order to keep the output size minimal.

*Table 8 - **ID_KEY** Generation algorithm at the utility*

1	Generate **Nonce**
	ForEach $\mathbf{ID_i}$:
2	Generate $\mathbf{KEY_i}$
3	**ID_KEY_BUFFER** = allocate_**T**_bits()
4	S_1 = $HMAC_{shared\_KEY}$ (**ID**$_i$, **Nonce**)
5	PRG_seed = S_1
6	for *count*:= 1 to **T**:
7	ID_bit_pos = PRG() mod **T**
8	ID_KEY_BUFFER [ID_bit_pos]= **ID**$_i$ [*count*]
9	Fill rest of ID_KEY_BUFFER with **KEY**$_i$
10	**ID_KEY**$_i$ = $Encrypt_{shared\_KEY}$(ID_KEY_BUFFER, **Nonce**)

By exploiting the fact that AES is a secure block cipher which was discussed in section 4.4, the output from the AES block as depicted in Figure 23 will produce a pseudo random output which cannot be distinguished from a totally random number by an adversary.

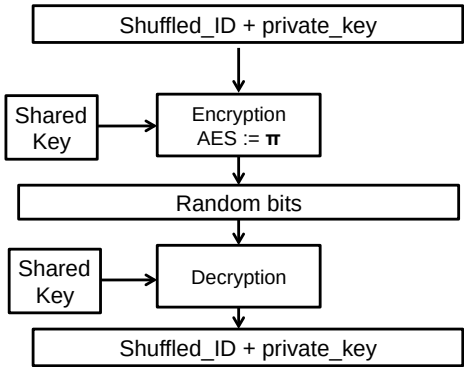

Figure 23 – Utilization of AES within proposed system

When a smart meter carrying $\mathbf{ID_i}$ receives an update packet, it follows the complementary steps of generating $\mathbf{ID\_KEY_i}$. The smart meter accepts the key only if the decrypted version of $\mathbf{ID\_KEY_i}$ contains the correct values of $\mathbf{ID_i}$ bits in their correct positions. Note that the

meter will be able to generate the same `ID_bit_pos` values using the same seed S_1 only when a deterministic PRG is used.

To assess the strength of the authentication mechanism, we calculate the probability of an attacker forging an authentic packet. An attacker might try to randomly generate a bit string with the goal of forcing the meter to accept a new bogus key. Although the attacker will not know the value of the key, however she can prevent the smart meter from further communicating with other neighbors. An attack is considered to be successful if the attacker forges a bit string that when decrypted, contains ID bits with values and positions that match the calculated ones within the meter. Hence, the probability of a successful attack is:

$$\Pr(Correct\ ID\ values) \times \Pr(Correct\ ID\_bit\ locations) = \frac{1}{2^N} \times \frac{(T-N)!}{T!} \times N!$$

Where T and N are **`ID_KEY`** and **`ID`** sizes in bits. Figure 24 plots the probability of collision for different sizes of ID for **`KEY`** of size 160 bits.

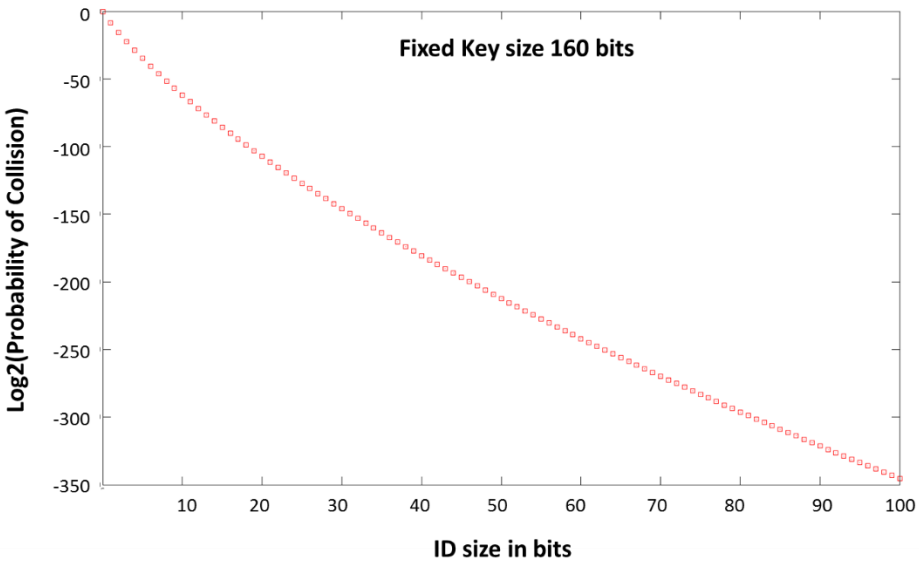

Figure 24 - Probability of collision for different ID sizes

It can be seen from the graph that for a key size of 20 bytes and ID size of 4 bytes, the probability of collision is $\approx 2^{-153}$.

4.6 Simulation Analysis and Results

In this section, we demonstrate the low latency overhead of the key authentication algorithm on a fast wireless mesh 802.11s network. We also evaluate the effectiveness of multicast key delivery mechanism. First, we perform hardware testing to evaluate the computational overhead placed on the microcontroller of a smart meter.

4.6.1 Hardware Testing

The computational overhead of the key authentication algorithm is assessed on the STM32F217IGH microcontroller, which is based on the ARM Cortex M3 architecture and is chosen in order to mimic smart meters; many of the smart meter vendors use microcontrollers based on the Cortex M3 architecture. Additionally, many of the STM32F2x family of microcontrollers are equipped with cryptographic processors, hash processors and random number generators. The cryptographic processor implements AES, DES and 3-DES in ECB, CBC and CTR modes. The hash processor implements SHA-1, MD5 and HMAC-SHA-1-MD5. The random number generator (RNG) is a true random generator that generates random bits based on continuous noise signal.

We developed the code using the Keil MDK5 environment, compiled with time optimization and relied on the wolfSSL/CyaSSL [35] embedded library as the cryptographic library. To reduce the runtime, we relied on the built-in cryptographic and hash libraries to implement the cryptographic code blocks. However, we did not use the true RNG hardware block because a deterministic RNG is needed to produce identical results on both ends (the utility and user). The input and key sizes used in the simulations are based on real scenarios. AES was run in using a key size of 128 bits with an initialization vector (IV) of 4 bytes and an input equal to twice the block size. The HMAC-SHA1 was run using a key size of 20 bytes and an input of 4 bytes. The deterministic RNG was based on the RC4 stream cipher where the output was 32 bytes. Table 9 presents the algorithm runtime data along with individual clock cycles as obtained from the real time in circuit debugger. It is clear that the added latency due to the proposed algorithm computation is small and

acceptable. This will be further emphasized in the network simulation results presented in the following section.

Table 9 - Real time debugger data.

	AES128 Init.	AES128 Dec	HMAC-SHA1 Init.	HMAC-SHA1 Gen.	DRNG	Total
Clock cycles	157	481	1252	1498	961	4349
Latency (us)	1.3	4.0	10.4	12.5	8.0	36.3

4.6.2 Network Simulations

In this section, we compare different key delivery mechanisms taking into consideration the key authentication computational overhead. Two networks are adopted. The first network is a chain shaped network and the second network is a tree shaped network such that each node can have a maximum of 2 children. Figures Figure 25 and Figure 26 display the networks used for simulation. The root (node 0) represents the utility and the remaining nodes represent the smart meters in the neighbourhood area network. The distance separating the nodes is 95 meters and the solid lines connecting the nodes indicate the presence of a communication link between the two. For the tree network some children are dropped to enable sufficient spacing between nodes to maintain tree configuration in a wireless mesh network.

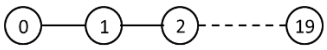

Figure 25 - Chain based network used in simulation

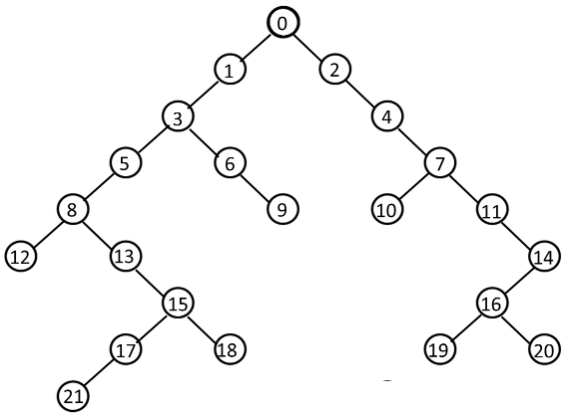

Figure 26 - Tree based network used in simulation.

We simulated the networks using the network simulator NS-3. The wireless mesh networks described are interconnected using IEEE 802.11s, which is an IEEE 802.11 amendment for mesh networking. We set the link rate to 6 Mbps and the orthogonal frequency division multiplexing (OFDM) is used as the modulation scheme. The network layer is based on IPv4 and uses the User Datagram Protocol (UDP) as the transport layer. The newly update keys are placed in the application layer. A proactive, tree-based routing protocol is used prior to running the key update to establish the routing tables. The cost of the establishment of the routing tables is excluded in the results of the simulations.

We compare the following key delivery scenarios in our simulations:

1- **Unicast**: this represents the typical one key per packet delivery.
2- **Multicast noDrop**: This represents the proposed multicast key distribution without dropping any chunk of data of the packet en route. In this scenario, a node extracts its key, and forwards the packet as is to its children without modifying the packet.
3- **Multicast withDrop**: This represents the proposed multicast key distribution. In this scenario, a node extracts its key, and then forwards to its children only the set of keys intended for their respective subtrees. We assume that routing information is available from the routing protocols.
4- **Multicast withDrop Delay**: In order to increase the authenticity of the simulations, the results obtained from the hardware assessments discussed in the previous section are embedded in the latencies of the network simulator. Hence, this represents scenario 3 with the overhead of cryptographic computations obtained in the previous. The latencies incurred are added to the network delays of scenario 3.
5- **Multicast NoDrop 1MAC**: This represents a multicast key distribution scenario with a single MAC being used to authenticate the whole packet as opposed to the proposed authentication mechanism. A node extracts its own packet and forwards it without dropping any bytes. The MAC occupies 16 bytes.

The amount of data on the application is dependent on the key delivery mechanism. For the unicast, the application layer contains a nonce used for the encryption process, the

encrypted private key and a message authentication code. Unlike the unicast, the multicast scenarios have a variable amount of data on the application layer, as it is dependent on the amount of nodes the packet is targeted to. Thus, the multicast packet contains one nonce and multiple IDs along with the corresponding encrypted blocks. In all simulations, the nonce has a size of 4 bytes, ID has a size of 4 bytes and the private key has a size of 20 bytes. The clear-text of an encrypted block corresponds to the private keys mixed with the bits of the ID as discussed in previous sections.

The two main criteria used to assess the performance of the proposed key delivery mechanism are (1) the latencies incurred on the delivery of the keys and (2) the traffic generated during the key delivery process. Traffic assessment helps identify traffic footprints of the different methods. Reduction in the traffic aids in reducing the congestion seen in the wireless medium, thereby facilitating the wireless medium access to regular smart meter data.

For the chain network, we assess the benefits of the different scenarios. Figure 27 presents the normalized latency with respect to the maximum unicast delay and averaged over consecutive runs. Table 10 and Table 11 present the average traffic reduction for the chain-based network. We observe the following:

1. Compared to unicast, the proposed methodology leads up to 70% reduction in latency. This is an extra 20% compared to the 1-MAC multicast approach. The delays introduced by the authentication scheme discussed in the previous section do not impose timing overheads and can be considered negligible.
2. Compared to *Unicast,* the *Multicast withDrop* reduces traffic on average by 20%.

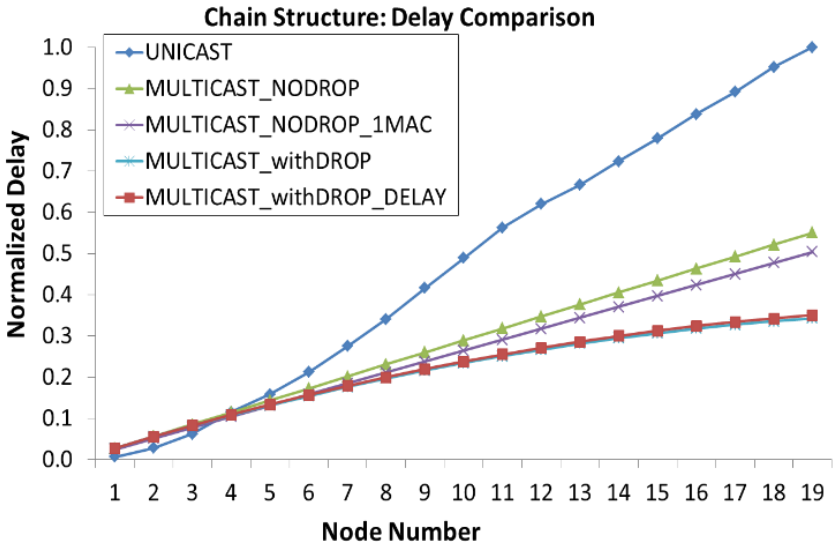

Figure 27 - Delay comparison for the chain based network

Table 10 - Average reduction in transmitted data in chain based network

	Multicast noDrop 1-MAC	Multicast withDrop
Unicast	12%	20%
Multicast noDrop 1-MAC	-----	8%

Table 11 - Average reduction in received data in tree based network.

	Multicast noDrop 1MAC	Multicast withDrop
Unicast	10%	22%
Multicast noDrop 1MAC	-----	12%

Figure 28 compares the averaged and normalized latency values obtained at each node for the tree-based network. As for the traffic observed during the key update process, Tables Table 12 and Table 13 present the average reduction in the transmitted and received traffic packets in the tree-based network.

1. As expected, the *Multicast withDrop* scenario reduces the latency compared to the *multicast NoDrop 1MAC* since multiple keys are dropped upon forwarding. On average

the *Multicast withDrop* reduces the latency up tp 80% compared to the *Unicast* approach and by 44% compared to the *Multicast noDrop 1MAC*.

2. The *Multicast withDrop* approach achieves on up to 30% reduction in the amount of traffic compared to both the *Unicast* and *Multicast noDrop 1MAC* scenarios.

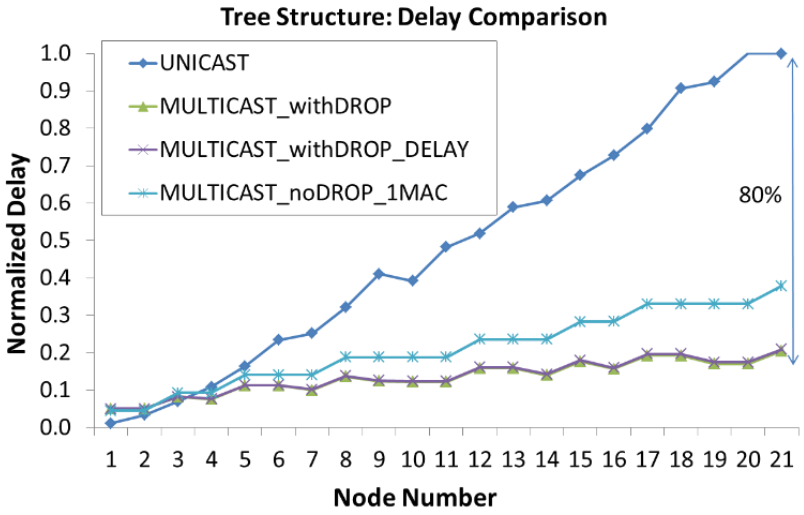

Figure 28 - Delay comparison for the tree based network

Table 12 - Average reduction in transmitted data in tree based network.

	Multicast noDrop 1MAC	Multicast withDrop
Unicast	6%	27%
Multicast noDrop 1MAC	-----	22%

Table 13 - Average reduction in received data in tree based network.

	Multicast noDrop 1MAC	Multicast withDrop
Unicast	9%	31%
Multicast noDrop 1MAC	-----	23%

Taking a closer look at the received bytes at each node, Figure 29 shows that the *Unicast* based key delivery generates the most traffic and the *Multicast withDrop* generates the least traffic. Note that nodes such as 9 and 10 in the tree structure of Figure Figure 26 are leaf nodes, and this explains the low received traffic in Figure 29.

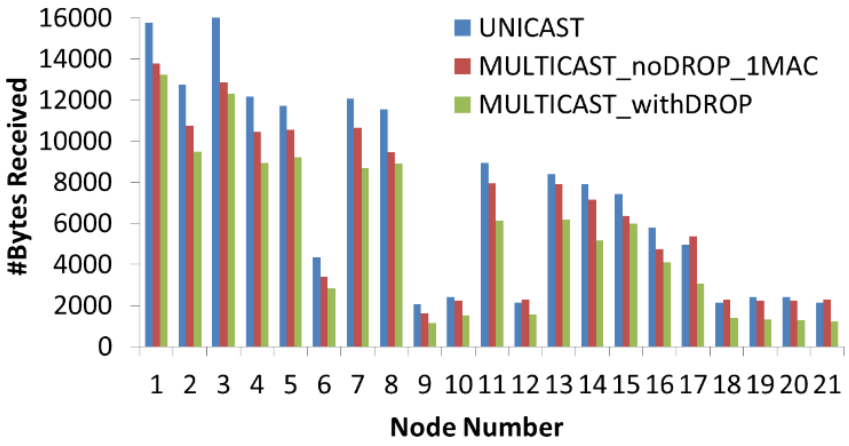

Figure 29 - Bytes received at each node for a tree based network.

Chapter 5 – Efficient Forward Secure Meter-to-Utility Communication

5.1 Introduction & Motivation

Meter to utility communication at the application layer contains sensitive data such as real time metering data or data related to demand and response messages and dynamic pricing. The data transmitted contains private and sensitive user information that must be protected from adversaries. For example, if this data is captured and analysed by an adversary, she may forecast about household activities as show in Figure 30 [36].

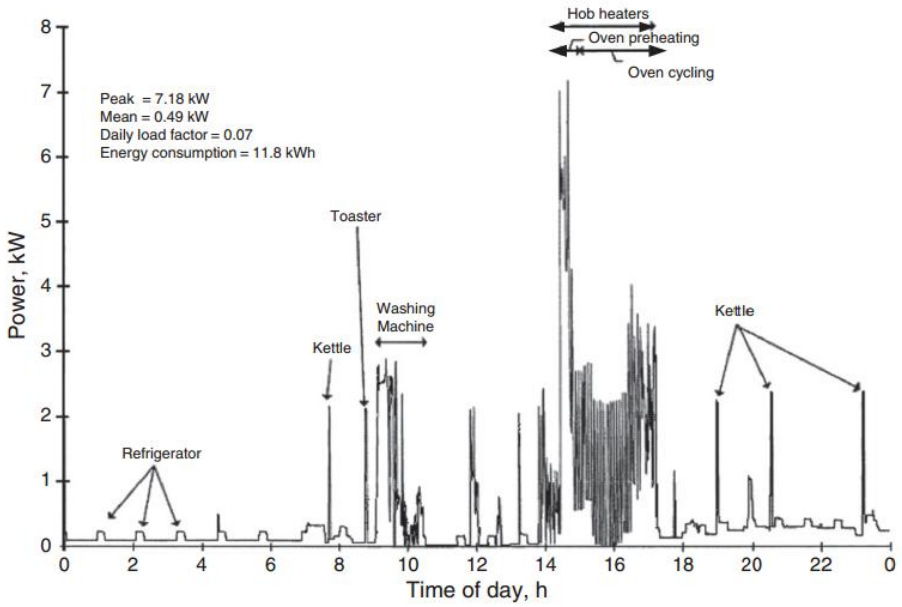

Figure 30 - Power usage to personal activity mapping

The following privacy threats, which might be exploited by criminals, governments etc., might arise:

- Determining personal behaviour patterns
- Determining specific appliances used
- Performing real time surveillance
- Target home invasions

Assuming a long-term secret key is used is used to encrypt the transmitted data, one undertakes that the data is safe from adversaries because adversaries do not have access to the secret keys used to secure communication. But what if the adversary was logging all the encrypted data? And what if at some future instance he was able to retrieve the adopted long-term key? Although hard, a committed adversary might be able to gain access to the key by one of the following ways:

- Compromising the utility's database of secret keys
- Exploiting the weak cipher suites used
- Extracting key from neighborhood meters if a single shared key is used

Once the adversary is able to retrieve the key, she will be able to compromise to break any past or future communications. In order to solve this problem, a forward secure communication needs to be adopted based on volatile session keys. In this chapter an efficient forward secure communication scheme is proposed based on identity based non-interactive key exchange and elliptic curve Diffie Hellman.

5.2 Diffie Hellman & Forward Secrecy

Diffie Hellman key exchange described in section 2.2.5 can be used as a key exchange between 2 parties. In order to achieve a forward secure communication based on the keys obtained from DH, the ephemeral DH (EDH) has to be used where the secret keys used are generated (non-static). The keys generated from EDH are generated per session and used as the symmetric random volatile keys for the cryptographic blocks. This makes the generated keys independent of past or future keys, thereby achieving forward secrecy. It is necessary to authenticate the EDH key exchange in order to prevent man in the middle (MITM) based attacks on the key exchange.

From a high level, the authentication can be based on asymmetric crypto where certificates and signatures are used or it can be based on symmetric crypto where message authentication codes (MAC) are used to validate the authenticity of the key exchange. Comparing the 2 approaches, in general, it can be stated that asymmetric crypto is computationally more demanding than symmetric crypto, however symmetric crypto has a

burden in key distribution. The question to be asked is: is it possible to merge the benefits of both in 1 scheme?

5.3 Elliptic Curve Diffie Hellman

In the scheme proposed, the elliptic curve version of the Diffie Hellman (ECDH) will be used instead of the regular DH. As discussed in section 2.2.5, 2 parties A & B have to agree on an elliptic curve $E(\mathbb{F}_p)$ and a point $P \in E(\mathbb{F}_p)$. The parties then choose secret integers n_A and n_B respectively and calculate their public keys $Q_A = n_A P$ and $Q_B = n_B P$ respectively. After exchanging the public keys, the 2 parties will be able to derive a symmetric key $SK_{A,B} := n_B Q_A = (n_A n_B)P = n_A Q_B$. This is illustrated in the figure below where parties A & B are denoted by Alice & Bob.

Public Parameter Creation	
A trusted party chooses and publishes a (large) prime p, an elliptic curve E over $\mathbb{F}_p$, and a point P in $E(\mathbb{F}_p)$.	
Private Computations	
Alice	**Bob**
Chooses a secret integer n_A. Computes the point $Q_A = n_A P$.	Chooses a secret integer n_B. Computes the point $Q_B = n_B P$.
Public Exchange of Values	
Alice sends Q_A to Bob $\longrightarrow$	Q_A
Q_B $\longleftarrow$	Bob sends Q_B to Alice
Further Private Computations	
Alice	**Bob**
Computes the point $n_A Q_B$.	Computes the point $n_B Q_A$.
The shared secret value is $n_A Q_B = n_A(n_B P) = n_B(n_A P) = n_B Q_A$.	

Figure 31 – ECDH key exchange illustration

In our scheme, parties A & B are the smart meter and the utility.

ECDH is chosen over DH since breaking the DLP requires operations of order $O\left(e^{c\sqrt{(\log p)(\log\log p)}}\right)$ thus taking sub-exponential time to solve; whereby breaking the ECDLP requires $O(\sqrt{p})$ thereby considering it to have an **exponential** runtime. This practically results in much smaller security parameters to achieve same levels of security. Other than providing the public parameters provided during the **Setup** phase used for the ID-based cryptosystem, the utility has to provide the public parameters used for the ECDH. The public parameters are:

- Prime p that specifies size of finite field
- Coefficients a and b of the Weierstrass equation $y^2 = x^3 + ax + by$

- base point $\mathbf{G}$ that generates the subgroup
- order $\mathbf{n}$ of the subgroup

5.4 Identity Based Non-Interactive Key Exchange

The non-interactive key exchange (ID-NIKE) used to derive the authentication keys for the EDH at the application layer will build upon the same mathematical concepts discussed in chapter 3 for the non-interactive key distribution system used for the security of the link layer.

5.5 Proposed Methodology

Once any of the smart meter or the utility wants to communicate with the other party, it has to first establish a volatile session key, after which this symmetric session key is used in the needed cryptographic blocks. The steps to complete the key exchange are depicted in Figure 32 where the initiator of the connection is assumed to be the smart meter. RND is a random number and T is a timestamp in seconds used to add entropy into the system and protect it against replay attacks. ID_i and ID_r are the identities of the initiator and the responder respectively. Both are known by the 2 parties without exchanging any additional packets. $\boldsymbol{d_i}$ is the volatile secret key generated and $\boldsymbol{Q_i}$ is the public key. Note that $\boldsymbol{Q_i} = \{\boldsymbol{x_i}, \boldsymbol{y_i}\}$ is a point on the elliptic curve and can be denoted as a pair. In order to further optimize the runtime of the proposed scheme, both the initiator and the responder can precalculate the pair $\{\boldsymbol{d_i}, \boldsymbol{Q_i}\}$ and $\{\boldsymbol{d_r}, \boldsymbol{Q_r}\}$ and store them in memory to be used later once a session is to be opened. The function $\hat{e}()$ used to generate the symmetric key exhibits the 3 useful bilinear properties of (1) bilinearity, (2) non-degeneracy and (3) computability as discussed in the previous sections. Note that while performing the calculations of Figure 32, the session key derived will be a point on the elliptic curve and will have an $\boldsymbol{x}$ component and a $\boldsymbol{y}$ component. As described in [19], the $\boldsymbol{y}$ component will be neglected and the $\boldsymbol{x}$ component will be treated as the shared session key. The session key will be used as the key perform the current session's symmetric crypto.

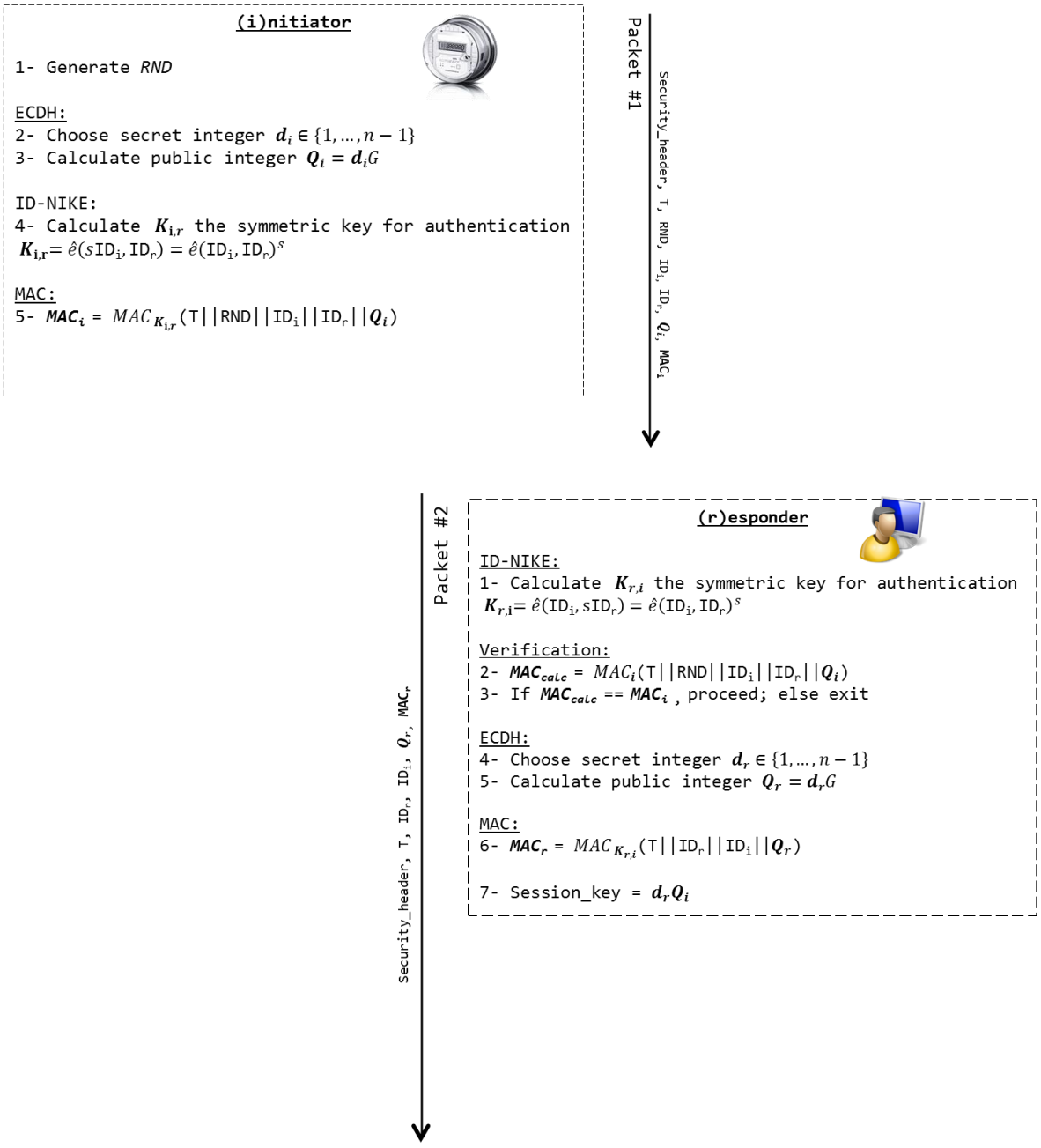

Figure 32 – ID-NIKE with ECDH

5.6 Simulations & Results

In this section we evaluate and compare the proposed system with the existing literature in smart grid security related to forward secure communications and DH based key exchanges within the AMI network. In the first part of this section, hardware tests will be performed in order to evaluate the cryptographic blocks and latencies incurred within the microcontroller that will deployed within the smart meters that will be used on the field. The second part of the evaluation will rely on simulating the system in an 802.11s mesh network of meters.

5.6.1 Hardware Testing

To evaluate the latencies incurred by the cryptographic blocks used in hardware, we evaluate each of the following cryptographic blocks in an ARM cortex microcontroller used by the smart meter manufacturers and vendors. The microcontroller used is the STM32F407 that belongs to the ARM cortex M4 family with a maximum clock speed running at 168 MHz. The results obtained will be embedded within the network simulator of the wireless mesh network. The cryptographic blocks used within our scheme are: HMAC-SHA1, ECDH-ID-NIKE and ECDH-RSA. HMAC-SHA1 and ECDH-ID-NIKE are blocked used within our proposed scheme. ECDH-RSA is used to evaluate the runtime of the scheme proposed in [16]. The pairings within the ID-NIKE are not evaluated because of their low frequent runtimes i.e. they are running when new keys are obtained or new neighbours discovered. We also assume that the meters perform several precomputations during its idle time such as generation of the public and secret keys used in the ECDH. shows the runtimes obtained from the hardware debugger during the runtime of the cryptographic blocks on data with realistic sizes. By realistic we mean that the size of the input to the cryptographic blocks is in lieu to what is expected when meters are deployed on the field.

Table 14 – Real time debugger data

Crypto block	Runtime	Comment
HMAC-SHA1 over 100 bytes	68.8 us	Used in ECDH-RSA
HMAC-SHA1 over 70 bytes	60.27 us	Used in ECDH-ID-NIKE
ECDH-224 pairwise key Generation	164.18 ms	-
RSA-2048 encryption	57.86 ms	Performed over 70 bytes
RSA-2048 decryption	944 ms	Performed over 70 bytes

Note that according to NIST [33], Table 7 shows that ECDH-224 and RSA-2048 achieve the same level of security.

5.6.2 Network Simulations

In this section, we evaluate the proposed system and compare it with that of [16]. We perform the evaluations based on square networks of different sizes as show in Figure 33 ($N \times N$ network). The green node represents the data concentrator unit (DCU) and the remaining nodes represent the smart meters in the neighbourhood area network. The distance separating the nodes is 95 meters and the solid lines connecting the nodes indicate the presence of a communication link between the two. We simulated the networks using the network simulator NS-3. The wireless mesh networks described are interconnected using IEEE 802.11s, which is an IEEE 802.11 amendment for mesh networking. We set the link rate to 6 Mbps and the orthogonal frequency division multiplexing (OFDM) is used as the modulation scheme. The network layer is based on IPv4 and uses the User Datagram Protocol (UDP) as the transport layer. A proactive, tree-based routing protocol is used prior to running the key update to establish the routing tables. The cost of the establishment of the routing tables is excluded in the results of the simulations. The simulation is ran for 30 seconds where every meter sends a packet every 4 seconds.

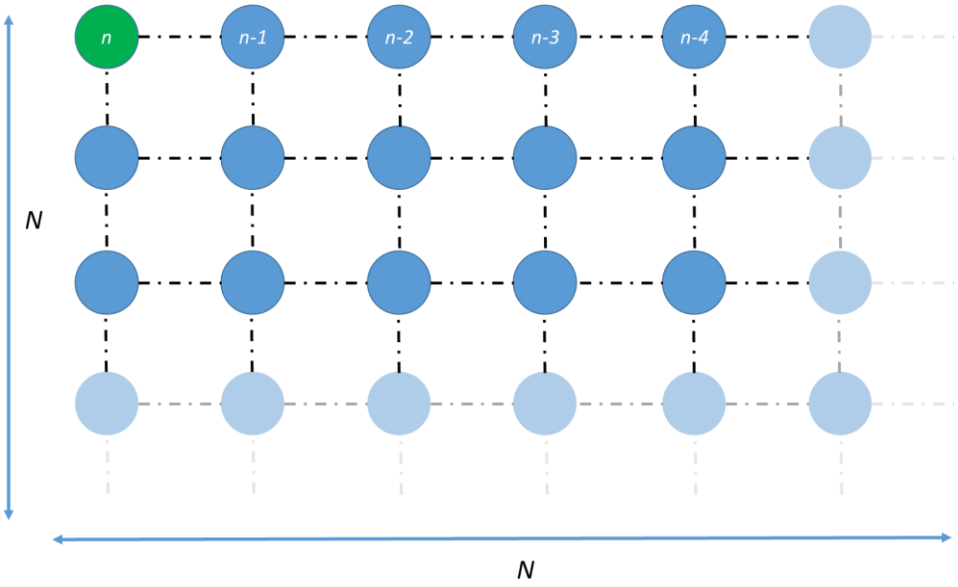

Figure 33 - Square network used in simulations

2 square networks with different sizes were evaluate:

- Network (1): 7x7 square grid with 48 meters and 1 DCU
- Network (3): 11x11 square grid with 120 meters and 1 DCU

The goal of the simulations is to evaluate the round-trip-time (RTT) of the regular meter data sent from the meter to the utility and the amount of traffic generated for a forward secure meter-to-utility data communication. Therefore, the RTT and the traffic generated can be seen as the 2 main criteria to assess the proposed system. The RTT will include the timing overhead incurred while cryptographic keys are generated, negotiated or exchanged and the time needed to send 116 bytes of data at the application layer and obtain a confirmation of 46 bytes (both of which are authenticated). Traffic assessment helps identify traffic footprints of the different methods. Reduction in the traffic aids in reducing the congestion seen in the wireless medium, thereby facilitating the wireless medium access to regular smart meter data.

The simulations performed are divided into 3 scenarios:

ECDH-RSA: In this scenario the meter initiates a key exchange. This key exchange is based on the ECDH-224 key exchange and RSA-2048 as in [16]. The packet at the application sums up to 256 bytes.

ECDH-ID-NIKE: In this scenario the meter initiates a key exchange. This key exchange is based on the ECDH-224 key exchange and ID-NIKE. The contents of the packet are described in section 5.5. The packet at the application layer is depicted in Figure 34. The size of the packet sums of to 86 bytes.

Sec_Header (2 bytes)	RND (4 bytes)	ID_i, ID_r (8 bytes)	{Qx,Qy} (56 bytes)	MAC (16 bytes)

Figure 34 - Packet form at the application layer for ECDH-ID-NIKE

Normal-Data: In this scenario the meter transmits periodic meter data to the utility. The transmitted data is assumed 100 bytes [37]. A message authentication code generated on the 100 bytes is appended to the packet by HMAC-SHA1, making the size of the packet 116 bytes. The key used is either derived from ECDH-RSA or ECDH-ID-NIKE. The response for this packet is 30 bytes and its MAC is 16 bytes, making the total response size 46 bytes.

Figures 35-37 present the data resulted from the simulations performed on the 7x7 network. The horizontal axis represents the node number. The numbering is made on the basis that the DCU has the highest node number (most upper left node) as depicted in Figure 33. The rest of the nodes are numbered in a decremented order. The one on the right side of the DCU has a node# of n-1 and the one below the DCU has a node# of n-N-1 with N being the size of the edge of the network. The vertical axis has a unit of ms.

Figure 35 displays the time needed for a regular authenticated meter data of 116 bytes to be transmitted to the utility plus the time needed to get a reply of size 46 bytes.

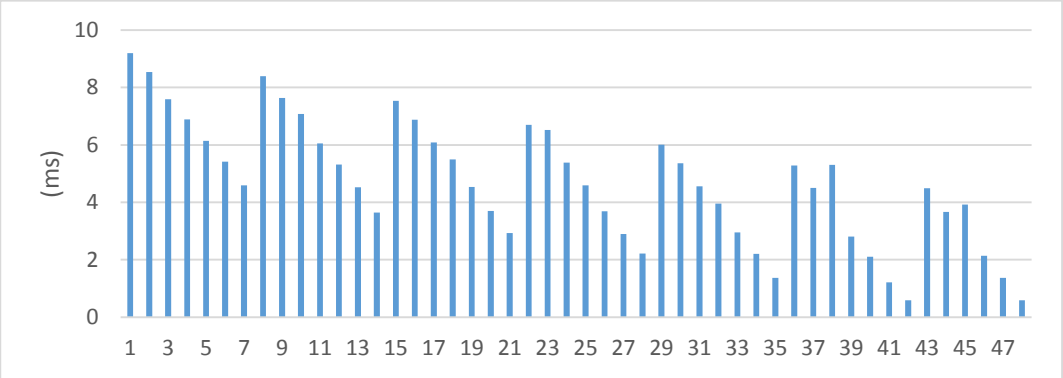

Figure 35 - Normal metering data in 7x7 network

In figures 36 and 37, the vertical axis denotes the time needed to exchange the keys between the utility and meter. Note that in all figures, the hardware delays are integrated. As can be noticed in all of the figures, the closest nodes to the DCU have the least delays which is logical as the number of hops required to reach the DCU is smaller.

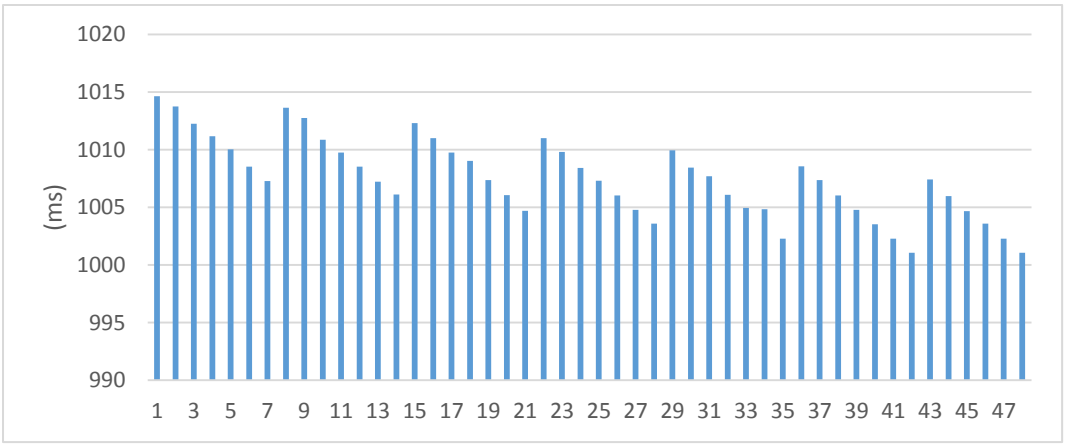

Figure 36 - ECDH-RSA key exchange in 7x7 network

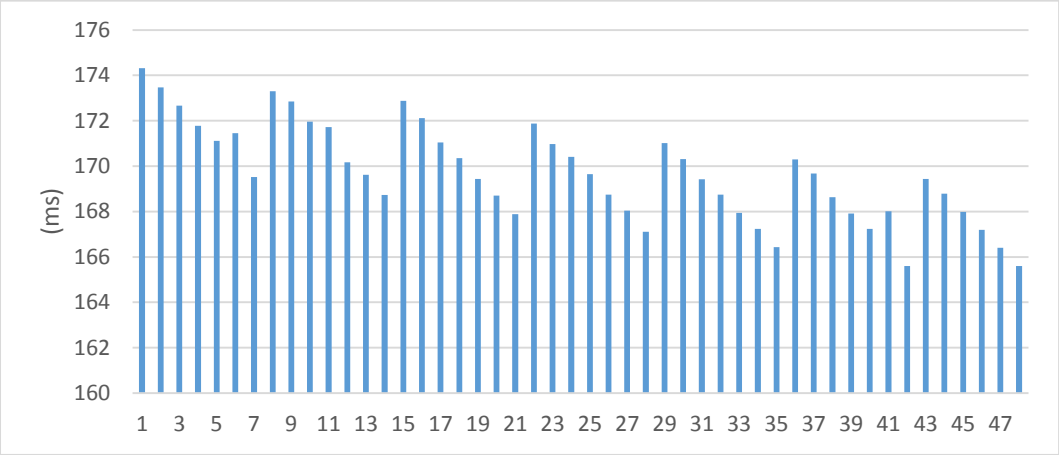

Figure 37 - Proposed key exchange in 7x7 network

For a 7x7 network, the averaged RTT in ECDH-ID-NIKE is found to be 174ms, whereas in the scheme of [16] it is found to be 1012ms.

Figures 38-40 present the data resulted from the simulations performed on the 11x11 network. The numbering of the nodes and the conventions used are similar to that of the figures that present the 7x7 network.

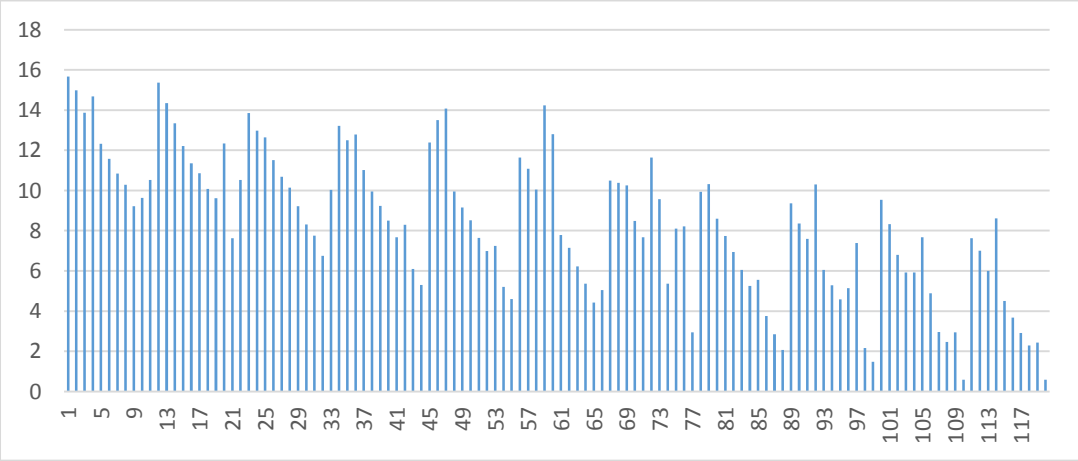

Figure 38 - Normal metering data in 11x11 network

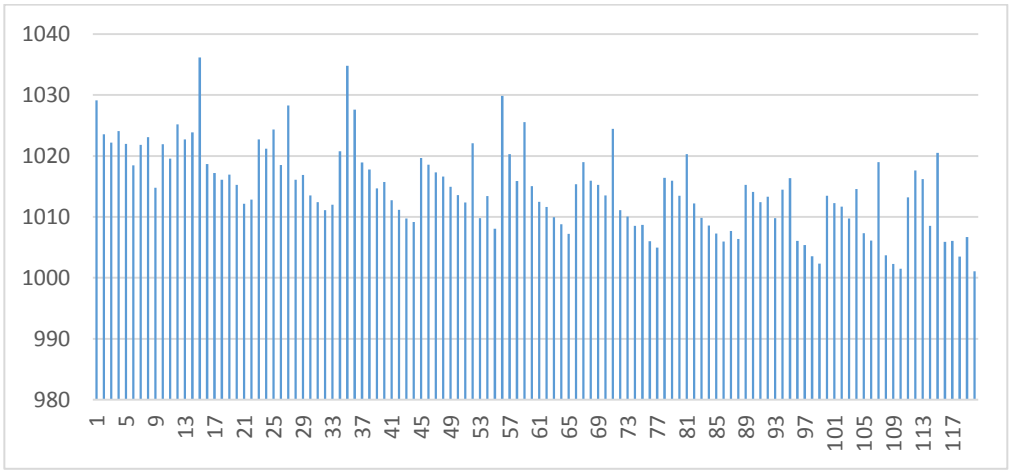

Figure 39 - ECDH-RSA key exchange in 11x11 network

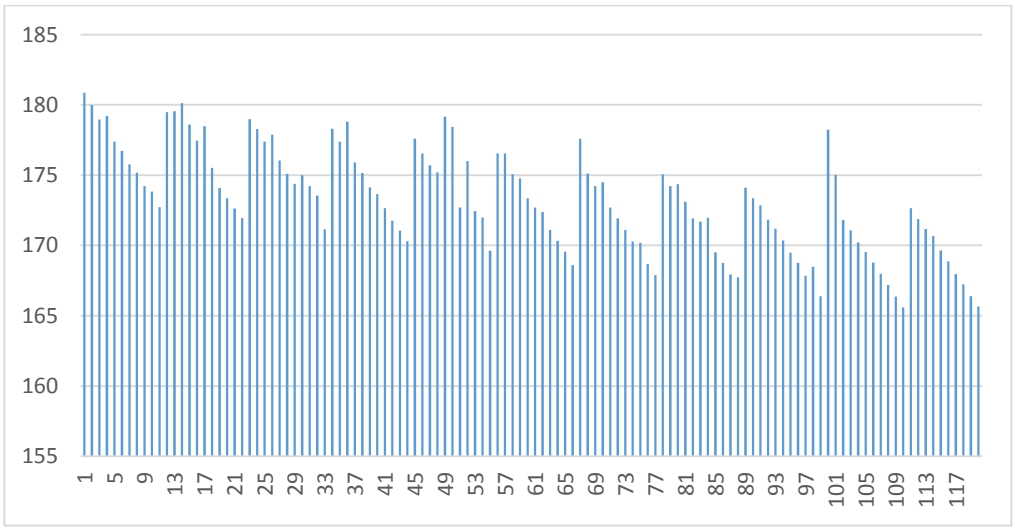

Figure 40 - Proposed key exchange in 11x11 network

For a 11x11 network, the averaged RTT in our scheme is found to be 181ms, whereas in the scheme of [16] it is found to be 1023ms.

Table 15 and Table 16 show present the average transmitted and received bytes in the 7x7 and 11x11 networks. These numbers were harvested by observing and averaging the total number of bytes transmitted and received on each wireless network interface of every smart meter on the network. Note that the number of bytes are not only related to the application layer, but also to all the bytes transmitted and received for the lower OSI layers. Both the size of the network and the protocol implemented impact on the traffic generated. As the size of the network increases or as the number of byte exchanges needed

for a protocol increases, the amount of the transmitted and received bytes increases. A similar phenomenon can be also observed when the network is congested and collisions are taking place on the wireless links, leading to retransmissions.

Table 15 - Average number of Rx bytes

	7x7 Network	11x11 Network
ECDH-RSA	51295	96679
ECDH-ID-NIKE	34461	61536

Table 16 - Average number of Tx bytes

	7x7 Network	11x11 Network
ECDH-RSA	39747	70866
ECDH-ID-NIKE	24066	38959

Based on Tables Table 15 and Table 16 it is found that ECDH-ID-NIKE has a better network footprint than ECDH-RSA. Adopting the proposed system in a 7x7 network, the average amount of bytes transmitted is less than 40% than that of ECDH-RSA whereas the received bytes are less by about 33%. As for the 11x11 network the amount of bytes transmitted is less than 45% whereas the received is less by 36%.

By comparing the latencies incurred and the traffic generated of both ECDH-RSA and ECDH-ID-NIKE we can see that ECDH-ID-NIKE outperforms ECDH-RSA.

Chapter 6 – Conclusion

In this thesis we have focused on the security of the advanced metering infrastructure (AMI) network, which is seen as the last mile of the smart grid, connecting smart meters to the utility.

We have covered topics related to efficient and scalable key management by relying on identity based cryptographic primitives.

In the first part of the thesis we propose an identity based non-interactive and scalable key distribution system for generating pairwise symmetric keys to be used for link-layer security. In addition, to prevent the discovery and access of the cryptographic keys stored on accessible memory locations physical unclonable functions are adopted.

The second part of the thesis presents a new lightweight key update and delivery method to securely update the private keys utilized by the identity based cryptosystem.

The last part of the thesis introduces a new method for deriving keys for the purpose of ensuring a forward secure communication between the meters and the utility at the application level. The proposed method is based on merging identity based cryptography and Diffie-Hellman key exchange for the purpose of deriving forward secure keys.

All proposed ideas and algorithms are assessed on hardware platforms that mimic processors used by smart meter vendors and on event driven network simulators in order to simulate the proposed systems in the context of AMI wireless mesh networks.

References

[1] W. Wang and Z. Lu, "Cyber security in the Smart Grid: Survey and Challenges," *The International Journal of Computer and Telecommunications Networking,* vol. 57, no. 5, pp. 1344-1371, 2013.

[2] Z. Fan, P. Kulkarni, S. Gormus and C. Efthymiou, "Smart Grid Communications: Overview of Research Challenges, Solutions, and Standardization Activities," *IEEE Communications Surveys & Tutorials,* vol. 15, no. 1, pp. 21-38, 2012.

[3] Y. Q. H. S. a. D. T. Ye Yan, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 1, 2013.

[4] Y. Yan, Y. Qian, H. Sharif and D. Tipper, "A Survey on Cyber Security for Smart Grid Communications," *IEEE Communications Surveys & Tutorials,* vol. 14, no. 4, pp. 998 - 1010, 2012.

[5] D. Grochocki, J. H. Huh, R. Berthier, R. Bobba, W. H. Sanders, A. A. Cárdenas and J. G. Jetcheva, "AMI threats, intrusion detection requirements and deployment recommendations," in *Smart Grid Communications (SmartGridComm), 2012*, Tainan, 2012.

[6] "The Smart Grid Interoperability Panel - Cyber Security Woking Group, Guidlines for smart grid cyber security, NISTIR".

[7] Z. Fan, P. Kulkarni, S. Gormus and C. Efthymiou, "Smart Grid Communications: Overview of Research Challenges, Solutions, and Standardization Activities," *IEEE Communications Surveys & Tutorials,* vol. 15, no. 1, pp. 21 - 38, 2013.

[8] F. M. Cleveland, "Cyber security issues for Advanced Metering Infrasttructure (AMI)," in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, Pittsburgh, PA, 2008.

[9] D. P. P. M. Stephen McLaughlin, "Energy Theft in the Advanced Metering Infrastructure," in *Critical Information Infrastructures Security*, pp. 176-187.

[10] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka, A. Delozier and P. McDaniel, "Multi-vendor penetration testing in the advanced metering infrastructure," in *Proceedings of the 26th Annual Computer Security Applications Conference*, 2010.

[11] R. Anderson and S. Fuloria, "Who Controls the off Switch?," in *Smart Grid Communications (SmartGridComm)*, Gaithersburg, MD, 2010.

[12] H. Li, L. Zhou and B. Yang, "An Efficient Merkle-Tree-Based Authentication Scheme for Smart Grid," *IEEE Systems Journal,* vol. 8, no. 2, pp. 655 - 663, 2014.

[13] K. Akkaya, K. Rabieh, M. Mahmoud and S. Tonyali, "Efficient Generation and Distribution of CRLs for IEEE 802.11s-based Smart Grid AMI Networks," in *IEEE International Conference on Smart Grid Communications*, 2014.

[14] Y. Yan, R. Q. Hu, S. K. Das and H. Sharif, "An efficient security protocol for advanced metering infrastructure in smart grid," *IEEE Network,* vol. 27, no. 4, pp. 64 - 71, 2013.

[15] H. K. H. So, S. H. M. Kwok, E. Y. Lam and K. S. Lui, "Zero-Configuration Identity-Based Signcryption Scheme for Smart Grid," in *Smart Grid Communications (SmartGridComm), 2010*, Gaithersburg, MD, 2010.

[16] M. M. Fouda, Z. M. Fadlullah, N. Kato and R. Lu, "A Lightweight Message Authentication Scheme for Smart Grid Communications," *IEEE Transactions on Smart Grid,* vol. 2, no. 4, pp. 675 - 685, 2011.

[17] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, San Diego, CA, 2007.

[18] M. Nabeel, S. Kerr, X. Ding and E. Bertino, "Authentication and key management for Advanced Metering Infrastructures utilizing physically unclonable functions," in *Smart Grid Communications (SmartGridComm), 2012*, Tainan, 2012.

[19] J. P. J. S. J. Hoffstein, An Introduction to Mathematical Cryptography, Springer, 2008.

[20] Y. L. Jonathan Katz, Introduction to Modern Cryptography, Second Edition, Chapman and Hall/CRC , 2014.

[21] W. Stallings, Cryptography and Network Security: Principles and Practice (6th Edition).

[22] M. F. Dan Boneh, "Identity-Based Encryption from the Weil Pairing," 2001.

[23] S. W. Smith, Cryptographic scalability challenges in the smart grid..

[24] H. Khurana, M. Hadley, N. Lu and D. A. Frincke, "Smart-grid security issues," *IEEE Security & Privacy,* vol. 8, no. 1, pp. 81 - 85, 2010.

[25] A. E. Régis Dupont, "Provably secure non-interactive key distribution based on pairings," *Discrete Applied Mathematics,* vol. 154, no. 2, p. 270–276, 2006.

[26] S. K. Michael S. Kirkpatrick, "PUF ROKs: a hardware approach to read-once keys," in *6th ACM Symposium on Information, Computer and Communications Security*, 2011 .

[27] G. E. Suh, C. W. O'Donnell, I. Sachdev and S. Devadas, "Design and implementation of the AEGIS single-chip secure processor using physical random functions," in *Computer Architecture, 2005. ISCA '05. Proceedings.*, 2005.

[28] K. Paterson, "Cryptography from Pairings," in *Advances in Elliptic Curve Cryptography*, Cambridge University Press , pp. 215-252.

[29] A. N. S. P. S. f. D. C. Networks, "ANSIC 12 in context," 2008.

K.-H. Chang and B. Mason, "The IEEE 802.15.4g standard for smart metering

30] utility networks," in *Smart Grid Communications (SmartGridComm)*, 2012.

31] G. R. Hiertz, D. Denteneer, S. Max and R. Taori, "IEEE 802.11s: The WLAN Mesh Standard," *IEEE Wireless Communications,* vol. 17, no. 1, pp. 104 - 111, 2010.

32] G. Leon, "Smart Planning for Smart Grid AMI Mesh Networks".

33] E. Barker, "Recommendation for Key Management - Part 1: General (Revision 3)," NIST Special Publication 800-57 Part 1, Revision 3, 2012.

34] V. S. Dan Boneh, A Graduate Course in Applied Cryptography, 2015.

35] "WolfSSL Embedded SSL Library".

36] K. A. Nico Saputro, "On preserving user privacy in Smart Grid advanced metering infrastructure applications," *Security and Communication Networks,* vol. 7, no. 1, 2014.

37] "High-Level Smart Meter Data Traffic Analysis," Engage Consulting Ltd. for the Energy Networks Association (ENA), London, U.K.:, 2010.

38] G. D. M. K. Klaus Kursawe, "Privacy-Friendly Aggregation for the Smart-Grid," in *11th International Symposium, PETS 2011*, Waterloo, ON, Canada, 2011.