Design of security mechanism for communication networks in smart grid

Gurbakshish Singh Toor

2016

Gurbakshish Singh Toor. (2016). Design of security mechanism for communication networks in smart grid. Master’ s thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/69109

https://doi.org/10.32657/10356/69109

DESIGN OF SECURITY MECHANISM FOR COMMUNICATION NETWORKS IN SMART GRID

GURBAKSHISH SINGH TOOR

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

2016

DESIGN OF SECURITY MECHANISM FOR COMMUNICATION NETWORKS IN SMART GRID

GURBAKSHISH SINGH TOOR

School of Electrical & Electronic Engineering

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

2016

ACKNOWLEDGEMENTS

My sincerest gratitude to my supervisor, *Prof. Maode Ma* from Nanyang Technological University (NTU), for his invaluable guidance, support and constant encouragement throughout the course of my Masters Research degree. His support, technical discussions and suggestions have enabled me to become a keen researcher. The profound knowledge in network security, excellent writing skills and great research integrity possessed by *Prof. Ma* has inspired my enthusiasm in research field. *Prof. Ma* is also a mentor of my life, who always provides me with priceless opinions for career and life. It is quite delightful and lucky to have a chance to work under *Prof. Ma*'s supervision.

I would also like to thank *Mr. Ng Teng Kwee* for his help and guidance in accessing the Lab and his technical suggestions.

I dedicate this dissertation to my parents and my brother. Their love, support and encouragement have always been the motivation of my life.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS I

TABLE OF CONTENTS II

SUMMARY V

Chapter 1. Introduction 1

1.1 Background 1

1.1.1 Architecture 4

1.2 Security of Smart Grid Networks 7

1.2.1 Security Goals 7

1.2.2 Security Attacks 9

1.2.2.1 Insider Attacks 9

1.2.2.2 External Attacks 10

1.3 Motivation 11

1.4 Contribution 13

1.5 Organization 14

Chapter 2. Literature Review 15

2.1 Solutions for Authenticity 15

2.2 Solutions for Other Goals 18

2.2.1 Solutions for Privacy 18

2.2.2 Solutions for Integrity 21

2.2.3 Solutions for Availability 23

2.2.4 Solutions for Authorization 25

2.2.5 Solutions for Non-repudiation 26

Chapter 3. Security Enhancement for Dynamic Key Refreshment in Neighborhood Area Network 28

3.1 System Model 30

3.2 Vulnerability of the MKHSH Scheme 33

3.3 Proposed Scheme 37

3.4 Logical Analysis and Formal Verification 40

3.4.1 Logic Derivation 41

3.4.2 Formal Verification 47

3.5 Efficiency Analysis and Simulation Results 49

3.5.1 Efficiency Analysis 49

3.5.2 Simulation Results 51

3.6 Summary 54

Chapter 4. SDN Based Authentication Scheme for Neighborhood Area Network 55

4.1 SDN Architecture and Preliminary 56

4.1.1 SDN Architecture 57

4.1.2 One-way Accumulator 58

4.2 Proposed Scheme 60

4.2.1 System Model 60

4.2.2 Design Goals 62

4.2.3 Assumptions and Parameters 62

4.2.4 Detailed Scheme 64

4.3 Security Analysis and Formal Verification 70

4.3.1 Logic Derivation 70

4.3.2 Security Analysis 77

4.3.3 Formal Verification 79

4.4 Efficiency Analysis and Simulation Results 83

4.4.1 Efficiency Analysis 83

4.4.2 Simulation Results 85

4.5 Summary 88

Chapter 5. Conclusion and Future Work 89

BIBLIOGRAPHY 92

Appendix A 100

Appendix B 108

Design of Security Mechanism for Smart Grid Networks

Gurbakshish Singh Toor
School of Electrical and Electronic Engineering,
Nanyang Technological University

SUMMARY

The evolution of the traditional electricity infrastructure into smart grids promises more reliable and efficient power management, more energy aware consumers and inclusion of renewable sources for power generation. These fruitful promises are attracting initiatives by various nations all over the globe in various fields of academia. However, this evolution relies on the advances in the information technologies and communication technologies and thus is inevitably prone to various risks and threats. Even though many solutions have been proposed in the recent literature to overcome the security threats in smart grid networks, many issues still need to be addressed to make smart grids a reliable and efficient innovation. In this thesis, we first introduce the background, network architecture, security threats and the security requirements of smart grid networks. Our work focuses on the security aspects of Neighborhood Area Network (NAN) subsystems of smart grid. We present some of the prominent threats and attacks, specific to this subsystem, which violate the specific security goals requisite for its reliable operation. The proposed solutions and countermeasures for these security issues presented in the recent literature have been deeply reviewed to identify the promising solutions with respect to the specific security goals. Then we propose an improved

dynamic key refreshment strategy for mesh security in the NAN and an authentication scheme based on software defined network (SDN) using dynamic one-way accumulators. The proposed dynamic key refreshment scheme can protect the mesh network system based on IEEE 802.11s standard from DoS attacks during the key refreshment whereby the intruder could launch the attack using the information from previous key refreshment cycle as proposed in the original key refreshment scheme. The use of simple hash based operation makes the scheme cost effective for the resource limited network devices. The proposed scheme also adds an enhancement to the sub-protocol of the original key refreshment scheme for enhanced security and reliability. The proposed SDN based authentication scheme employs one-way dynamic accumulators combined with zero-knowledge proofs for easy and cost efficient authentication process. The availability of the cross authentication among different NAN devices enables us to replicate the mesh network architecture. Using SDN as the backbone of the scheme helps us accommodate the advances of the upcoming wireless technologies where we can update the changes in the scheme conveniently. Our analysis shows that the proposed schemes can achieve the requisite authentication while withstanding multiple attacks and the balance between security and system performance is also achieved.

Keywords—NAN; Smart Grids; Security; Authentication; SDN

Chapter 1. Introduction

1.1 Background

As the technological advances are gaining pace, the demand for energy is following the same tracks all over the world. This growth calls for a much more sustainable and efficient power distributions system as opposed to the traditional ones. The amalgamation of the information and communication technologies with the traditional power systems lead to the creation of the smart grids, leading towards a smart future. This new energy infrastructure incorporates the renewable energy resources while providing optimal power consumption through real-time feedback of the power consumption information. Not only the providers but the users will greatly benefit from this new infrastructure. The future grids will allow a two-way flow of energy and information [1] between the consumers and the utility to conquer these future goals.

This evolution brings forward the deployment of smart appliances and smart meters at the consumer end [2], capable of not only monitoring the power utilization but also optimizing it via real time evaluation of the energy flow in the power infrastructure, thus enabling the consumers to contribute to the smart future. These smart meters not only report the power consumption to the grid but also allow the grid to send control signals to the user end appliances in order to manage the power consumption based on dynamic pricing, peak consumption hours, system load requirements etc., making the users more energy aware [3].

Smart grid deployment provides benefits to both the consumer and the utility ends in multitude of scenarios. Smart grid communication infrastructure allows the monitoring of real-time information regarding the power generation, transmission and user consumption. This data collaborated with the market price set by the service provider or by the users generating power at their end, will allow to rate the energy dynamically rather than having a fixed rate at all times. For instance, if the user wants energy resource during peak hours, they have to pay higher amounts. Thus by referring to these dynamic prices the users can optimize their utilization to reduce their bills and enables the service providers to help maintain efficient grid operations and automated management [4].

Smart grid infrastructure supports the exploitation of renewable energy at both the consumer and provider ends. This approach will reduce the burden on the environment to meet the current energy needs [5]. Besides integrating renewable sources, the communication infrastructure in smart grids allows efficient monitoring of power usage, using which the electricity losses can be significantly reduced. This leads to lesser consumption of carbon fuels and reduction in emission of greenhouse gases [6]. Also the optimized usage will reduce the possibilities of blackouts. The user experience will be improved significantly [7].

Although the deployment costs of smart grid technology are significant, but automating the system promises to yield long term benefits. Also the real-time generated data on distributed energy generation helps the utilities to better determine which sections or components are likely to fail or in need of replacement [8]. Smart grid also provides improvement in load shedding, the power supply is

intentionally switched off under critical situations to avoid damage to the grid system. If the demand suddenly increases in a particular section or the supply of power significantly shortfalls to meet the demand, the demand has to be reduced instantly to stabilize the grid. [9].

Although smart grid deployment provides multiple benefits, but a fruitful implementation of such infrastructure requires efficient communication channels between different elements and devices at each level of the architecture. Hence this architecture is bound to include multitude of communication standards catering to the needs of such complex and heterogeneous environment. Such degree of complexity gives ample room to the adversaries to jeopardize the system security. However, the attacks on the system by availing on of these vulnerabilities can cause significant damage causing blackouts, economic deficit or even an opportunity for terrorist activity. Hence security aspect of the future grids and future homes is attracting the recent research. However the work is still in early stages and enormous contribution is requisite to ensure the reliability of this infrastructure.

The motive of this work is to investigate the security aspects Neighborhood area network (NAN) subsystem of smart grids in detail. This domain have been emphasized because although the power generation and transmission system pose highly critical security demand and primary focus but a system is as strong as its weakest link. The unintended exploitation of the customer domain can jeopardize the entire system and cause a severe impact that we may not be accustomed to face.

1.1.1 Architecture

The model architecture provided by National Institute of Standards and Technology (NIST) describes the smart grid framework in terms of seven domains: Bulk generation, transmission, distribution, operations, service providers, markets and customer domains [10]. However, the layered model architecture has been usually considered in the recent literature which is merely the conversion of the conceptual model provided by NIST, into three layers HAN, NAN and Wide Area Network (WAN) as shown in Figure 1. The lowest layer can also include Business Area Network (BAN) or Industrial Area Networks (IAN) as well.

HAN should enable a constant interaction between the smart appliances and the Advanced Metering Infrastructure (AMI) and also interaction with the smart grid. Major HAN entities are smart appliances, smart meters and HAN gateway. Some other entities could be distributed energy resources for renewable energy generation, Plug in Electric Vehicles (PHEV) or Plug in Electric Hybrid Vehicles (PEHV). The smart home appliances of prime consideration are the heavy load appliances such as Air Conditioners, washers and dryers, pool pumps etc., whose operation can be optimized as their delayed availability can be accepted [11]. The smart appliances interact with the smart meters for reporting the energy consumption and also to receive the control commands for operation optimization. The smart meter interacts with a data aggregation entity i.e. a HAN gateway which further interacts with NAN data aggregation unit.

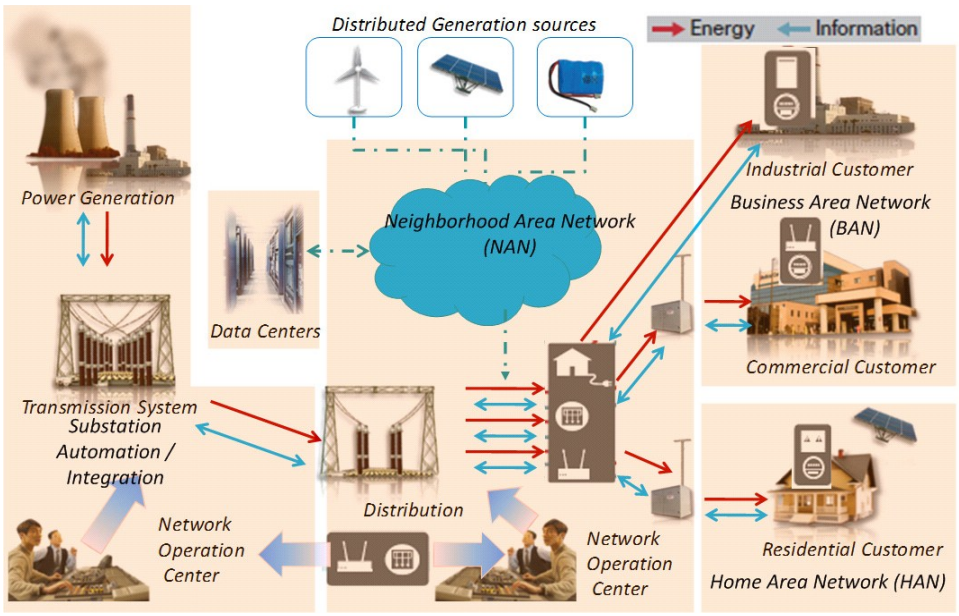

Figure1. General Architecture of Smart Grid

NANs are the communication facility of the distribution domain of smart grids allowing it to monitor the energy delivered to the customer domain and optimize the distribution according to the demand and availability options. NAN gateway collects the data from various HANs and forwards it to access points, where the data is aggregated and is further relayed it to the upper layer. Similarly, the control signals from the control centers are relayed back to HAN via NAN. Thus it bridges the gap between HAN and WAN [12]. Figure 2 shows the mesh network topology of NAN.

The final layer consists of a Wide area networks accommodating various NANs. All the aggregated data from various NAN is acquired and processed in this layer by the supervisory control and data acquisition system (SCADA). The operation of billing based on consumer consumption, load management according to the demand response paradigm and other functionalities such as outage management, acquisition and management of customer information can be included in the final layer.

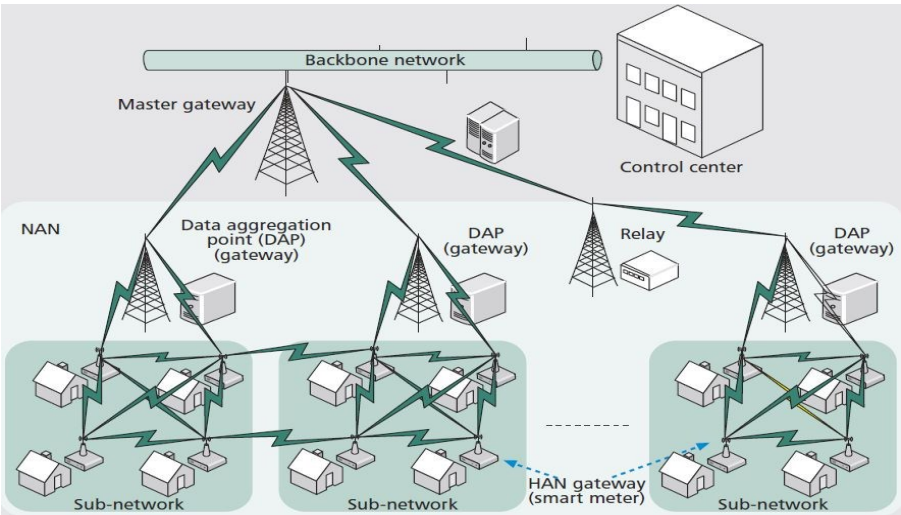

Figure2. Mesh network topology of smart grid NAN[6]

As the traditional communication networks are not suitable according to the requirements of the future grids, advanced communication systems are being analyzed in various research works to select the ones capable to satiate the needs of smart grids. The most optimal communication standard considered for HAN is ZigBee, also known as low-rate wireless personal area networks (LR-WPANs) [13], [14]. It is based on IEEE 802.15.4 standard and employs small power digital radios. The communication standards considered for NANs are IEEE 802.15.4g and IEEE 802.11s standards. 802.15.4g is the newly developed standard derived from IEEE 802.15.4 making amendments in the PHY and MAC layer to specify the requirements of Wireless Smart Metering Networks (SUNs). Another standard is IEEE 802.11s, derived from IEEE 802.11 to extend its MAC protocol for Wireless Mesh Networks. Hence it is oriented towards addressing the network operation issues of NANs [12]. A WAN connects several NANs and the coverage area is the high up to thousands of square miles. Thus the bandwidth requirements are up to 10-100Mbps and hence for WAN, optical fibers, WiMAX or cellular networks are most optimum options [14].

1.2 Security of Smart Grid Networks

Although smart grid technology provides us with numerous benefits, it is highly vulnerable to various cyber or physical attacks. The communication medium used amongst the various entities is not standardized for smart grids and the highly distributed network with large number of entities makes the problem more complex. To ensure the security of the system we must first analyze what are the requirements or goals that are to be met to make it a reliable system.

1.2.1 Security Goals

We consider the following generally security goals [11], [15], requiring the prime importance. Prime goal of security is to mitigate risk, factoring in both the likelihood of the risk and the impact it will have on the system.

- ***Integrity****:* Integrity is to ensure that the data communicated between the entities does not undergo unintentional alterations. For example, if a control command is sent by an impersonator to switch on all the appliances to millions of users during the peak hour can lead to a blackout.
- ***Authentication****:* It is used by the communicating node to validate that the other node claiming to be the intended node is in fact the intended node and similar validation for the massages sent by them. For e.g. in NAN the customer should be ensured of the NAN gateway's authenticity to which it is communicating with and gateway should be ensured that it is communicating with the assigned user equipment.

- ***Privacy**:* To ensure that the sensitive data is received by the intended party only and should not be disclosed to anyone else. For instance, the power usage information of a customer should be concealed from adversaries or neighboring HANs and NANs so that only utility receives it.
- ***Availability**:* The data or resources should be available to any authorized entity at all times. Since smart grids aim at optimizing the power consumption based on real time usage data and unavailability of system resources will hinder the transmission of such data.
- ***Authorization**:* Only the intended entities have the access to data or system resources to avoid the unintentional exploitation. For instance, the usage data of customers stored in the system should be accessible to different users such as customers, researchers or service providers according to their access priorities.
- ***Non-Repudiation**:* Non-repudiation is critical to verify the claim of truthfulness of any entity so that no authorized entity can deny the claim of sending specific information. This is a critical requirement especially in case of financial information transactions.

Since smart grid communication technology is still developing and possesses unique vulnerabilities and the design of smart grid communication networks must also fulfill the following requirements:

- Forward Security: The designed security protocols must retain the security efficacy in all the consecutive runs and the information passed on in the previous runs should not affect the security of the consecutive protocol runs.

- The architecture of smart grid infrastructure is divided into different domains having different security requirements. Hence, the security protocols must be designed to accommodate these requirements accordingly and provide a seamless transition among different domains.
- The computation and storage costs incurred by the proposed security protocols must not be dependent on the number of users present in the network, as the number of users and the smart devices in smart homes are predicted to increase significantly in the near future.

1.2.2 Security Attacks

An action either intentional or unintentional that jeopardizes the achievement of above stated goals is a security threat. These security threats are dependent on various factors and can impact the system to various degrees. Some of the major and commonly considered attacks to the HAN and NAN are stated below [16], [17]. The attacks have been categorized into insider attacks and external attacks:

1.2.2.1 Insider Attacks

In case of insider attacks, the intruder either has access to the network or collaborates with other authorized users to attack the network.

1. ***Customer Impersonation****:* In this attack the adversary impersonates as a legitimate client to the NAN. The impersonator can request the control system for the detailed consumption information of the client it is impersonating to gain the personal information.
2. ***Replay Attacks****:* It includes false data sent to the entities. In this case, an older message or signal is repeatedly used to cause system disruption. The adversary can intercept a data

consumption message and replay it to the smart meter causing to report false readings. In another scenario a signal could also be replayed to the consumer equipment.

3. ***Repudiation****:* Repudiation can also be a threat to the NAN under various scenarios. If the control center finds certain problem in the aggregated data consumption due to a significantly varying data consumption report of a consumer, then the consumer should be able to prove the authenticity of its data.

4. ***Man in the Middle Attack****:* The attacker can impersonate to be a NAN aggregator to all the HANs under it. This will lead the attacker to control the entre information flow of that specific NAN. This attack can not only cause false data reporting to the utility but can also enable the attacker to alter the control signals to large number of smart meters leading to sudden significant increase in demand or drop of demand in the system.

1.2.2.2 External Attacks

In case of external attacks, the intruder acts as an outside node and tries to abrupt the network security without access to the network.

1. ***Eavesdropping****:* Eavesdropping refers to the unauthorized monitoring of information communication between two entities with the aim of intercepting some useful information from it. The adversary gains hold of the content being communicated but does not alter it, making it difficult to detect. One of such scenarios could be an intruder trying to interpret the energy consumption patterns of an individual. This data can reveal information about an individual's

lifestyle, such as when he is at home or not, what devices are being used at what times, even the travelling patterns can be revealed [18].

2. ***False Data Injection****:* In case of false injection of data, the adversary attempts to alter the data or introduce fraudulent data in some specific pattern rather than sending random bogus signals. The attacker gains access to the communication network. Then it can confuse the system involving large number of subscribers by sending fraudulent signals [19]. In case of NAN the data being transferred to the NAN aggregation unit could be altered resulting in false reporting. This will mislead the utility in estimating the demands of a NAN region [20].

3. ***Jamming****:* Jamming is one of the physical layer attacks where a jammer fills the communication medium with noise, rendering the entities from communicating. Such attacks can also lead to distortion or destruction of sensitive and valuable data [19].

4. ***Access Restriction****:* This is similar to jamming attack but here the number of targeted entities is larger. The jammer tries to occupy the communication medium first every time so that other nodes will sense the medium busy and suspend their transmissions, thus blocking the legitimate nodes from communication initiation or may cause packet collision.

1.3 Motivation

Smart Grids offer many fruitful outcomes and will bring a paradigm shift to our day to day lives. They promise a more sustainable future and efficient use of energy while incorporating the renewable energy in hand. The deployment of such infrastructure will bring great economic and environmental

benefits, improvement in living standards and conservation of energy all across the world. However, the security requirements become much more stringent when dealing with power consumption. Smart grids cannot be successfully deployed unless these security threats have been properly addressed. Many attacks on power plants in the past indicate the indispensable need for strong security systems.

One of the most well-known attacks is Stuxnet, a 500 Kbyte computer worm which was discovered in mid-2010. This advanced piece of malware infected the software of about 14 industrial units in Iran. One of these units was a Uranium-enrichment plant. The author of the malware had enough control to damage the centrifuges without the knowledge of the operators. It had the capability to spread onto systems not even connected to the internet. [21], [22]. Another well documented attack is Shamoon also known as Disttrack that was encountered in Saudi Arabia. This destructive cyber-attack had an impact on nearly 30,000 systems targeting one of the largest oil producers. The major aim was to disrupt the functioning of various computer systems causing downtime at the targeted company [23].

These malwares not only demonstrate that the security of critical infrastructures can be breached but also the impact that these attacks can lead to. The smart grid security is thus gaining the interest of the research community. However, majority of the work been done is dedicated to the HAN domain and limited work has been presented so far on the NAN domain especially for the authentication mechanisms tailored to meet the requirements of NAN architecture. As authentication is one of the basic requirements and most critical requirements of the communication systems, in this thesis, we focus on the authentication mechanisms of NAN domain of smart grid.

1.4 Contribution

In the thesis, we present an advanced key refreshment scheme and a novel authentication scheme for the NAN domain of smart grid. The communication protocols that have been considered most optimum in the literature for the NAN are IEEE 802.11s and IEEE 802.15.4g and IEEE 802.15.4e. These protocols adapt best to the NAN architecture and security requirements. However, no protocol specifically tailored for NAN operations has been officially proposed. This work first explores the vulnerabilities of the IEEE 802.11s protocol which has been modified to be used in the NAN systems, in chapter 3. Then, a new authentication protocol has been proposed in chapter 4, which takes the advantages of the recent development of the software defined networks (SDNs) to meet the security requirements of the NAN in smart grid rather than modifying the already existing communication protocols.

The contributions in this thesis can be summarized as follows. (1) An advancement in key refreshment scheme for IEEE 802.11s based mesh authentication scheme, countermeasure to prevent the DoS attack caused by the previous key refreshment scheme, a novel authentication scheme based on dynamic one-way accumulator and SDN using zero-knowledge proofs has been proposed to mutually authenticate the smart meters and the HAN and NAN gateways. (2) The key refreshment scheme enhances the security and reliability of the IEE 802.11s based mesh networks by updating the keys in the sub-protocol Mesh Key Holder Security Handshake (MKHSH) in Efficient Mesh Security Authentication (EMSA) multiple times during one master key session. (3) The MKHSH protocol was prone to DoS attacks while using the previous key refreshment strategy, which has been eradicated

using a simple hash function implementation to cause minimal computation cost. (4) The authentication scheme employs a dynamic one-way accumulators based on the strong RSA assumption as the cryptographic tool combined with zero-knowledge proof protocols to generate an efficient and reliable authentication scheme while having minimal computation cost for the resource limited NAN network devices. The Software Defined Network architecture has been used as the backbone of the scheme to make the scheme comply with the standards of the upcoming technologies and to make the scheme more flexible. (5) All the schemes have been logically derived using the Protocol Composition Logic (PCL) and have been verified formally using the Process Analysis Toolkit (PAT) to verify the reliability of the proposed schemes. Efficiency analysis and simulation results for the schemes have been provided and compared to analyze the efficacy of the proposed schemes compared to previous schemes proposed in the literature. Hence a good balance of the system security and performance have been achieved while considering the resource limitations of the NAN domain elements.

1.5 Organization

The remainder of the thesis is organized as follows. In Chapter 2, the various solutions presented in the literature to counteract the potential security threats to the NAN domain of smart grids have been reviewed and have been presented corresponding to the various security goals to be followed by smart grid architecture, as mentioned earlier. In Chapter 3, an advanced key refreshment scheme to improve the reliability and efficiency of NAN security has been presented in detail. In Chapter 4, the authentication scheme for NAN based on SDN using one-way accumulators has been presented in detail. Finally, the conclusion and future work have been presented in Chapter 5.

Chapter 2. Literature Review

In this section we review some of the proposed solutions in the recent literature to counteract the security issues in NAN as indicated in the previous section. However, some of the work in the literature has a general approach towards these security measures, rather than having specific orientation to NAN domain. Some of these solutions have also been reviewed in this section. We first review the security solutions for authentication and then the security solutions for the rest of the security goals.

2.1 Solutions for Authenticity

Seung-Hyun Seo et al. [24] introduces a certificate-less key management approach for end-to-end security in AMI. In their approach the utility supports PKI and has its own public key certificate but for generating and managing keys for smart meters, certificate-less public key cryptography is used. Half of the user's private key is generated by a key generating center (KGC) and the other half by the user. Thus even a compromised KGC will not be able to access user's private key.

Fangming Zhao et al. [25] introduce a key exchange and revocation scheme while utilizing broadcast encryption cryptographic protocol and using a media key block. Two sub protocols are employed. In first protocol the unrevoked devices share a media key which is encrypted from a broadcast encryption using their device keys. This protocol is mandatory. The confirmation phase is optional for advanced security. Using the randomly generated number, device A calculates a parameter along with its leaf number and open ID and device B does the same. With their public keys they verify the corresponding signatures and confirms if the open ID is surely of the intended device.

Sangji Lee et al. [26] provide a coupled ID based authentication and PKI mechanism for mutual device authentication. A certificate authority is employed to provide authorization to meter data management system for certificate issuance to smart meters. SA and SM perform mutual authentication using the issued certificates. After verification new certificates are generated and authentication reply with the new certificate is sent to SM. Similar procedure is followed by SA. Now SA and SM authenticate using their own certificate encrypted by partner's public key which they decrypt using their own private key and thus perform mutual authentication.

Nian Liu et al. [27] introduced a key management framework for hybrid transmission nodes. The framework is based on key graph supporting a large number of smart meters. They aim to make a more common key management system. Another issue dealt with is the limited computational powers of the equipment. Symmetric cryptographic algorithms are used. A session key is generated using metering data, metering date, value count and user key. The information to be sent is encrypted using this key. The data is transmitted after attaching a signature. Similarly, receiver generates its own session key and with this key the signature is verified.

Rongxing Lu et al. [28] propose an efficient aggregate authentication protocol (EATH). In their protocol, all the smart grid sensor nodes aggregate their collected data and send it to the control center once they receive a request from it for state estimation. First the public and private keys of the control center and the sensor nodes are calculated and two secure cryptographic hash functions are selected. Each sensor node collects their sensed data and computes its hash value. Then attributing a signature

to it using their private keys, the data is sent to the aggregation node which is randomly chosen each time. The aggregation node verifies the data received from each node and then aggregates all the signatures to one aggregated signature. This collected data along with the aggregated signature is sent to the control center which again verifies it.

The use of SDN based authentication schemes is also gaining recent interest. Xiaoyu and Xianbin [29] proposed an authentication handover scheme for 5G hetnets using the SDN architecture. They employ an authentication module at root controller and use user dependent secure context information for authentication handover. However, the system model followed by [29] has the authentication management uses only one centralized controller, thereby making it a critical target point for the network security. Also the security of the context information has not been proved.

Hamid and Bin Hu [30], [55] present a protection scheme for 4-way handshaking protocol and a dynamically updating key distribution scheme for wireless mesh networks (WMN), which are considered for deployment in NAN. They suggest that the newly adopted standard IEEE 802.11s for WMN has Simultaneous Authentication of Equals (SAE) protocol for security but since is based on single shared password it is vulnerable. Another protocol EMSA (Efficient Mesh Security Association) which uses 4-way handshaking protocol is a good replacement for SAE. But in this protocol the unprotected message1 and message3 can be forged to cause denial of service attack. They propose a hash based encryption scheme for the protection of these messages. They assume a multi-gate mesh network, having master gateway as the Mesh Authenticator (MA). After authentication with MA, each

supplement gateway initiates 4-way handshake, to obtain PTK (Pair wise Transit Key). Since message1 from authenticator to the supplement is not encrypted, forging it will cause generation of new PTK which is not consistent with the original and hence will halt the process. Similarly, forged message3 will lead to mismatch of RSNE (Robust Security Network Element). To avoid this issue, one way hashing scheme is proposed. MA uses this hash function to encrypt PMK and insert it in the message1. Similar process is followed for message3. Through key refreshment scheme the PTK, GTK and PMK are also periodically updated.

Summary: Most of the work proposed for authentication uses complex cryptographic operations without taking into consideration the resource limitation of the HAN and NAN devices. The work in [25] and [28] use multiple cryptographic encryption and decryption operations. In [26] the scheme relies on a secure third party and also has the issue of key management. The work in [29] has a novel approach but does not provide the proof of the security of the context information being transferred. The work in [30] improves the reliability of the system but key refreshment but leads to vulnerabilities such as DoS attacks in the consecutive key refreshment cycles.

2.2 Solutions for Other Security Goals

In this section, the work related to the rest of the security goals has been reviewed.

2.2.1 Solutions for Privacy

In the deployment of smart grids, the major concern about privacy requirements lies at the user end. Customers' lack of trust towards the safety of their private information revealing their habits or

daily consumption patterns bring this concern to light. Threat to the customer's privacy does not affect the system in a critical manner but the user acceptance towards the deployment of smart meters makes it a primary concern.

Min Lu et al. [31] present 'Practical Privacy Preserving Aggregation (PPPA) scheme', where the differential privacy technique and cryptographic techniques are jointly used. The goal of differential privacy is to provide the information as a whole, without disclosing individual details. This notion is tailored for the smart grid design. The user nodes are arranged into a quad tree structure with users representing the leaf nodes. The next level in the tree corresponds to a set of four users and so on. The root node's sum estimate gives the electricity consumption of the user area. If one of the users fails to respond, the control center calculates the estimate of all yellow blocks, as shown in Figure 3 and forwards it.

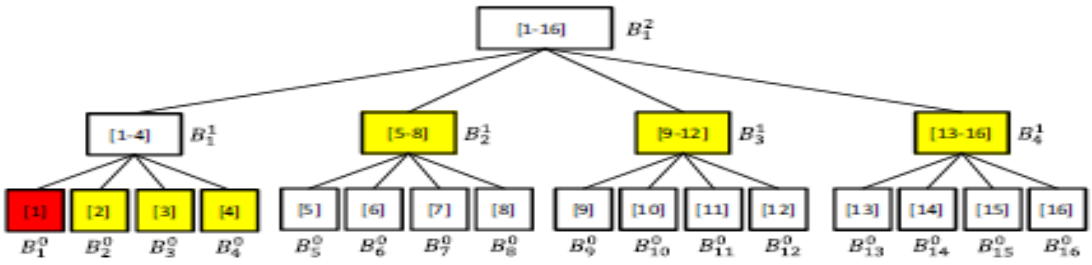

Figure3. A special case of the proposed PPPA scheme [31]

Andrew et al. [32] provide a privacy preserving scheme for Wireless Mesh Networks based on IEEE 802.11s standard, using data obfuscation technique. Elliptic Curve Cryptography is used to reduce the overhead. Using an asymmetric technique also helps to provide non-repudiation. Their approach is implemented in two phases. In the first phase, the gateway creates the obfuscation vector.

Then by using the corresponding smart meter public key, the elements of the obfuscation vector is encrypted and sent to each gateway. In the second phase, the smart meter calculates the obfuscated power reading using the received vectors, which are time stamped and digitally signed before sending to the gateway. The gateway verifies them and sends the readings to the utility.

Sook-Chin Yip et al. [33] employ an incremental hash function to conceal the consumer data while allowing the utility company to check the integrity of user data. The approach is divided into a three-step procedure: 1) the real time power consumption is converted to a corresponding cost using a quadratic cost function which is then hashed with an incremental hash function. 2) The collected hash energy cost is then summed by the operation center and then forwarded to the utility provider. 3) The utility provider verifies the integrity of the received report by comparing the total hashed energy cost received from operation center to the hashed total energy cost of the power that has been generated.

Weiwei et al. [34] introduce a new type of attack, Human–factor-aware Differential Aggregation (HDA) attack, in which human factor is taken into consideration. They have suggested that even with privacy preserving protocols applied, an attacker can infer sensitive information about an individual by monitoring the aggregation data before and after the user leaves his residence. The system model assumes n users and an aggregator. Every consumer sends the consumption information after a particular time slot. Using this information and the private key, each share is encrypted by the meter. To aggregate a specific time slot's readings, the time-series information is computed by the aggregator and using its own private key, the data is further encrypted and aggregated.

Summary: most of the proposed schemes employ homomorphic encryption or data masking as well as differential data aggregation. However, the proposed schemes have certain vulnerabilities. The scheme in [31], makes an assumption that only external adversaries are present and no internal attacks can be achieved. The incremental hashing employed in [33] has large computational expense and other innovative schemes such as [34] will induce other vulnerabilities due to human errors.

2.2.2 Solutions for Integrity

Liu et al. [35] has exposed false data injection attacks against the state estimation in power grids, which have not been considered by the existing detection schemes. Their work has considered two scenarios where the intruder has either limited resources or can attack specific meters only. Then they have proved that if the attacker has the access to the system configuration information, it can cause random errors to the state variables without being detected.

Lei Yang and Fengjun Li [36] claim that the individual smart meter data is encrypted using homomorphic encryption and then aggregated to conceal individual readings. However, the malleable property of such encryption makes it difficult to detect malicious nodes which maybe injecting false data. To tackle this issue, they have proposed a distributed outlier localization scheme based on dynamic grouping and data re-encryption. In their scheme, the data aggregation tree is partitioned into logical groups, with the root of each group storing historical data of the member meters. If the collector detects an anomaly, these root nodes are employed to find the malicious node in their sub tree. A

‘revised aggregation scheme’ is also devised so that the verifiers can access the metering data as time-series data for detection but do not violate the privacy goals.

Kebina et al. [37] introduce the use of Kalman filters to detect various system attacks including false data injection. As the attacks in the power system are reflected in the form of voltage, current or phase change, they derive the state space representation using the power grid voltage signal having amplitude and phase as variables. Euclidian distance metric based detector is employed for detection of complicated false data injection. Kalman filter uses the measurement vector collected from sensors and the state equation to compute the next state of the system. The chi-square detector compares a predefined threshold with these statistics generated and if the detected difference between the projected estimates and actual value is large, the system indicates a possible attack.

Jeffrey and Carlos [38] provide intrusion detection enabling the detection of zero-day and other elusive cyber-attacks without being confined to a specific platform. Their technique of ‘power finger printing’, as the name suggests, is based on fine grained measurements of a processor’s power consumption. These measurements are compared alongside the references of trusted software using signal processing and pattern recognition techniques. If the traces do not match the corresponding stored signature to a bearable tolerance, an intrusion is indicated.

Mohammad et al. [39] propose two techniques for stealth attack detection based on machine-learning approach. The first method employs statistical based anomaly detection algorithm. The historical data collected from SCADA is mapped to a low-dimension space. Then a Gaussian density

function is applied to the preprocessed data and the newly generated data is compared with historical data for anomaly detection. The second approach employs distributed Support Vector Machine (SVM) to detect the stealthy false data injection. The historical data is prepared to train the SVM. The basic approach is that if the historical data contains class labels, identifying normal and tempered data, then a classifier can be trained to detect the attacks.

Hanie and Edmond [40] suggest that although the PMUs are being highly deployed in smart grids for fast measurements, but it is highly uneconomical to have a PMU at each node. So they are replaced by state estimators, which open a window to the false data injection attacks. Hence they have proposed a decentralized scheme which utilizes 'Markov graph of bus angles'. To learn the structure of the grid, conditional covariance test (CCT) is employed. The power grid graph and the Markov graph are then compared using the DC power flow model. If any discrepancy is encountered, a false data injection is reported.

Summary: The work presented in the literature for integrity preservation is very limited. Most of the proposed solutions do not take into consideration the resource limitation of network devices of HAN and NAN domains. The process of re-encryption in [36] and employment of kalman filters [37] make a significant increase in the computational costs. Also the schemes in [38]-[40] tend to require hardware changes in the entire network which may not be feasible for the smart grid architecture.

2.2.3 Solutions for Availability

The availability of the network is jeopardized when the access to a resource is restricted. Major attacks causing this restriction include jamming, denial of service, replay attacks, flood attacks etc.

Nasim et al. [41] propose a low cost in band solution for intrusion detection specifically tailored for NAN. The proposed approach is a combination of signature based and anomaly based detection systems. After turning on, a smart meter starts discovering the neighbor nodes to connect to the NAN. After authentication the best path for sending data to the collector is searched. This best will give minimum number of hop counts in the packet's field. If the number of hop counts indicated in the sent packet's field has been smaller than the estimated minimum hop count, a wormhole attack is detected.

Yichi Zhang et al. [42] propose another distributed intrusion detection scheme. In this approach several analyzing modules employing artificial immune system (AIS) are deployed at level of smart grid network, specific to the requirements of that level. The AM at HAN has three parts: Intrusion data acquisition, AIS models for classification and result recording and awareness evaluation. Similarly, the AM of NAN consist of its own AIS model and interface among HAN IDS. The AM at each level incorporates clonal selection classification algorithms for identifying the attacker

Zhuo Lu et al. [43] suggest that most of the research on jamming resilience considers case-by-case methodology, using spread spectrum systems to ensure delivery of data. But in smart grids, to minimize the delay, worst case methodology is adopted for performance insurance. They show mathematically that comparatively large amount of traffic aids the delay performance of worst case scenario. Thus, 'transmitting adaptive camouflage traffic (TACT) is employed to minimize the delay. Since the jammer

cannot differentiate camouflage traffic from useful data, the jamming capability is wasted improving the system performance.

HongboLiu et al. [44] provide a communication subsystem that has intelligent local controller. This approach allows sufficient meter readings to be communicated for system requirement estimation under jamming attack, hence gives the system self-healing capabilities. The basic approach is to exploit all the available resources on hand.

Summary: Denial-of-service attacks are equally as crucial to the availability of communication network. Many intrusion detection schemes are proposed in the recent literature to tackle this issue and to upgrade the available countermeasures. The intrusion detection systems can be divided into three major categories: signature based- can detect only known attack patterns stored in the database. Anomaly based – detects new attacks but high rate of false positives. Specification based- combination of better haves of other two, as it can detect new attacks with low false positive rate.

2.2.4 Solutions for Authorization

Sushmita and Amiya [45] consider HAN, BAN and NAN gateways for data privacy and access control. The data collected by various HANS is forwarded to the BAN gateways and the NAN gateway further aggregate the readings from the gateway BANs and send it to the nearest substations. The aggregated data at each step is encrypted using Paillier encryption. To obtain the access control, the authors have tailored an attribute-based encryption. The data collected by substations is monitored by

remote terminal units (RTUs) that encrypt the collected data under a set of these attributes before sending it to the repository to be stored.

Qinghai Gao [46] claims that the biometric techniques can make the cyber-security systems more reliable, but the issue with biometric data is that it is not exactly reproducible. They provide two methods for protecting the fingerprint data. First method is for protecting minutiae-sparse fingerprint, where chaff minutiae points are added to the original template to get a chaffed template. This template is stored for enrollment but during authentication, original template is constructed but never stored. Hence a hacker could not identify the chaffed minutiae from real minutiae. For minutiae-rich fingerprint, a sub template is created by random selection of minutiae points from the original. This template is stored for enrollment and the original is constructed during authentication but never stored.

Summary: The work oriented to tackle the issue of efficient authorization is still very limited in the current literature. Although the work in [45] is tailored for smart grid architecture, the issue of key management and excessive cryptographic operation still lies. The work in [46] would not only demand extra equipment such a fingerprint device which will not only increase the installation cost but also the cost for recording and processing fingerprint data.

2.2.5 Solutions for Non-repudiation

Various Jaeduck et al. [47] non-repudiation taking into account the power consumption issue for cryptographic operations. Their work has made use of two secret keys a1 and a2 in a hash chain

relationship a1 = h(a2). The SM releases a2 to AMI and stores a1. Then these keys are used by SM to generate two MAC values for transmitting authenticated data to AMI. The key a1 serves for non-repudiation goal. Similarly, AMI generates two keys b1 and b2 and b1 serves for non-repudiation.

Zhifeng Xiao et al. [48] present a mutual inspection strategy to resolve the issue of non-repudiation in smart grid neighborhood network. The approach proposes the installation of two smart meters with one electric wire for connection between the subscriber and the service provider. The bill readings exchanged between them are used to calculate the difference between these readings. A threshold value is computed to accommodate unexpected losses and if the dispute does not lie within the range of this threshold value, the accountability is lost and the service is terminated.

Asadet al. [49] uses SDN and LTE based architecture for smart grid security. Some user specific attributes such as the meter number, are sent along with the sum of the past 24-hour metering data to the utility. A designated program at the utility generates the sum of the last 24-readings and compares it to the received data.

Summary: The goal of non-repudiation has also not been researched in depth for the smart grid network requirements. The work in [47] presents a novel and simple scheme however use of only two keys without any key refreshment makes the scheme very vulnerable over a long haul. The approach in [48] raises the question of increased cost for installation of double the number of smart meter to support this scheme. The work in [49] tends to incorporate the upcoming technologies such as SDN to make it more compatible to the future communication systems.

Chapter 3. Security Enhancement for Dynamic Key Refreshment in Neighborhood Area Network

The wireless mesh networks (WMN), are a suitable option for the connectivity in a sizable geographic area. Hence, the wireless mesh networks based on the standard IEEE 802.11s [50] and IEEE 802.15.4g [51] are being considered most adequate to deploy in the NAN domain [52]- [53]. IEEE 802.11s standard inherits the security framework of IEEE 802.11i, therefore its security vulnerabilities have been passed on to the IEEE 802.11s standard. Also, since WLANs have a single hop technology, the extension to the multi-hop networks tends to increase the threats of cyber-attacks. However, the WMNs complement the requirements and the features of the NANs, such as scalability, ease of accommodating new nodes to the network, lower investment expense and self-healing.

The IEEE 802.11s standard [50] employs the simultaneous authentication of equals (SAE) security protocol for successful mesh peer authentication based on a single shared password. However, a single password becomes a critical assumption as the disclosure of this password will compromise the network security. Another protocol named efficient mesh security association (EMSA) [54], which employs a key hierarchy to obtain the corresponding secure authentication, presents an alternative to the SAE. To improve the reliability of these protocols over a long haul, a periodic key refreshment strategy has been proposed in [55]. In the proposal, all the key materials starting from the master session key (MSK) are updated periodically before the key expiration to maintain the existing routes. However, this strategy leads to replay of message 1 in the mesh key holder security handshake (MKHSH) in the consecutive key refreshment cycles leading to possible Denial of Service (DoS)

attacks. An improvement to the EMSA has been presented in [56], named as mesh security association (MSA). In the modified protocol, message 1 is still unprotected, but it can ensure that the forged duplicates of message 1 cannot cause failure of the handshake. However, if the key refreshment is employed, this message eavesdropped in one cycle and replayed back in the subsequent cycles will still cause a DoS attack. Motivated by this reason, in this chapter we propose a new scheme named as security enhanced dynamic key refreshment (SEDKR). Our scheme employs a one-way hash based encryption and enhanced key refreshment to mitigate these vulnerabilities. Table 1 presents the abbreviations used in this chapter for the ease of understanding.

Table 1: List of Abbreviations

Abbreviation	**Definition**
MKHSH	Mesh Key Holder Security Handshake
MSK	Master Session Key
SEKDR	Security Enhanced Dynamic Key Refreshment
MA	Mesh Authenticator
MKD	Mesh key Distributer
PTK	Pairwise Transit Key
GTK	Group Transit Key
PMK	Pairwise Master Key
MKDK	Mesh Key Distribution Key
MA-ID,MKD-ID	MAC addresses of MA and MKD
MKDD-ID	MKD Domain ID
MPTK-KD	Mesh PTK for Key Distribution
SAE	Simultaneous Authentication of Equals
EMSA	Efficient Mesh Security Association
MIC	Message Integrity Code

3.1 System Model

The MSA allows a supplicant mesh point (MP) to establish the link security among other peer MPs using a key hierarchy without performing IEEE 802.1X authentication each time. Two key holders namely mesh authenticator (MA) and mesh key distributor (MKD) are used to implement the authentication and key hierarchy. A NAN is responsible for interconnecting the smart meters in home area networks (HANs) to a distribution access point to aggregate the data from different HANs and forward the aggregated data to its upper layers. A smart meter represents the gateway of each of the various HANs providing access to the home appliances and communicates its integrated data to the NAN access point. The aggregated data from different gateways of NANs in a geographical area will be provided to the wide area network (WAN) gateway, representing the advanced metering infrastructure (AMI) head-end. In the system shown in Figure 4, a WAN gateway implements both the functionalities of the MA and the MKD.

Each WAN gateway will have multiple NAN gateways and smart meters. The NAN gateway first acts as a supplicant MP and establishes security association with the MKD. Once the key hierarchy has been established, it assumes to have the role of a MA by completing the MKHSH. Once the NAN gateway becomes a MA, it can then authenticate the smart meters, acting as MPs, to join the MKD domain. Also once a smart meter has successfully established the key hierarchy and completed MKHSH, it can also act as a MA.

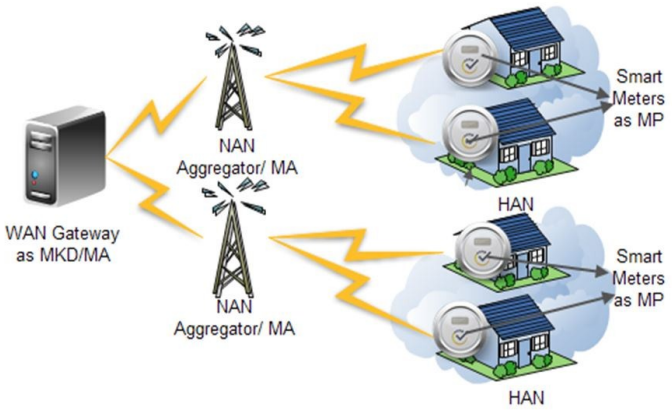

Figure 4. Implementation of MSA in NAN

The supplicant MP first executes a mesh discovery process using the beacon and probe frames that advertise the MSA capability, to detect the potential neighbors. In the initial authentication, the MP establishes a peer link with the MA. Two primitives of the Active Peer Link Open Request and the Active Peer Link Open Confirm are used to establish the link. After the successful peer link establishment, an authentication following the IEEE 802.1X standard will be employed for the establishment of the mesh key hierarchy. This procedure, also referred as the initial MSA authentication, is used only when an MP establishes a peer link in a MKD domain for the first time.

The authentication process is initiated by the MA. The messages from the supplicant MP will be relayed to the MKD by the MA using the EAP message transport protocol. If the authentication is successful, a MSK will be established between the MKD and the MP. The first level keys of the link security branch and the key distribution branch, MKD Pairwise Master Key (PMK-MKD) and Mesh Key Distribution Key (MKDK), will be derived by using the MSK, respectively. The second level key of the link security branch, MA Pairwise Master Key (PMK-MA), will be derived mutually by the

supplicant MP and the MKD. The MKD delivers the PMK-MA to the MA by using the mesh key distribution protocol. It leads to the initiation of a 4-way handshake between the supplicant MP and the MA to establish the Pairwise Transit Key (PTK) for unicast communication and Group Transit Key (GTK) for multicast communication.

Before a MP can assume the roles of a MA itself, the MKHSH has to be performed to establish the second level key, Mesh PTK for Key Distribution (MPTK-KD), of the key distribution branch. An 'aspirant MA' establishes a security association with the same MKD which derives its PMK-MKD. Thereafter, it can authenticate other supplicant MPs. The key hierarchy is shown in Figure 5.

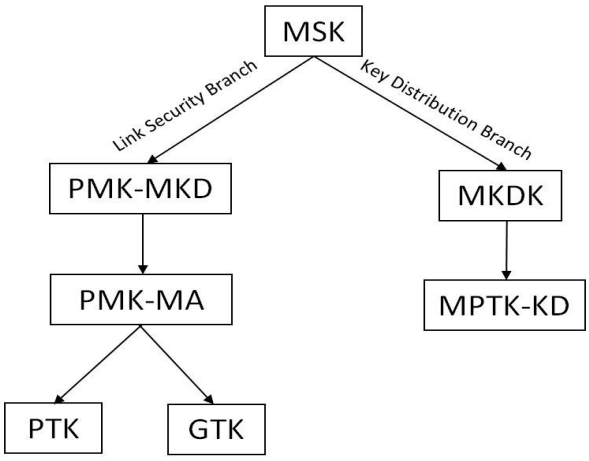

Figure 5. EMSA Key Hierarchy

Once the key hierarchy has been established, the MP can establish the authenticated peer links with other authenticated MPs using the abbreviated MSA authentication. In this case, a peer link establishment is followed by the PMK selection. It is not required to perform the IEEE 802.1X authentication again.

3.2 Vulnerability of the MKHSH Scheme

The key refreshment scheme in [55] for the EMSA protocol helps to improve the resilience of the network security over a long haul and mitigate vulnerabilities. However, this strategy can lead to message forging in the MKHSH protocol in the consecutive key refreshment cycle, leading to a DoS attack. Figure 6 shows the abstract messages that are being exchanged during the MKHSH scheme while the detailed description of the algorithm can be found in [56]. The term INFO in both the figures is being used as an alias for the collective values of MA-ID, MKD-ID, Mesh Security Capability Information Element (MSCIE) and the Mesh-ID.

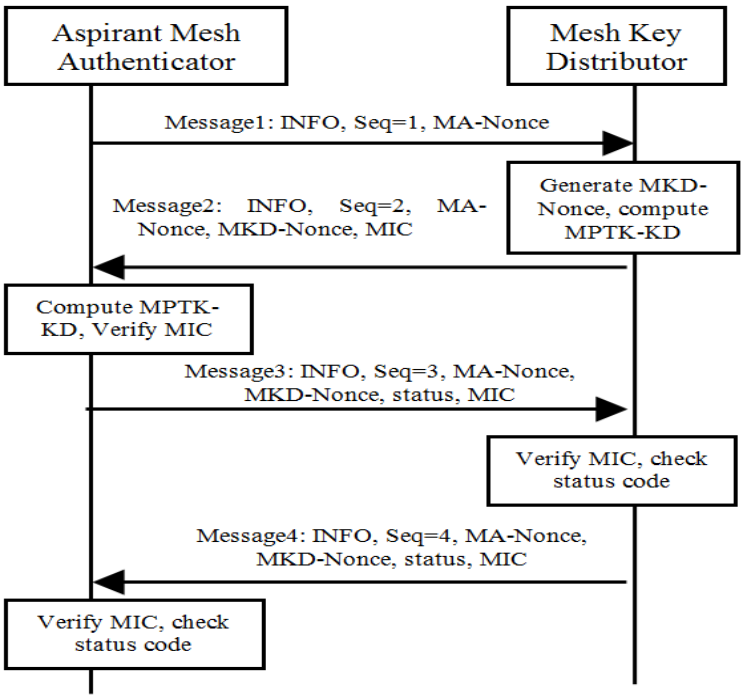

Figure 6. Mesh Key Holder Security Handshake Procedure

Note that, the first message sent from the aspirant MA to the MKD, including the above mentioned values and the MA-Nonce generated by the aspirant MA, is not encrypted and hence can be easily

tampered. Although the message cannot be forged in the standard handshake, if the key refreshment strategy is applied, this message can be replayed by the adversary in the next key refreshment cycle.

In the default key refreshment strategy [55], the process of mesh creation, as explained in section 3.1, is followed until the MKHSH and all the keys starting from first level MSK to the last level PTK and MPTK-KD keys have been established. Then, before the expiration of the master key material, the process of EAP authentication is invoked to refresh all the keys and generate new key material and thus all the procedures except the peer link establishment are repeated.

In the case of the current run of the MKHSH, if an intruder eavesdrops the message 1 and sends the forged message 1 with a new MA-Nonce to the MKD, after it has sent message 2, the MKD simply retransmits the already computed message 2 and hence the attack could not work successfully. However, during the key refreshment cycles, the adversary can eavesdrop the current message 1 and once the MP has completed the initial authentication and the 4-way handshake, the adversary can replay this forged message 1 obtained from eavesdropping. The attack scenario is depicted in Figure 7. In message 1, the fields of MA-ID and MKD-ID, depicted by INFO, would remain unaltered as the MA is establishing a security association with the same MKD. Since in the key refreshment, the peer link establishment is not repeated, thus the values of Mesh ID and the MKDD-ID element of the MSCIE, also depicted by INFO, will also not be changed. Now, the intruder can include the forged nonce as MA-Nonce**',** different from the eavesdropped message and replay the message.

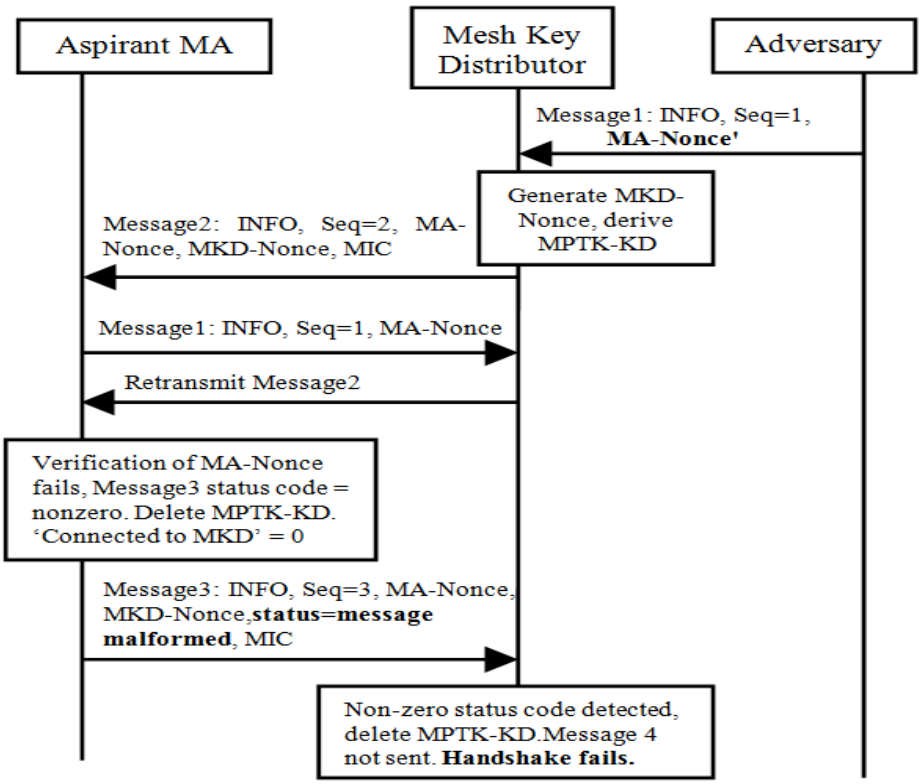

Figure 7. Mesh Key Holder Security Handshake Under DoS Attack

On receiving the forged message, the MKD verification of the values of the Mesh ID, MKDD-ID and MKD-ID of message 1 will succeed. Since the MA-ID from eavesdropped message 1 represents a legitimate node, the verification to determine if the aspirant MA is authorized to become a MA will also be successful. The MKD generates a MKD-Nonce and uses it with the forged MA-Nonce' sent by the adversary to compute the MPTK-KD and generate the message 2. Now, if the aspirant MA's message 1 with the correct new MA-Nonce reaches the MKD, after message 2 has already been sent, it will be considered as a duplicate message 1 and will retransmit message 2 without processing the correct message. Hence, the correct MA-Nonce will not be accepted by the MKD.

On receiving message 2, the aspirant MA will calculate the MPTK-KD and the MPTK-KDShortName. If these values are calculated using the original MA-Nonce generated by the aspirant MA, the verification will fail at this step only and the message will be discarded. However, if the aspirant MP computes the MPTK-KD using the forged MA-Nonce', received in message 2, then the verification of the MIC will succeed. But in the next step, the MA will ultimately verify the MA-Nonce with the one sent in message 1. Since these values will not match, the error status code in message 3 will be set to "The mesh key holder security handshake message was malformed". Consequently, the message 3 is sent with the non-zero status code, causing the MA to delete the computed MPTK-KD. Since the status code in the received message 3 would be set to non-zero, the MKD will delete the computed MPTK-KD and message 4 will not be sent. Hence, the handshake will fail and the MP will not be able to assume the functionality as an authenticator and cannot authenticate the candidate MPs. So the purpose of proposed mesh architecture cannot be fulfilled.

The strategy in [55] also tends to amplify the reliability of the authentication and association process for mesh networks by updating the key materials periodically. All the key materials starting from the MSK for the MSA will be updated before the expiration of the key materials. This is accomplished by initiating the EAP authentication and the 4-way handshaking to obtain a new set of keys. The MA can thus refresh the MSK through the EAP authentication, T_{EMSA} seconds before the expiration of the MSK lifetime, which is referred to as the MSK session. In one MSK session, multiple PTK/GTK updates can be performed by initiating the 4-way handshaking protocol to improve the network resilience against cyber-attacks. But the MPTK-KD is refreshed only once every cycle, just

as all the other keys at the higher levels of the security link and the key distribution branch. The MPTK-KD key is used for the EAP traffic transport between the mesh key holders, and also for the secure delivery and deletion of the derived keys from the MKD to the MA. If this key is compromised, the adversary can disrupt the EAP authentication or the key distribution protocols to obtain access to the network or deny access to the legitimate users. Hence, the vulnerability of this key should not be undermined.

3.3 Proposed Scheme

Since the aforementioned strategy is prone to cyber-attacks, we propose the SEDKR scheme to overcome the vulnerability of the key refreshment for the MKHSH protocol. By the SEDKR scheme, a one-way hash function is employed to protect the message 1 in the MKHSH protocol. Moreover, an enhancement to the key refreshment strategy is also proposed to further improve the resilience of the MKHSH. The supplicant MP can use one-way hash functions such as SHA-1[57] or SHA-2[58] for securing message 1. Since both of the supplicant MP and the MKD derive the MKDK mutually from the MSK, it is reasonable to assume that the key is only known to these two parties. To protect the message 1, the aspirant MA uses *MA-Nonce, MA-ID, MKD-ID* and *MKDK* as the inputs to the one-way hash function to compute a hashed value and insert it in the message 1. It is noted that the MKDK information is not being included in message 1. Since a one-way hash function has been used, it would be computationally impossible for the adversary to compute the MKDK back from this hash value. Now, if the adversary tries to replay message 1, eavesdropped from the previous key refreshment cycle, it will not be able to generate the correct hash value. The MKD will use the values of MA-Nonce, MA-

ID and MKD-ID obtained from message 1 and its own MKDK to compute the hash value and compare it to the one included in the message. Since, due to key refreshment, both of the aspirant MA and the MKD possess the new MKDKs and the hash value in message 1 sent by the adversary was computed using the old MKDK, this message will not be verified by the MKD and will be discarded. Hence, the hash value would be valid only for the current run of key refreshment and the adversary will not be able to launch this attack in any of the consecutive runs.

To increase the efficiency for verification of the authenticity of message 1 in MKHSH, the one-way hash function can be replaced by the Merkle tree implementation as used by [55]. The Merkle tree serves as a binary tree, in which non-leaf nodes are represented by the hash of concatenation of their child nodes. For example, the node Z_{12} is represented by hash (Z_1 || Z_2) and Z_1 is represented by hash(X_1) where X_1 represents MA-Nonce as shown in Figure 8. The deployment of the Merkle tree helps preventing any further replay attacks. When the MKHSH protocol is repeated, a second Merkle tree could be constructed by using the random authentication tokens and hence no Merkle tree is used repeatedly. The one-way hash function makes it impossible to retrieve these authentication tokens from the disclosed authentication path, thereby preventing the replay attacks. However, if the MKDK is not refreshed for a longer period of time, more authentication tokens will have to be generated, causing an increase of the computational cost. Hence, the balance between the security functionality and the computational cost has to be managed based on the user requirements. Figure 8 shows the Merkle Tree construction to protect message 1.

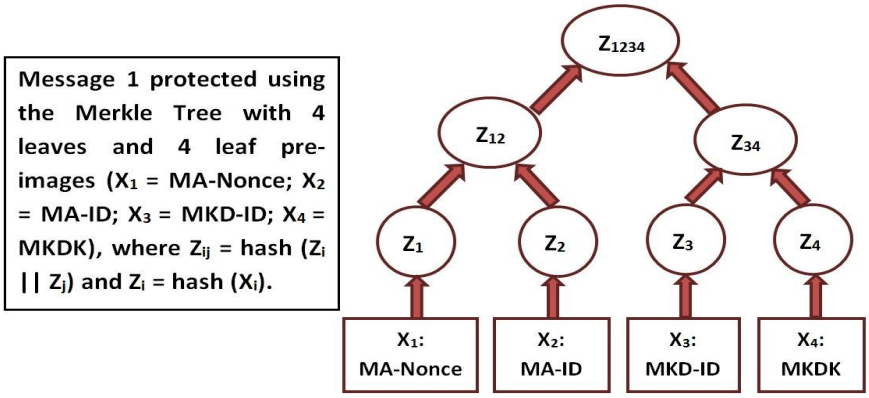

Figure 8. Merkle Tree Construction

To protect message 1, the *MA-Nonce, MA-ID, MKD-ID* and *MKDK* can be used as the leaf tokens to calculate the root of the tree, which is obtained by performing a hash operation on the concatenation of two leaf tokens by the parent node. This process is repeated until the root of the tree is derived. This root value is then sent along with the message 1. When the MKD receives the message, it uses the *MA-Nonce, MA-ID* and *MKD-ID* from the received message and its own MKDK to verify the message.

Besides, to further enhance the security, the enhanced key refreshment scheme will update the MPTK-KD multiple times in the MSK session. The updating can be accomplished by initiating the MKHSH before the expiration of the MKDK. The lifetime of the MKDK is bound to the lifetime of the MSK and thus cannot exceed it. Each time the MPTK-KD expires, the "connected to MKD" and the "Mesh Authenticator" bits will be set to zero. To regain the role of a MA, the MP reinitiates the MKHSH scheme. After the successful completion of the MKHSH, the "connected to MKD" and the "Mesh Authenticator" bits can again be set to 1. Figure 9 represents the enhanced key refreshment strategy with multiple MPTK-KD updates per MSK session.

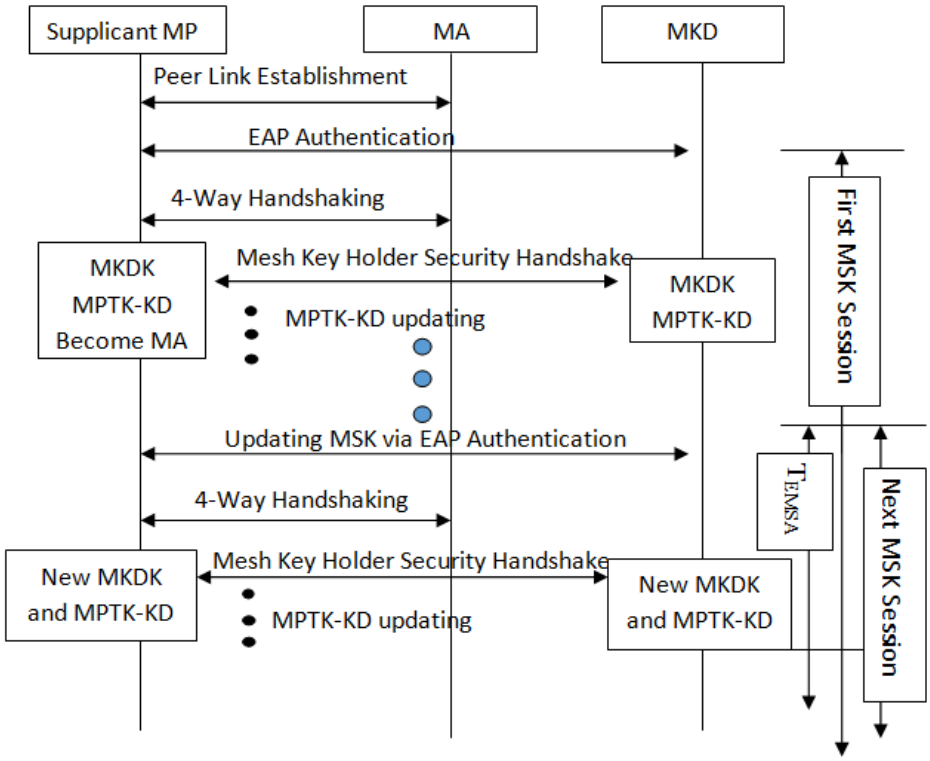

Figure 9. Enhanced Key Refreshment Strategy

3.4 Logical Analysis and Formal Verification

The logical correctness of the proposed scheme can be verified by using the Protocol Composition Logic (PCL). PCL provides a rigorous, efficient and flexible approach in proving security of network protocols. PCL is a formal methodology which is used to state the security properties of various cryptographic protocols and subsequently prove these properties. The security protocols are modelled in PCL using cord calculus. The two parties in the protocol are represented by two threads that form a cord. All the actions of the security protocols such as encryption, decryption and random number generation are performed within these threads which are bounded by pre and post conditions. Most model checking tools have a bound on the number of participants in the protocol. However, the PCL

is ready for the unbounded protocol execution possibilities. Another advantage of the PCL is that it does not only provide local reasoning for the proof components but also functions in the overall environmental conditions so that there is no interference from the other protocols using same keys and certificates. However, PCL is not optimal to prove the properties such as aliveness of the security protocols as it is most effective for signature based protocols. Since the protocols involved in this work are signature based ones, this tool is the most optimal to be used. The proof methodology of the PCL is described in [59]-[63].

The detailed proof of the standard MSA security has been presented by using the PCL in [64]. We derive the proof of the MKHSH scheme based on the work in [64] and present the required modifications in order to accommodate the security requirements for our proposed SEDKR scheme.

To further verify the effectiveness of the proposed scheme, we also perform a formal verification on our proposed scheme in the attack scenario of the MKHSH in the consecutive key refreshment cycle. The Process Analysis Toolkit (PAT) is used for the formal verification, which is the state-of-the-art model checking tool to verify the correctness of a system. This self-contained framework supports the reachability and deadlock-freeness analysis as well as the refinement checking and full linear temporal logic (LTL) model checking.

3.4.1 Logic Derivation

Based on the work in [64], Figure 10 presents the modified strands, the invariants and the security goals of the MA and the MKD. A strand is the sequence of actions that are being performed by the

designated thread. Security invariants are the security goals that hold throughout the protocol execution even if the protocol does not complete successfully, whereas the security goals hold on successful completion of the protocol only. The preconditions, the security invariants and the security goals, other than the ones mentioned in Figure 10 remain unchanged and hence have not been discussed. The aspirant MA is denoted by X and the MKD is denoted by T. The terms $INFO_X$ and $INFO_T$ include the mesh network and the security domain identifiers with a list of the supported protocols by the MKD. Select () function is used for the simultaneous selection of protocol options from $INFO_X$ and $INFO_T$ [64]. Also to be noted, the HASH component included in message 1 is a one-way HASH and not a keyed HASH computation, used for encrypting the other messages.

The strands of the aspirant MA, who is the initiator of the protocol:

MKHSH:INIT = ($X, \hat{T}, INFO_X$)

[new x; mtch $hash_0$/ HASH($x, \hat{X}, \hat{T}$, $mkdk_{X,T}$);

send "MKH1", $x, \hat{X}, \hat{T}$, $INFO_X$, $hash_0$;

rcve "MKH2", $x, t, \hat{X}, \hat{T}, INFO_T, mic_0$;

mtch $SELECT(INFO_X, INFO_T)/CS$;

mtch $HASH_{mkdkX,T}(x, t) / mptk_{X,T}$;

mtch $mic_0/HASH_{mptkX,T}$("MKH2", $x, t, \hat{X}, \hat{T}, INFO_T$);

mtch $mic_1/HASH_{mptkX,T}$("MKH3", $x, t, \hat{X}, \hat{T}, INFO_X$);

send "MKH3", $x, t, \hat{X}, \hat{T}, INFO_X, mic_1$;

rcve "MKH4", $x, t, \hat{X}, \hat{T}, INFO_T, mic_2$;

mtch $mic_2/HASH_{mptkX,T}$("MKH4", $x, t, \hat{X}, \hat{T}, INFO_T$)]$_X$

The strands of the MKD, which acts as the responder to the initiator's messages:

MKHSH:RESP = (T, $INFO_T$)

[rcve "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$;

mtch $hash_0$ / HASH(x, $\hat{X}$, $\hat{T}$, $mkdk_{X,T}$);

mtch $SELECT(INFO_X, INFO_T)$ / CS ;

new t; mtch $HASH_{mkdkX,T}(x, t)/mptk_{X,T}$;

mtch $HASH_{mptkX,T}$("MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)/mic_0 ;

send "MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0 ;

rcve "MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1 ;

mtch mic_1/$HASH_{mptkX,T}$("MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$);

mtch mic_2/$HASH_{mptkX,T}$("MKH4", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$);

send "MKH4", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, $mic_2]_T$

The preconditions to be followed by MKHSH:

$\Theta_{MKHSH,1} :=$ Has(X, $mkdk_{X,T}$) $\wedge$ Has(T, $mkdk_{X,T}$)

The security invariants for MKHSH that will hold throughout the execution of the protocol:

$\Gamma_{MKHSH,1} :=$ Honest($\hat{X}$) $\wedge$ Send(X, m)$\wedge$

(Contains(m, HASH("MKH1", ($\hat{Z}$,$\hat{Y}$)))$\vee$

Contains(m, $HASH_{mptkX,T}$("MKH2",($\hat{Y}$, $\hat{Z}$)))$\vee$

Contains(m, $HASH_{mptkX,T}$("MKH3", ($\hat{Z}$, $\hat{Y}$)))$\vee$

Contains(m, $HASH_{mptkX,T}$("MKH4", ($\hat{Y}$, $\hat{Z}$)))) $\supset \hat{Z} = \hat{X}$

The security goals of the aspirant MA:

$\Phi_{MKHSH,AUTH,MA} :=$ KOHonest($mptk_{X,T}$, {$mkdk_{X,T}$}) $\supset$

Send(*X,* “MKH1”*, x,*$\hat{X}$*,* $\hat{T}$*, INFO*$_X$*,hash*$_0$) <

Rcve(*T,* “MKH1”*, x,* $\hat{X}$*,* $\hat{T}$*, INFO*$_X$*, hash*$_0$) <

Send(*T,* “MKH2”*, x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_T$ *, mic*$_0$) <

Rcve(*X,* “MKH2”, *x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_T$*, mic*$_0$) <

Send(*X,* “MKH3”, *x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_X$*, mic*$_1$) <

Rcve(*T,* “MKH3”*, x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_X$*, mic*$_1$) <

Send(*T,* “MKH4”, *x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_T$*, mic*$_2$) <

Rcve(*X,* “MKH4”, *x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_T$*, mic*$_2$)

The key freshness security goals of MA:

$\Phi_{MKHSH,KF,MA}$:= KOHonest(*mptk*$_{X,T}$*, {mkdk*$_{X,T}$*}*) ⊃

(new ($\hat{X}$, x) ∧x⊆*mptk*$_{X,T}$∧ new ($\hat{T}$, t)∧ t⊆*mptk*$_{X,T}$)∧

FirstSend(*X, x,*“MKH1”*, x,*$\hat{X}$*,* $\hat{T}$*, INFO*$_X$*,hash*$_0$)∧

FirstSend(*T, t,*“MKH2”*, x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_T$ *, mic*$_0$) ∧

The security goals of the MKD:

$\Phi_{MKHSH,AUTH,MKD}$:= KOHonest(*mptk*$_{X,T}$*, {mkdk*$_{X,T}$*}*) ⊃

send(*X,*”MKH1”, *x,* $\hat{X}$*,* $\hat{T}$*, INFO*$_X$*, hash*$_0$) <

rcve(*T,* “MKH1”, *x,* $\hat{X}$*,* $\hat{T}$*, INFO*$_X$*, hash*$_0$) <

send(*T,* “MKH2”, *x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_T$ *, mic*$_0$) <

rcve(*X,* “MKH2”, *x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_T$*, mic*$_0$) <

send(*X,* “MKH3”*, x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_X$*, mic*$_1$) <

rcve(*T,* “MKH3”, *x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_X$*, mic*$_1$) <

send(*T,* ”MKH4”, *x, t,* $\hat{X}$*,* $\hat{T}$*, INFO*$_T$*, mic*$_2$)

Key freshness security goal of the MKD. It is the same as the initiator, i.e. the aspirant MA.

$\Phi_{MKHSH,KF,MKD} := \Phi_{MKHSH,KF,INIT}$

Figure 10. Modified Strands, Invariants and Security Goals of the MKHSH

The formal security theorem has been given below. The first two enumerations are the security goals of the aspirant MA and the MKD, respectively. The third enumeration is to verify that the protocol adheres to the invariants.

Theorem

(i) $\Gamma_{MKHSH,1} \wedge \Gamma_{MKHSH,SI,1} \vdash$

$\Theta_{MKHSH,1}[\textbf{MKHSH:INIT}]_X$

$\Phi_{MKHSH,\{AUTH,\ PTK,\ GTKD,\ KF,\ INFO\},MA}$

(ii) $\Gamma_{MKHSH,1} \wedge \Gamma_{MKHSH,SI,1} \vdash$

$\Theta_{MKHSH,1}[\textbf{MKHSH:RESP}]_T$

$\Phi_{MKHSH,\{AUTH,\ PTK,\ GTKD,\ KF,\ INFO\},MKD}$

(iii)MKHSH $\vdash \Gamma_{MKHSH,1} \wedge \Gamma_{MKHSH,SI,1}$

The enumerations (i) reads given the precondition $\Theta_{MKHSH,1}$ and the invariants $\Gamma_{MKHSH,1}$ and $\Gamma_{MKHSH,SI,1}$, once the initiator role is executed, the goal $\Phi_{MKHSH,\{AUTH,\ PTK,\ GTKD,\ KF,\ INFO\},MA}$ is guaranteed to hold. Enumeration (ii) is analogous to (i), but holds for the respondent. Enumeration (iii) states that $\Gamma_{MKHSH,1}$ and $\Gamma_{MKHSH,SI,1}$ are the security invariants of MKHSH.

By matching the conversation security goals of the aspirant MA and the MKD, we prove that the above theorem holds. Here we give only a brief walk through of the proof from the MKD's view. Full version of the proof is presented in appendix A. From the precondition $\Theta_{MKHSH,1}$, both of the aspirant MA and the MKD must have established $mkdk_{X,T}$. On receiving message MKH3, the MKD knows that the sender of this message possesses the $mptk_{X,T}$ and must have computed the MIC of MKH3. Based on the security invariants and the message identifiers, if the all parties who have access to the $mptk_{X,T}$ (i.e. X or T) are behaving honestly, then the MKD knows that the other participants in the MKHSH must be X i.e. the aspirant MA who has sent MKH3. As a participant of the MKHSH, the MKD also knows that the aspirant MA must also have verified the MIC of MKH2, which was sent earlier by the MKD, before sending MKH3. Also from the one-way hash value in the message MKH1 combined with the system invariants, the MKD can prove that this message also came from X. Thus, the MKD can be ensured that every message sent and received by the aspirant MA matches MKD's. With some other temporal tricks, the matching conversations of MKHSH can be achieved and the goal $\Phi_{MKHSH, AUTH, MA}$ can thus be established and the other security goals follow.

Similarly, in the aspirant MA's view, it receives the MKH4. Using the same approach as the MKD's view, the aspirant MA can prove that the MKD must have computed the MIC in MKH4. Hence, it knows that the MKD is a participant in the MKHSH and must have verified the MIC of MKH3 before sending MKH4 to the aspirant MA. Also from the MIC of MKH2 and using the system invariants, the aspirant MA knows that this message was also sent by the MKD. Hence, the MKD must also have computed the one-way hash value in MKH1, in order to verify it, before sending MKH2.

The aspirant MA is thus ensured that it shares the same variable with the MKD and thus can be proved that the goal $\Phi_{MKHSH,\ AUTH,\ MA}$ holds.

3.4.2 Formal Verification

By the attack models used in PAT for formal verification, the intruder is assumed to possess the contents of message 1 from the previous run of the key refreshment. In the current run, the intruder monitors the conversation of the aspirant MP until it has completed the 4-way handshake and established the PTK. Then, it sends the forged message 1 to the MKD before message 1 from the aspirant MA can reach the MKD. All the contents of the message 1 including MA-ID, MKD-ID, MSCIE and Mesh-ID, are kept the same as before. MA-Nonce field can be altered. However, the key refreshment strategy will cause the generation of a new MA-Nonce by the aspirant MA anyway. So this field could also be kept unaltered and would still yield the same result.

The execution trace as depicted in the assertion verification in Figure 11 shows that this attack becomes successful at the verification stage of the status code of message 3 by the MKD depicted by the action ‘verify_status_code_msg3’, if message 1 sent by the intruder reaches MKD first. It can also be observed that when the correct message from the aspirant MA reaches the MKD, it retransmits the same message 2 that was generated after processing the forged message 1 from the intruder depicted by ‘resend_message_2’. It can be noted that the status code field of message 3, represented by the second to last field in ‘ca.MAID.MKDID.MESHID.MKDDID.3.INONCE.MKDNONCE.1.782’, is set to 1 because the MA-Nonce verification fails in the verification of the message 2 contents. The

field ‘ca’ represents the sender aspirant MA and the rest of the fields are self-explanatory as they have been mentioned in the protocol description. As the protocol states that if the status field is set to 1 during the verification of message 3 the protocol will be terminated. Hence the MPTK-KD keys generated by both the MA and the MKD are deleted and consequently the handshake fails.

However, when we employ our proposed SEDKR scheme on message 1, the attack is no longer feasible. The one-way hash value included in the message 1 is first verified by the MKD before any other verification. The intruder does possess a hash value from the message 1 of the previous run of the SEDKR protected MKHSH. However, this information is not sufficient for the intruder to compute back the new MKDK to generate the correct hash value by forging the MA-Nonce value. The message with incorrect hash value will be discarded silently and hence the attack in any subsequent refreshment cycle will not be possible. The assertion verification in Figure 12 depicts that after employing our scheme, there is no possible execution trace that will lead to the failure of the protocol.

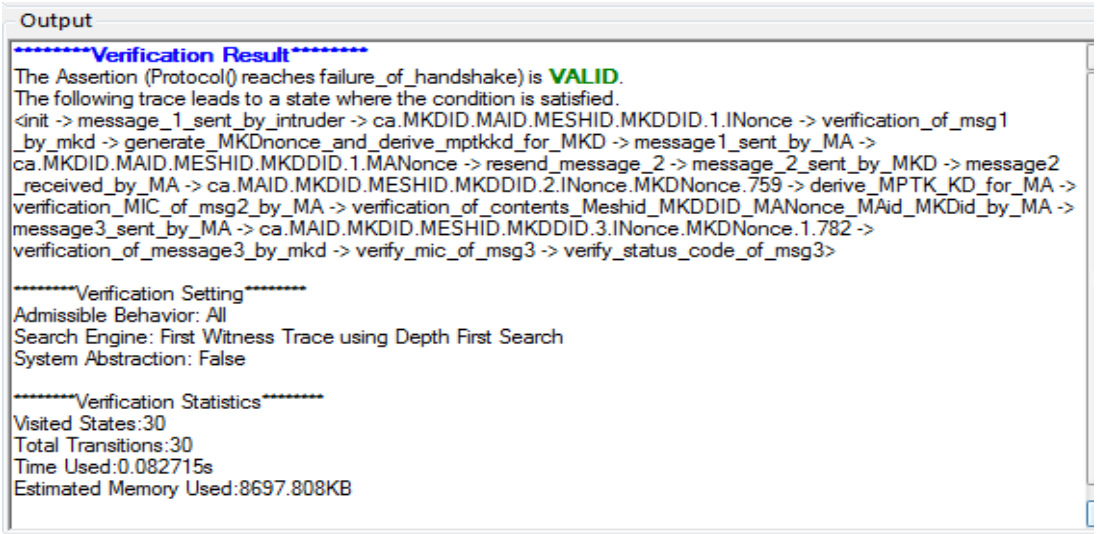

Figure 11. The DoS Attacks on the MKHSH in the Standard Key Refreshment

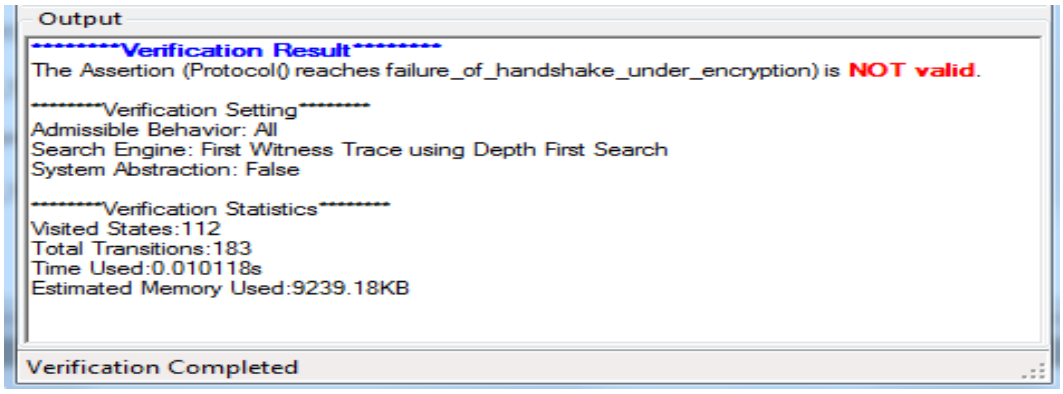

Figure 12. The Result of the DoS Attacks on the SEDKR Based MKHSH

3.5 Efficiency Analysis and Simulation Results

In this section, we perform the efficiency and performance analysis of the proposed scheme compared to the original scheme to verify that the proposed scheme can provide fruitful results while does not overburden the system resources to fulfill the required security requirements. First, we perform the efficiency analysis to compare the computation and storage costs of the proposed scheme with those of the original scheme. Then, we present the results of the simulation executions to compare the average end-to-end delay and the power consumption caused by our scheme to those of the proposed in original key refreshment scheme.

3.5.1 Efficiency Analysis

Here we evaluate the computation and storage costs of our proposed scheme. The cost for random nonce generation is considered small enough to be ignored. The computation cost in the MKHSH for both of the aspirant MA and the MKD comes out to be $3C_k+C_h+C_{oh}$, where C_k is the computation cost for message encryption using MIC, C_h is the cost for key derivation and C_{oh} is the cost introduced by

one-way hash calculation. Since C_h and C_{oh} are much less than C_k, C_{oh} does not have a significant impact on the computational cost of the aspirant MA. The MKD, however, may be involved in multiple MKHSH sessions simultaneously. So for the MKD, the cost increase is nC_{oh}, where n is the number of MPs involved in the MKHSH with the same MKD. On the other hand, the MKD has a higher computational power to accommodate multiple authentication sessions.

Assuming the storage cost for a one-way hash algorithm to be S_{OH} and S_I to be the rest of the storage cost for the MKHSH without the encryption scheme on message 1. Each aspirant MA will have an additional storage cost of S_{OH}. Even if the MKD is involved in multiple sessions, the storage cost will be increased by S_{OH} as well because the same hash algorithm is to be used with each aspirant MA.

If the Key refreshment enhancement for the MKHSH is employed, the computation cost per each MSK session will be increased to q * ($3C_k$+C_h+C_{oh}) for both the MA and the MKD, where 'q' represents the number of the MKHSH updates per MSK session. The storage cost will not be impacted, as the old key materials will be deleted every time the MKDK expires and the remaining storage information does not alter. The majority of computation and storage cost has only been increased for the MKD, while saving the resources of the MA i.e. the smart meters.

Table 2 depicts the comparison of the computation cost and the storage cost of the proposed scheme to the costs in [54], [55] and [56]. As we can observe that the proposed scheme costs slightly higher computation resource but the security provided by the new scheme is higher as message 1 of MKHSH has been protected.

Table 2: Comparison of Computation and Storage Cost for Different Schemes

Entity	[11]	[13]	SEDKR without Key Refreshment	[12]	SEDKR with Key Refreshment
Computation Cost	$2C_k+C_h$	$3C_k+C_h$	$3C_k+C_h+C_{oh}$	$q*(3C_k+C_h)$	$q*(3C_k+C_h+C_{oh})$
Storage cost	S_I	S_I	$S_I + S_{OH}$	S_I	$S_I + S_{OH}$

3.5.2 Simulation Results

In this section, we investigate the impact on the delay performance and energy consumption of the nodes implementing the MKHSH protocol when our proposed scheme has been adopted. To access the delay and energy consumption, the simulation experiment by using MATLAB has been conducted to simulate the handshake between two nodes implementing the roles of the MKD and the aspirant MA. The cryptographic operations involved in the MKHSH are the HMAC- SHA256 algorithm for calculating the MPTK-KD key, SHA-256 for calculating the MPTK-KDShortName and the AES-128-CMAC algorithm to calculate the MIC.

For calculating average delay between the two nodes, we adopted the performance benchmarks for the above algorithms, coded in C++, compiled with Microsoft visual C++ 2005 and ran on Intel Core 2, 1.83 GHz processor under windows vista in 32-bit mode. These benchmark values are tailored for MKHSH to evaluate the time expenditure for each message of the handshake.

The performance benchmark for the energy consumption analysis of the MKHSH protocol has been derived from the findings of [65]. In their experimental setup, the energy consumption of a handheld device has been computed, which is connected to a server via wireless LAN. The handheld is a Compaq iPAQH3670 clocked at 206MHz, equipped with a memory of 64MB and 16MB Flash ROM. It connects to the WLAN using a Cisco Aironet 350 series card and is powered by a Li-Polymer battery having a 950mAh rating. The energy consumption values have been computed by implementing these algorithms and evaluating the amount of current drawn from the power supply. We have tailored this energy consumption data for MKHSH and evaluated the energy consumption per node. The algorithm is run under a set of 10,000 attack scenarios with increasing rate of unknown attacks. The unknown attacks account for the interruption of the execution of the MKHSH protocol due to undiscovered attacks on the algorithm or due to other unanticipated implications.

In Figure 13, we assess the delay performance of the execution of both the unprotected MKHSH protocol and our proposed scheme, SEDKR. As it can be observed, when the execution of the MKHSH scheme is under only the known attacks, such as the DoS attacks described earlier, the proposed scheme results in the minimum average delay between the nodes. This is because by the known attacks, the unprotected MKHSH will fail only after the third message of the scheme has been processed, if message 1 was unprotected. Hence, the amount of delay incurred is fixed. This delay is even lower for the SEDKR because it does not get interrupted until the completion of the protocol. However, as the fraction of unknown attacks increases, the average delays for both the SEDKR and unprotected MKHSH increases as they can be randomly interrupted at any step of the protocol. The interrupted

protocol session is reinitiated and hence the delay overhead is getting accumulated. However, the delay performance for the SEDKR scheme still remains better than the unprotected protocol. Hence we can conclude that the proposed scheme significantly improves the delay performance of the protocol under attacks especially when we the system is under only known attacks.

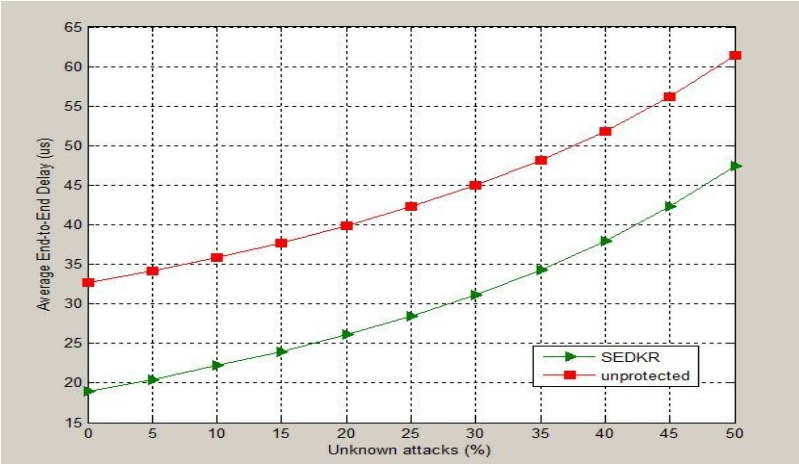

Figure 13. Average Delay during SEDKR and Unprotected MKHSH under Attacks

Similarly, in Figure 14, we evaluate the impact on energy consumption when two protocols are under both known attacks and unknown attacks. It is clear when they are under the known attacks i.e. 0% unknown attacks, the SEDKR scheme has a lower power consumption than that of the unprotected MKHSH protocol. But as the number of unknown attacks increases, the energy consumed by the nodes increases for both of the SEDKR scheme and the unprotected protocol. However, the unprotected protocol still exhibits worse performance and hence the proposed scheme helps improve power consumption even while under unknown attacks. By comparing and analyzing the results of these simulations we can observe the improvement impact of the proposed scheme on the performance of

the MKHSH protocol when it faces the security vulnerabilities caused by implementing the periodic key refreshment scheme.

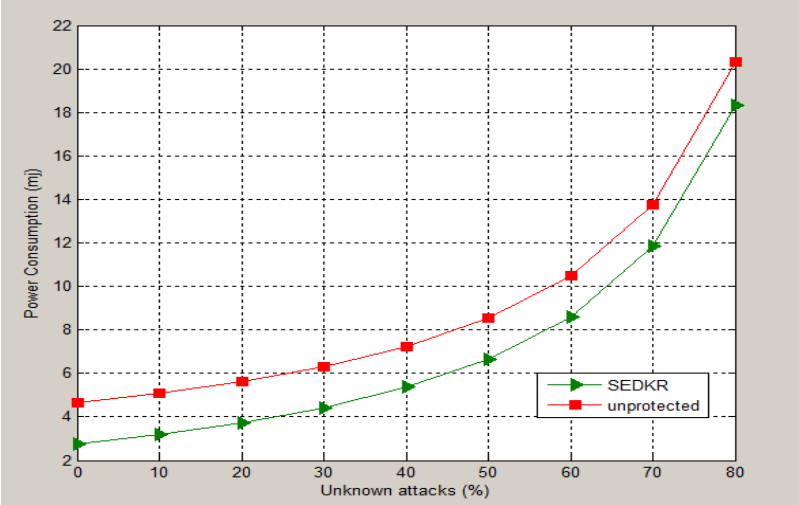

Figure 14. Average Energy Consumption during SEDKR and Unprotected MKHSH under Attacks

3.6 Summary

The resilience of the security in the NANs in smart grid has been greatly enhanced by using the existing dynamic key refreshment. But it can also make the system prone to security vulnerability in the MKHSH protocol. In this chapter, a combination of a one-way hash based protection scheme and an enhancement in the key refreshment scheme has been designed and verified to improve resilience of the MKHSH against various malicious cyber-attacks. The formal verification using PAT and PCL shows that the proposed SEDKR scheme is secure and effective in mitigating the security vulnerability in the original key refreshment scheme including the DoS attacks. The simulation results demonstrate the effectiveness of the proposed strategy under the known and the unknown attacks compared to that of the original unprotected scheme.

Chapter 4. A SDN Based Authentication Scheme for Neighborhood Area Network

Most the work focused in the recent literature for the smart grid security is based on adopting the existing communication standards and modifying them to fulfil the needs of the smart grid architecture as seen in the previous chapter. But as the current generation of wireless communication is maturing, with only incremental development, the new era of 5G communication is far under development. As the demand for higher data rates, higher data volume and the number of communication devices is increasing exponentially, eventually, the smart grid infrastructure is also to be impacted by this growth. One of the major technologies being considered for the 5G infrastructure is the Software Defined Network (SDN), which will enable making the communication networks to be programmable. [66]

Motivated by these advances, it only makes sense to incorporate these upcoming technologies while planning the implementation of secure communication infrastructure in the smart grids. Hence we intend to incorporate SDN based security architecture, so that the programmable network design helps to efficiently implement and alter the security protocols as required. As the computational ability of the smart meters and even sometimes in case of the NAN aggregators is limited, the cryptographic tools for the authentication procedure should cause minimal computational expense on these entities. Taking these factors under consideration, in this chapter, we propose a novel authentication scheme named as low cost SDN based NAN authentication (LCSNA). Our scheme employs a dynamic one-way accumulator combined with the zero-knowledge security proofs to obtain low cost and efficient authentication scheme for the NAN domain of the smart grid architecture. The computational cost is

highly reduced as the simple communication procedure and optimal revocation makes the deployment more efficient, especially for large-scale dynamic distributed networks. To the best of our knowledge, this work is the first SDN based authentication scheme for the NAN in smart grids.

The process of authentication prescribed by our scheme is very easy and convenient. We know that the most optimum network architecture for the smart grid NAN domain is the mesh network. Since our proposed scheme allows the cross authentication of NAN and HAN and the authentication process does not have to rely on a third trusted party, it is very easy and fast to perform authentication among various parties simultaneously. Hence the proposed scheme is optimum for the mesh networks. Also due to the use of dynamic one-way accumulators, the number of users present in the network does not indicate the computation and storage overhead. Moreover, since the users do not have to reveal their actual identity to prove themselves to be the network, the privacy of the user is preserved in this aspect. As the SDN architecture shifts the control logic to the centralized intelligence, any changes or revisions of the protocol to be implemented do not require altering network device requirements.

4.1 SDN Architecture and Preliminary

To understand the efficacy of the proposed scheme, we give a brief introduction of the SDN architecture and the dynamic one-way accumulator which form the core base of the scheme. We use the dynamic one-way accumulator in our scheme that is based on the strong RSA assumption but only the fundamental terms have been mentioned for ease of understanding. For detailed description [67] can be referred.

4.1.1 SDN Architecture

SDN is being considered the new radical paradigm for programmable networking. To meet the requirement of 5G infrastructure, the network management and configuration of the traditional networks, which rely on numerous network devices with complex protocols, has to evolve into much more efficient and scalable networking infrastructure. SDN promises to accomplish the same by shifting the control logic from the underlying switches and routers to a centralized controller in the control plane. By separating the control plane from the data plane, the SDN controller is able to globally control the network intelligence rendering the network devices to act merely as packet forwarding devices [68]. Although, the programmability of the network using a centralized intelligence makes the network management and configuration much more efficient, but overloading the single controller for all the frequent network events is not very optimal for the scalability of a large-scale distributed network, especially for the smart grid infrastructure.

To preserve the scalability of the SDN system without compromising the general principles of SDN, an alternative route of distributed framework, Kandoo [69] can be employed instead. Kandoo is a distributed control plane, which has two layered hierarchy of controllers: (1) In the first layer, local controllers are deployed, which execute the local applications which do not require information of the network-wide state. (2) In the second layer, the centralized root controller is used to run the control applications which are not confined to the local operations and hence maintains the network-wide state. The local controllers can control one or a small number of network devices, whereas all of the local

controllers are controlled by the root controller. Any applications to be installed by the root controller are delegated to the respective local controllers. Figure 15 shows the implementation of the Kandoo architecture.

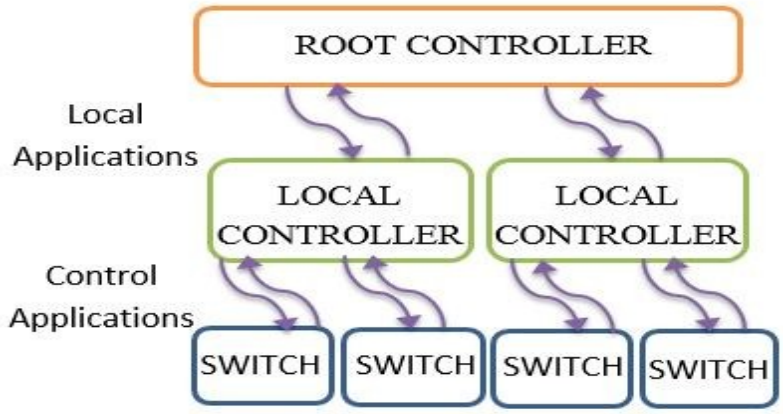

Figure 15. Kandoo: Layered SDN Structure

4.1.2 One-way Accumulator

Cryptographic accumulators were first proposed by Benaloh and de Mare [70] in 1993. One-way accumulators are cryptographic equivalent of data structures and are both space and time efficient. Hence their application is optimum for the highly distributed networks such as NAN, where they eliminate the requirement of a trusted third party. These asymmetric cryptographic accumulators create a fixed size accumulated value that represents a large set. A witness is also generated for each value of the set. The witness and the accumulated value are used together to prove the membership of the value in the set. Hence, all the values outside the given set should fail this verification. This becomes the security requirement of this model. Accumulators can act as an efficient alternative to digital signatures schemes for security algorithms and protocols.

The one-way accumulators can be defined as quasi commutative one-way hash function families.One-way hash function is of the form $H_i = \{h_j : X_j \times Y_j \rightarrow Z_j\}$, where $i \in N$. Also the hash value $h(. , .)$ is computable in polynomial time with regards to the integer i. If any probabilistic polynomial-time algorithm λ is given, then the probability to find a $xx \in X_j$, while $yy \in Y_j$ and a pair $(x, y) \in X_j \times Y_j$ (where $yy \neq y$), are already given, such that $h_j(x, y) = h_j(xx, yy)$ is less than $negl \langle i \rangle$ [70].

A function $f: X \times Y \rightarrow X$ is said to be a quasi- commutative function if it follows the following rule:

$$f(f(x, y_1), y_2) = f(f(x, y_2), y_1) \quad (1)$$

Where $x \in X$ and $(y_1, y_2) \in Y$. By using the above two properties provided by the one-way accumulator, we can calculate a small accumulation value Z, which encapsulates a large number of values belonging to a specific set say A. However, only the members of the set A can prove that their value was accumulated in Z without revealing their identity.

$$Z = h\,(h\,(h\,(x, y_1), y_2), \ldots , y_N) \quad (2)$$

Where x is the key or the seed value used for the accumulation resulting in the accumulation value Z. $y_1, \ldots, y_N$ are the members of the set A that are being accumulated. But for the members of the set to be able to verify themselves, a witness value is created to calculate another accumulation value that excludes one member of the set from the accumulation process for which the witness is being created. For example, for the member k,

$$Z_k = h(h(h(h(x, y_1), y_2), \ldots , y_{k-1}), y_{k+1}), \ldots\ldots, y_N) \quad (3)$$

If the value $h_x(y_k, Z_k) = Z$, then the verification is successful.

4.2 Proposed Scheme

In this section we provide the details of the proposed scheme and the design goals it intends to accomplish. Also the system model and the parameters and the assumptions made corresponding to the scheme have been mentioned before we present the scheme.

4.2.1 System Model

As we know the smart meters in the HAN are interconnected via the NAN aggregators to relay the information from the users to the Wide Area Network (WAN) for processing and billing purposes. So both the NAN gateways and the smart meter joining a specific NAN are to be authenticated before they can start the communication.

As shown in the Figure 16, the upper layer of the distributed control plane of Kandoo can be implemented at the utility or the AMI head-end. Hence the central root controller is employed at the utility and is provided with an authentication module (AM) for the authorization of the network users. The authentication module also stores the information about which entity can access what specific information. The NAN gateways and the HAN gateways joining the network are implemented at the local controllers. The smart meters being the network devices represent the function of the switch in the Kandoo architecture. Thus the programmability of the local controllers defines the operation of the smart meters. But for the ease of understanding, we will refer to the smart meters and the HAN gateways as users, being executed by the local controllers. The NAN gateways also act as the new users during

the initialization process of the network. Once they have been authenticated they can assume the role of verifiers and can authenticate the corresponding users.

Firstly, all the NAN gateways and the users have to establish a security association with the root controller. This step is referred to as the 'Initial Sign Up', which is accomplished by using the zero–knowledge proof of big integer being a product of two quasi-safe primes. Once the initial sign up process is complete, the users can undergo the 'Cross Authentication', to complete the user authentication. After completing the cross authentication, the Diffie-Hellman key exchange protocol can be employed, so that the users can obtain the session key and then send data to the NAN gateways. Further if required, we can also perform the key exchange among different smart meters. Such scenarios are plausible for certain cases being considered for the smart grid infrastructure; for example, users producing renewable energy at their end can sell to other users and negotiate the price.

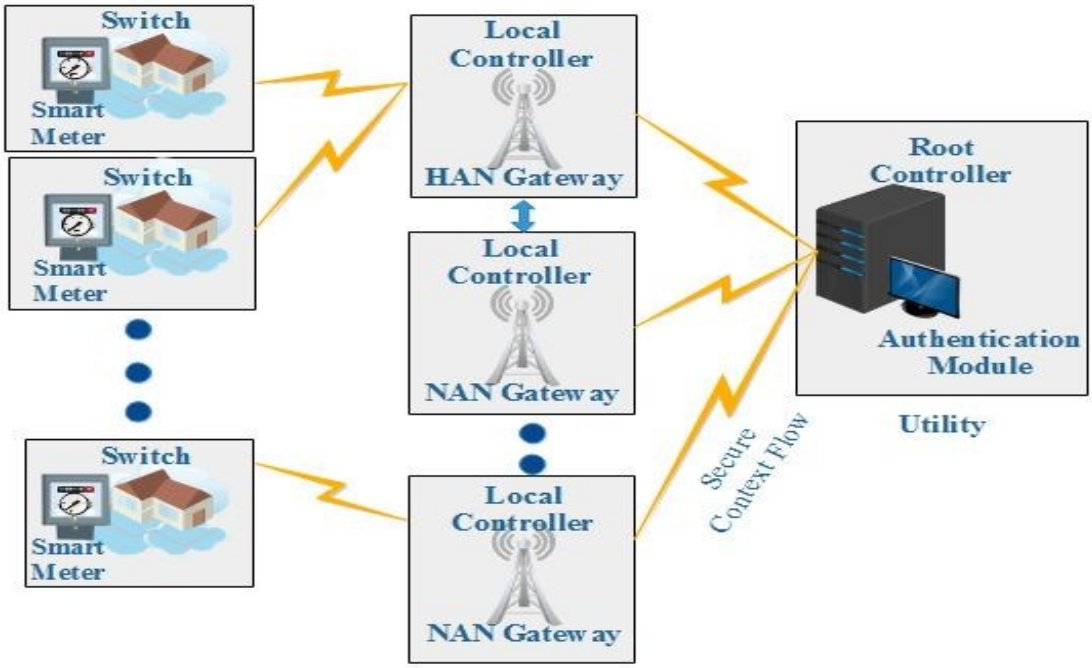

Figure 16: System Model

4.2.2 Design Goals

The design goal of our authentication scheme is to provide a cost effective and dynamic mechanism in the mesh networking environment of NAN using SDN. One-way accumulator and zero-knowledge proofs provide a simple communication process and accelerate the authentication scheme. To further enhance the security and reliability of the authentication scheme, the AM periodically performs re-authentication of NAN gateways. This operation can be performed in the off-peak times by the AM using its master key. This ensures that there is no leakage of privacy by the NAN gateways that could be compromised. If the NAN gateway is detected to be compromised, it is put in the blocked list to analyze the defect. That phase is not a concern of our scheme. As some of the authentication procedure is shifted to off-peak times, the root controller is relieved from some burden, improving the efficiency.

4.2.3 Assumptions and Parameters

We assume that the root controller is assumed to be located the utility's data center. The AM is implemented at the root controller which is responsible for authentication of the HAN gateways and the NAN gateways. No user can access the network services before it has undergone the initial sign up with the AM and the cross authentication procedures. The AM is also assumed to possess a master key pair (K_{pub}, K_{pri}), where the public key is known to all the users and the private key is only known to the AM.

The initial key space, represented by the matrix $G_{m \times m}$ is stored at all the users. Here 'm' represents the security parameter. Higher the value of m, higher would be the security and consequently higher would be the computation cost. So based on the resources available and security required, the value of

m can be opted. A function for key derivation *f(a,b)* and a lightweight encryption algorithm ENK is also stored at all the users. Using randomly generated seed value, the initial key space and the function *f(a,b),* the users can generate a one-time password and use it as an encryption key during the cross authentication phase.

The use of dynamic accumulator enables the easy addition new member and deletion of misbehaving members. If a new member 'i' is added to the network, the new node performs initialization and sends y_i , g^{pi} and g^{qi} to the AM, where accumulation value is updated as follows:

$$Z_{new} = Z^{y_i} \bmod N \qquad (4)$$

Z_{new} along-with y_i is broadcasted to all the nodes. Then the corresponding nodes update their partial accumulation values by using the relation:

$$Z_{pnew} = Z_p^{y_i} \bmod N \qquad (5)$$

If any of the users in the network are misbehaving or are suspected to be compromised, then their membership can be easily revoked by the AM. The accumulation value then can be updated as follows:

$$Z' = Z^{1/yj \bmod \varphi(N)} \bmod N \qquad (6)$$

This value along-with the ID of the deleted party is broadcasted to all the nodes. Then the nodes update their partial accumulation values as follows:

$$Z'_p = Z_p^b . Z^a \bmod N \qquad (7)$$

where the values '*a*' and '*b*' are obtained by Euclidean algorithm using the following relation:

$$a . y_i + b . y_j = 1 \qquad (8)$$

where y_j is the member being deleted and y_i is the node updating its partial accumulation value.

4.2.4 Detalied Scheme

As the major novel consideration in this paper, use of dynamic one-way accumulator combined with the zero-knowledge proofs are the soul of the authentication scheme. The authentication mechanism has two main functions: signing up the users in a NAN to the AM i.e. the members constituted in the accumulator and the authentication of the members among themselves to replicate the mesh networking architecture. Whenever a new NAN is to be set up, all the members need to be authenticated by the AM and the users need to be authenticated by NAN to set up a communication session. The integer values required to set up the accumulator are sent to the AM to verify that each value is unique and confines to the strong RSA assumptions. Zero-knowledge proof ensures the legitimacy of the users. After that phase, the members of the accumulator can authenticate each other and set up the secure communication. Using the initial key space, randomly generated values r, s and t and the function *f(a,b),* the prover generates a key K_λ, where 0<s, t<m. The encryption key generated is used by the lightweight encryption algorithm ENK to encrypt the corresponding messages of the zero-knowledge proof to accomplish the cross authentication. The session keys and the initial key space are updated frequently and the level of the security can be controlled by this frequency at the expense of the computational cost.

We describe the proposed scheme LCSNA in the following. The scheme is presented in three phases: key pre-distribution phase, initial sign up phase and cross authentication phase.

- **Key Pre-Distribution Phase**

During the key pre-distribution phase, the Users and the Authentication Module in the root controller take the following steps.

- Root Controller:

1. Authentication Module generates a big rigid integer $N = (p_{am} . q_{am})$. Here p_{am} and q_{am} are distinct safe primes. According to the definition of a rigid integer: $p_{am} = 2p' + 1$ and $q_{am} = 2q' + 1$ and p_{am}, q_{am}, p' and q' are all odd primes.
2. AM also selects x (seed value), such that $x \in QR_n$ and $x \neq 1$.
3. AM chooses a group $G \in Z_N^*$, with g as the generator for the group.

- Users:

1. The user node selects a prime integer $y = p . q$, such that it is relatively prime to p' and q'.
2. The following condition must be satisfied:

   $y = p . q \ \wedge p, q \in$ primes $\wedge p, q \neq 1 \bmod 8 \wedge p \neq q \bmod 8$. The restriction of the above condition is to be followed because the zero knowledge proof protocol used for the authentication [71] demands these conditions to be satisfied.

- **Initial Sign Up Phase**

In this process, the new users including the NAN gateways and the HAN gateways corresponding to these specific NAN get signed up with the AM.

Message1: The user *i* sends $y_i = p_i \,.\, q_i$, g^{pi} and g^{qi} to the AM. Along with these values, message 1 also includes a replay counter CNT_U.

Message2: when the AM receives the message 1, it checks the replay counter. Then it verifies that g^{pi} and g^{qi} coming from different users are not repeated. In case repeated values are found, the AM demands those users to select and send these values again. If no repeated values are found then the AM verifies $gcd\ (y_i, \varphi(N)) = 1$. If this is not satisfied, the user is to select and the values again. Once both the verifications are successful, the AM randomly chooses value $w_j \in Z^*_{yi}$ and publishes this value to the user *i* in message 2 along with the replay counter CNT_{AM}.

Message3: upon receiving the message from the AM the user performs following operations:

a) Checks the replay counter and computes a value *M* that is the inverse of $odd\ (N(N-1))^{-1}\ mod\ \varphi(n)$.
b) Computes $y = w_j^M\ mod\ N$
c) Selects a value $c \in \langle w_j \rangle$ and Computes $d = log_{wj}\, c$
d) Computes r_1 = square root of $\pm w_j$ or $\pm 2w_j\ mod\ N$ and r_2 = square root of $\pm d$ or $\pm d/2\ mod\ odd\ (\varphi(N))$.
e) Finally outputs the values: y, r_1, r_2 and CNT_U.

When the AM receives the message3 from the user it performs the following verifications:

a) Checks the replay counter.
b) If the value $y^{\,odd(N\,(N\,-\,1))} = x\ mod\ N$.
c) Verifies if r_1^2 is congruent to $\pm w_j$ or $\pm 2w_j$

d) Selects a value $e=2^{|N|}$ and verifies if $c^e \bmod N = w_j^{\pm 2er_2^2} \bmod N$ or $w_j^{\pm er_2^2} \bmod N$.

If all of the above verifications are successful then the signing up process has been completed and the user can move onto the cross authentication phase. But if any of the verifications fails, the user is simply discarded. Figure 17 shows the messages being exchanged during this process. The initial sign-up stage follows the zero-knowledge proof based on [71]. Once the verifications are complete the AM can compute the accumulation value $Z = x^{\sum_{i=0}^{m} y_i} mod\ N$ and the partial accumulation value $Z_p = x^{\sum_{i \neq p, i-1}^{m} y_i} mod\ N$. The partial accumulation is sent to all the corresponding nodes and the accumulation value is broadcasted to every node.

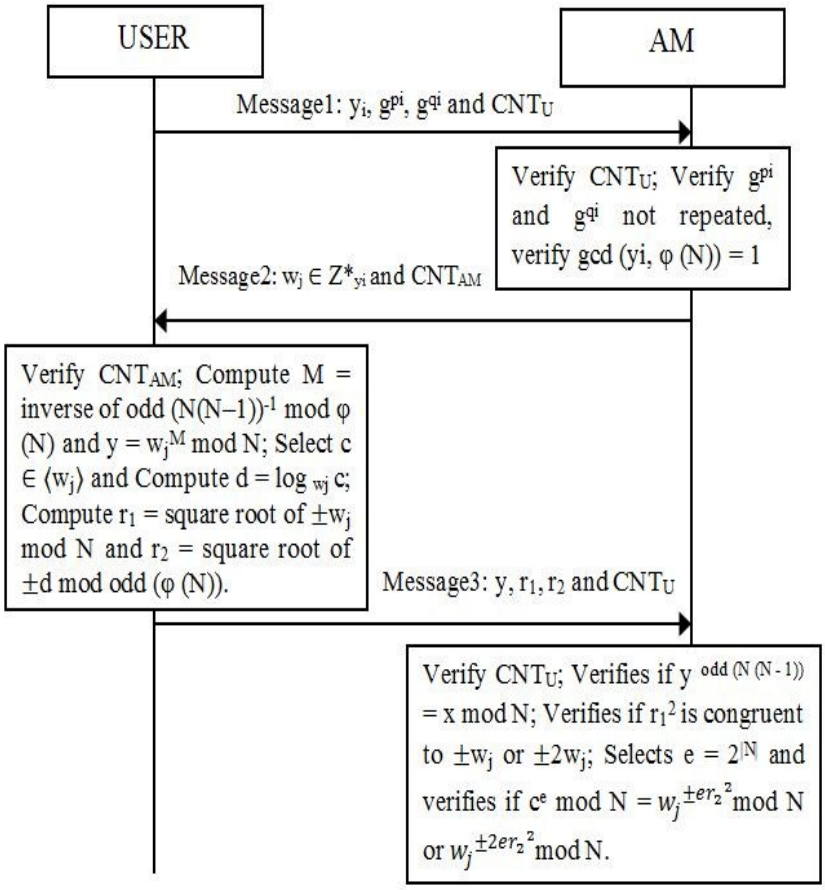

Figure 17: Initial Sign Up

- **Cross Authentication Phase**

In this stage, the signed up users authenticate each other. First the NAN gateways can authenticate the HAN gateways and once they have been authenticated, the different HAN users can authenticate each other. The two authenticating parties are named as prover and the authenticator. The process of authentication is shown below:

Message1: The prover generates three random values *r, s* and *t* and then calculates a key $K_\lambda = f(r, G_{s\times t})$. It then uses this key with ENK to encrypt message1 containing the partial accumulation value Z_p along with y_p and the values *r, s, t* and a message sequence *seq*=1 and then it sends it to the authenticator.

Message 2: When the authenticator receives the message 1, it calculates the key K_λ using *r, s, t* and $G_{m\times m.}$ It decrypts the message using ENK and checks the sequence. Then it verifies that the $Z = Z_p^{yp}$. If the verification fails, the authentication process is terminated here. Otherwise the authenticator choses a random value $h \in Z^*_{yp}$ and a big number l_a. The authenticator calculates the values $k = h^2 \ mod \ y_p$ and g^{la} and send both these values to the prover. The *seq* values is set to 2. The message is encrypted with ENK using $K_{\lambda.}$ If the authenticator receives the message1 again, the message is not processed but simply the message2 is transmitted again.

Message3: The prover first decrypts the message with ENK using K_λ and checks the sequence. After receiving the values *k* and g^{la}, the prover calculates the square root of *k* such that $S = \sqrt{\mathrm{k}} \ mod \ y_p$ and randomly choses a value $j \in Z^*_{yp}$. Then it calculates $c = j^2 \ mod \ y_p$ and $d = S^{g^{la}} \ j \ mod \ y_p$ and sends *c* and

d to the authenticator. The value of *seq* is set to 3. The message is encrypted with ENK using K_λ. If message2 is received again, it is not processed but the message3 is transmitted again to the authenticator.

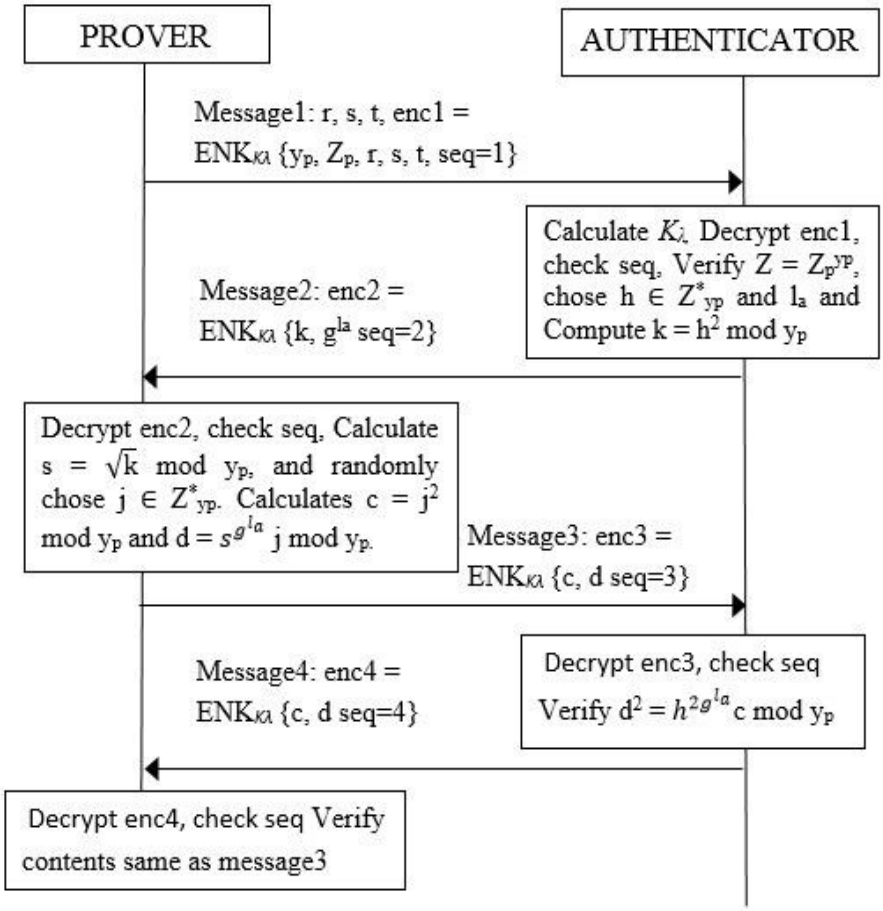

Figure 18: Cross Authentication

Message4: On receiving the message 3, the authenticator decrypts the message with ENK using K_λ and checks the *seq* value and verifies if $d^2 = h^{2g^{la}} c \ mod \ y_p$, If this verification is successful, the authenticator knows that the prover knows prime factorization of y_p and hence the prover is authenticated by the authenticator. Message4 is sent just as an acknowledgement of message3. The contents of message3 are encrypted with ENK and send to the prover with *seq* = 4. If message 3 is

received again after authentication is complete, message4 is simply retransmitted. The prover on receiving message4 verifies that the contents are similar to the ones sent in message3 after decrypting the message. Similar procedure can be applied for mutual authentication where the prover becomes the authenticator and vice-versa.

The cross authentication process follows the zero-knowledge proof mentioned in [72]. Once both parties are authenticated, they can obtain the session key g^{lalp} and can start the secure communication, where l_p is the random big number chosen by the prover. Figure 18 shows the process of cross authentication.

4.3 Security Analysis and Formal Verification

The logical correctness of the proposed scheme can be verified by using the Protocol Composition Logic (PCL). PCL provides a rigorous, efficient and flexible approach in proving security of network protocols. The proof methodology of the PCL is described in [59]-[63]. To prove the effectiveness of the proposed scheme, we also perform a formal verification of the different attack scenarios in our proposed authentication scheme by using the Process Analysis Toolkit (PAT).

4.3.1 Logic Derivation

Figure 19 presents the strands, the invariants, the preconditions and the security goals of the prover and authenticator in the cross authentication stage. Similar procedure can be followed for the user and the authentication module in the initial sign up stage. A strand represents the sequence of actions that are being performed by the designated thread. Security invariants are the security goals that hold

throughout the protocol execution even if the protocol does not complete successfully, whereas the security goals hold on successful completion of the protocol only. We omit the description for the initial sign up stage and the process of the proof due to space restriction. The User and the authentication module in the initial sign up are denoted by Y and Z respectively. While the prover and the authenticator in the cross authentication stage is denoted by X and T respectively.

The following shows the strands of the prover, who is the initiator of the protocol. These strands explain the process of messages being created and transferred during the cross authentication from the prover's view. The first line depicts that the prover selects the values r, s and t and use them to calculate the key K_{λ}. The second line depicts the encryption of the contents of message1 and the next line represents the creation and transfer of message1. Similar procedure can be followed for the rest of the strands.

MKHSH:INIT = (X, $\hat{T}$)

[new r, s, t; mtch HASH(r, $G_{s\times t}$) / K_{λ};

mtch enc_0/ $HASH_{K\lambda}$ (Z_p, y_p ,r,s,t, $\hat{X}$, $\hat{T}$, seq);

send "MSG1", r, s, t, $\hat{X}$, $\hat{T}$, enc_0 ;

rcve "MSG2", $\hat{X}$, $\hat{T}$, enc_1 ;

mtch HASH(r, $G_{s\times t}$) / K_{λ};

mtch enc_1/$HASH_{K\lambda}$ ("MSG2", $\hat{X}$, $\hat{T}$);

mtch enc_2/$HASH_{K\lambda}$ (c, d, seq, $\hat{X}$, $\hat{T}$);

send "MSG3", $\hat{X}$, $\hat{T}$, enc_2;

rcve "MSG4", $\hat{X}$, $\hat{T}$, enc_3;

mtch enc$_3$/HASH$_{K\lambda}$ (“MSG4”, $\hat{X}$, $\hat{T}$)]$_X$

The following shows the strands of the authenticator, which acts as the responder to the initiator’s messages. These strands explain the process of messages being created and transferred during the cross authentication from the authenticator’s view. Similar explanation as explained for the prover can be applied for the authenticator’s strands also.

MKHSH:RESP = (T)

[rcve “MSG1”, r, s, t, $\hat{X}$, $\hat{T}$, enc_0;

mtch HASH(r, $G_{s\times t}$) / K$_\lambda$;

mtch enc$_0$ / HASH$_{K\lambda}$ (“MSG1”, r, s, t, $\hat{X}$, $\hat{T}$);

mtch HASH$_{K\lambda}$ (k, g^{la} , seq, $\hat{X}$, $\hat{T}$)/enc_1 ;

send “MSG2”, $\hat{X}$, $\hat{T}$, enc_1 ;

rcve “MSG3”, $\hat{X}$, $\hat{T}$, enc_2 ;

mtch enc_2/HASH$_{K\lambda}$ (“MSG3”, $\hat{X}$, $\hat{T}$);

mtch enc_3/HASH$_{K\lambda}$ (c, d, seq, $\hat{X}$, $\hat{T}$);

send “MSG4”, $\hat{X}$, $\hat{T}$, enc_3]$_T$

The following shows the preconditions to be followed during the Cross Authentication process. The preconditions states that only the prover and the authenticator, which are represented by X and T have aces to the lightweight encryption algorithm.

Θ_{CA}:= Has(X, ENK) ∧ Has(T, ENK)

The following depicts the security invariants for cross authentication that will hold throughout the execution of the protocol. The security invariants are the goals that must uphold even if the protocol does not terminate properly. The first security invariant represents the order in which the messages are being transferred between the prover and the authenticator. The second invariant depicts that only the prover and the authenticator are able to get hold of the encryption key during the cross authentication. Both cases assume that the prover is an honest party.

$\Gamma_{CA,1}$:= Honest($\hat{X}$) ∧ Send(X, m)∧

(Contains(m, $HASH_{K\lambda}$ (“MSG1”, ($\hat{Z}$,$\hat{T}$)))∨

Contains(m, $HASH_{K\lambda}$ (“MSG2”,($\hat{T}$, $\hat{Z}$)))∨

Contains(m, $HASH_{K\lambda}$ (“MSG3”, ($\hat{Z}$, $\hat{T}$)))∨

Contains(m, $HASH_{K\lambda}$ (“MSG4”, ($\hat{T}$, $\hat{Z}$)))) ⊃$\hat{Z}$= $\hat{X}$

$\Gamma_{CA,2}$:=

KOHonest(K_{λ},{ENK})⊃

Has(Z, K_{λ}) ⊃ $\hat{Z} = \hat{X} \vee \hat{Z} = \hat{T}$

The security goals of the prover are presented follow. The security goals are to be held upon on the completion of the protocol. The first shows the authentication completion goal from the prover’s view. It shows the encrypted messages being exchanged and their corresponding responses in the corresponding order. The respective party responds only if receives the expected message.

$\Phi_{CA,AUTH,PROVER}$:=

KOHonest(K_{λ},{ENK}) ⊃

Send(X, "MSG1",$r, s, t, \widehat{X}, \widehat{T}, enc_0$) <

Rcve(T, "MSG1", $r, s, t, \widehat{X}, \widehat{T}, enc_0$) <

Send(T, "MSG2", $\widehat{X}, \widehat{T}, enc_1$) <

Rcve(X, "MSG2", $\widehat{X}, \widehat{T}, enc_1$) <

Send(X, "MSG3", $\widehat{X}, \widehat{T}, enc_2$) <

Rcve(T, "MSG3", $\widehat{X}, \widehat{T}, enc_2$) <

Send(T, "MSG4", $\widehat{X}, \widehat{T}, enc_3$) <

Rcve(X, "MSG4", $\widehat{X}, \widehat{T}, enc_3$)

The following presents the key freshness security goals of the prover. It depicts that the freshness of the encryption key is maintained if the values r, s and t are generated by the prover for the first time and the authenticator receives them the first time in the contents of message1 only. The values received otherwise are discarded.

$\Phi_{CA,KF,PROVER}$:=

KOHonest(K_λ,{ENK}) ⊃

(new ($\widehat{X}, r$), new $(\widehat{X}, s)$, new ($\widehat{X}, t$) ∧ $r, s, t \subseteq K_\lambda$)∧

FirstSend(X, r, s, t,"MSG1", $r, s, t, \widehat{X}, \widehat{T}, enc_0$)

The security goal for key security by the prover represents that both the prover and the authenticator have the access to the encryption key after the conclusion of the cross authentication protocol.

$\Phi_{CA,KEY,PROVER}$:=

KOHonest(K_λ,{ENK}) ⊃

Has(X, K_λ) ∧ Has(T, K_λ)

The following represents the security goals of the Authenticator. The first shows the authentication completion goal from the authenticator's view. It shows the encrypted messages being exchanged and their corresponding responses. The respective party responds only if receives the expected message.

$\Phi_{CA,AUTH,AUTHENTICATOR}$:=

KOHonest(K_λ,{*ENK*}) ⊃

Send(*X,* "MSG1",*r, s, t,* $\hat{X}$, $\hat{T}$, enc_0) <

Rcve(*T,* "MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_0) <

Send(*T,* "MSG2", $\hat{X}$, $\hat{T}$, enc_1) <

Rcve(*X,* "MSG2", $\hat{X}$, $\hat{T}$, enc_1) <

Send(*X,* "MSG3", $\hat{X}$, $\hat{T}$, enc_2) <

Rcve(*T,* "MSG3", $\hat{X}$, $\hat{T}$, enc_2) <

Send(*T,* "MSG4", $\hat{X}$, $\hat{T}$, enc_3)

The Key Security and the key freshness security goal of the authenticator is the same as the prover.

$\Phi_{CA,KF,AUTHENTICATOR}$:=$\Phi_{CA,KF,PROVER}$

$\Phi_{CA,KEY,AUTHENTICATOR}$:= $\Phi_{CA,KEY,PROVER}$

Figure 19: Strands, Invariants and Security Goals of the Cross Authentication Stage

The formal security theorem has been given below. The first two enumerations are the security goals of the prover and the authenticator, respectively. The third enumeration is to verify that the protocol adheres to the invariants.

Theorem

(i) $\Gamma_{CA,1} \wedge \Gamma_{CA,2} \vdash$

$\Theta_{CA}[\mathbf{CA:INIT}]_X$

$\Phi_{CA,\{AUTH,\ KEY,\ KF\},PROVER}$

(ii) $\Gamma_{CA,1} \wedge \Gamma_{CA,2} \vdash$

$\Theta_{CA}[\mathbf{CA:RESP}]_T$

$\Phi_{CA,\{AUTH,\ KEY,\ KF\},AUTHENTICATOR}$

(iii)CA $\vdash \Gamma_{CA,1} \wedge \Gamma_{CA,2}$

The enumerations (i) reads given the precondition Θ_{CA} and the invariants $\Gamma_{CA,1}$ and $\Gamma_{CA,2}$, once the initiator role is executed, the goal $\Phi_{CA,\{AUTH,\ KEY,\ KF\},PROVER}$ is guaranteed to hold. Enumeration (ii) is analogous to (i), but holds for the respondent. Enumeration (iii) states that $\Gamma_{CA,1}$ and $\Gamma_{CA,2}$ are the security invariants of CA.

By matching the conversation security goals of the prover and the authenticator, we prove that the above theorem holds. Here we give only a brief walk through of the proof for the cross authentication phase only from the authenticator's view. The detailed proof with matching conversation for cross authentication can be found in appendix B. The proof for initial sign-up is similar and straightforward and hence been omitted. From the precondition Θ_{CA}, both of the prover and the authenticator have the lightweight encryption algorithm. On receiving message MSG3, the authenticator knows that the sender of this message possesses the K_λ and must have computed the encryption of MSG3. Based on the security invariants and the message identifiers, if the all parties who have access to the Key and ENK (i.e. X or T) are behaving honestly, then the authenticator knows that the other participants in the protocol must be the prover who has sent MSG3. As a participant of the protocol, the authenticator

also knows that the prover must also have verified the encryption of MSG2, which was sent earlier by the authenticator, before sending MSG3. Thus, the authenticator can be ensured that every message sent and received by the prover matches the authenticator's. With some other temporal tricks, the matching conversations of protocol can be achieved and the goal $\Phi_{CA,AUTH,AUTHENTICATOR}$ can thus be established and the other security goals follow. Similar procedure can be followed for the prover's view and also to prove the logic derivation of the initial sign up stage.

4.3.2 Security Analysis

In this section, we analyze the security function of our proposed scheme.

- Mutual Authentication:

A mutual authentication between the users or the smart meters and the NAN gateways can be established using our scheme. First the group membership of the users is ensured by the AM. Only the legitimate users can prove the membership using the one-way accumulator. Then the NAN gateway ensures if the users is legitimate by checking if the intended user can encrypt and decrypt using the dynamic key. Only the legitimate user can create the key using the seed and the same lightweight encryption algorithm. The combination of this dynamic encryption with the zero knowledge proof of knowledge of prime factorization of y_p ensures the legibility of the intended users. Moreover, the flooding of the messages is controlled using the encrypted message sequence.

- Security against Man-in-the-middle Attacks:

Although the Diffie-Hellman protocol used for session key generation is prone to man-in-the-middle attack, but as we have amalgamated this protocol with identity authentication using prime factorization zero-knowledge proof, hence the changes made by the user to the message parameters will be detected by the cross authentication stage.

- Security against Replay Attacks:

The attacker can eavesdrop and replay the messages to the destination node pretending to be a legitimate node. However, the counter value being exchanged in the initial sign up is preserved by both the parties to verify that the messages are not being repeated. Hence the attack will not be successful. For the cross authentication stage, the sequence value is also encrypted and hence provides protection against replay attacks.

- Security against Impersonation Attacks:

If the intruder seeks the help of two authenticated nodes having $y_a = p_a \,.\, q_a$ and $y_b = p_b \,.\, q_b$, and Z_a and Z_b as partial accumulation values. Then the attacker can impersonate an identity $y_c = p_a.\, q_b$ and Z_c as partial accumulation value. But the value Z_c must be then equal to Z_a^{-pbqa} or Z_b^{-qapb}, but the attacker will be faced by the RSA problem to calculate these values and hence cannot launch an attack.

- Security against DoS Attacks:

The intruder can replay the message1 to the authenticator repeatedly to overburden it computationally and lead it to the DoS attack. The intruder can do the same with the prover by repeatedly sending message2. However, as we mentioned in the cross authentication procedure, once

the authenticator sends message2, any repeated message1 will not be processed and the previously calculated message2 will be retransmitted. Similarly, once the prover has sent message3, any copies of message2 received after that will not be processed by the prover and the previously calculated message3 will be retransmitted. Similarly, the counter value stored by the user and the AM during the initial sign up phase will cause all the repeated messages to be discarded. Hence the DoS attacks will not be successful.

- Forward Security:

The values l_a, l_p and r, s and t are randomly selected for every authentication session and are uncorrelated to previous session. These values along with the key generated using this value is deleted as the session ends. Hence even if the key is leaked in one session, it will not affect the security of sessions before that session.

4.3.3 Formal Verification

We perform the formal verification of the proposed scheme using PAT. PAT is the state-of-the-art model checking tool to verify the correctness of a system. This self-contained framework supports the reachability analysis and deadlock-freeness analysis as well as the refinement checking and full linear temporal logic (LTL) model checking. It has a powerful simulator and user friendly approach that enables us to verify the set assertions by simulating all paths that could possibly reach the defined assertions. This layered architecture supports eleven different modules according to the specific requirements of model checking for the various system analysis. Our attack model is based on the

Communicating Sequential Processes (CSP) module as it enables enough flexibility to model the system according to our requirements. The verification has been performed for three different attack scenarios to describe the verification procedure. Similar procedure has been followed for other attacks.

By the attack model used in PAT, the intruder is assumed to possess the ability to generate big prime numbers, to generate seed value, to eavesdrop the communication between the intended parties and also to collaborate with authorized users for internal node attacks. In the first attack scenario, the attacker monitors the conversation between the prover and the authenticator in the cross authentication stage. It can get access to the values r, s and t used for the encryption key derivation from the message1 sent by the prover to the authenticator. However, as the attacker does not have access to the initial key space and the lightweight encryption algorithm, it is not able to insert any illegal or undesired information to the authentication procedure by modifying the encrypted messages. The execution trace as depicted in Figure 20 shows that there is no possible path leading to failure of verification assertion of the LCSNA while it is under man-in-the middle attack.

In the second attack scenario, the attacker collaborates with two other nodes which have already been authenticated. It uses their identification i.e. the prime factorization of y_1 and y_2 and uses that to create a new identity. Similarly, it uses the partial accumulation value of these misbehaving nodes and create a new partial accumulation value and sends it to the AM in the initial sign up stage. But due to the strong RSA problem it is not able to verify the zero-knowledge proof.

The attacker also tries to resend previously stored valid messages from previous sessions. But as the values l_a, l_p and r, s and t are randomly generated every session and does not depend on values from previous sessions, the attacker doesn't have enough information to qualify for a valid member of the network and hence the attacks are not successful. Figure 21 show the execution trace where the assertion of failure of LCSNA under impersonation attacks fails.

In another attack scenario, the attacker passively listens to the messages being exchanged during cross authentication and captures message 1 and send it repeatedly to the authenticator and in second case tries to send message 2 repeatedly to the prover to increase the computation cost for both the parties ultimately leading to a DoS attack. But as the repeated messages are not processed and simply the previous messages are repeated, the attack is not successful. Figure 22 shows that no trace leads to the failure by DoS attack either.

Finally, in the last scenario the attacker uses the messages from previous cross authentication stage and repeat them in the current cross-authentication phase before the message1 from the intended prover can reach to the authenticator. However, the values used for key derivation are randomly chosen and not related to the previous authentication cycle and thus will not match to the current cycle of the protocol and hence these messages are discarded.

Also the messages repeated from the current run will not computationally overburden the authenticator or the prover because they are not reprocessed and only the already calculated response is

retransmitted. The assertion verification in Figure 23 depicts that there is no possible execution trace that will lead to the failure of the protocol under replay attacks.

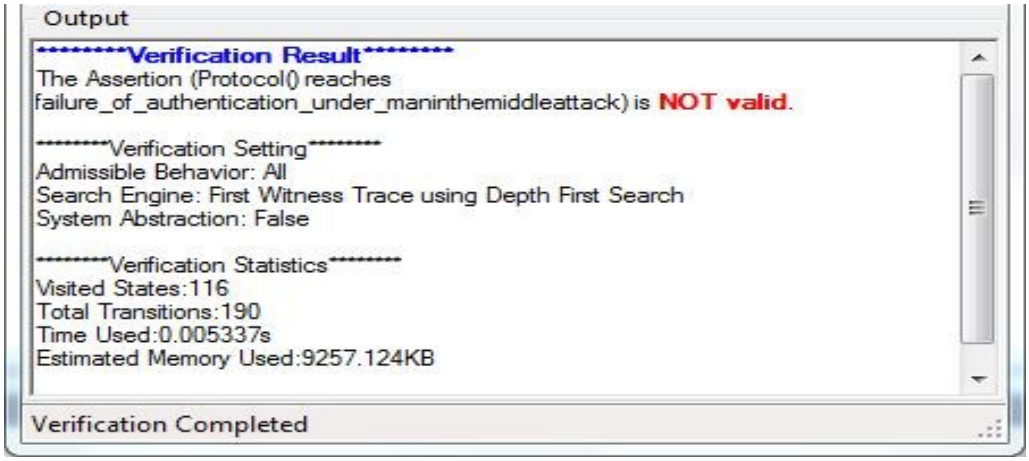

Figure 20: LCSNA under Man-in-the-Middle Attack.

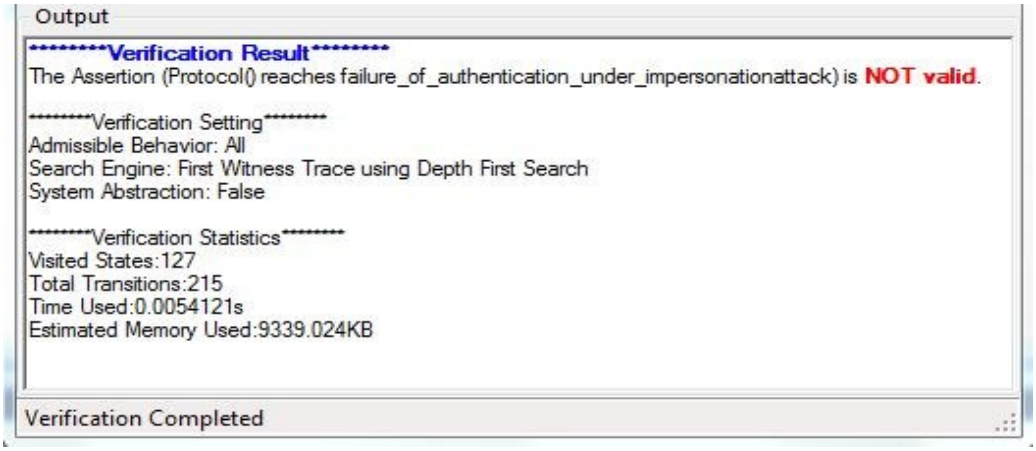

Figure 21: LCSNA under Impersonation Attack

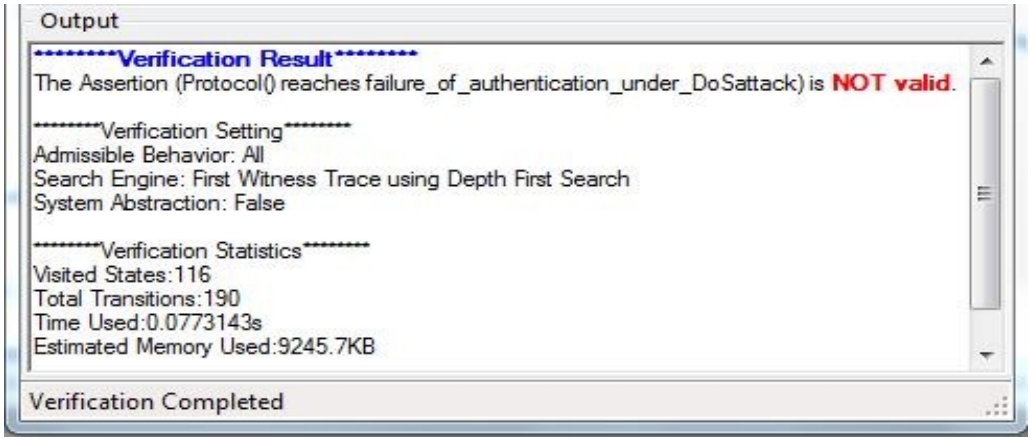

Figure 22: LCSNA under DoS Attack

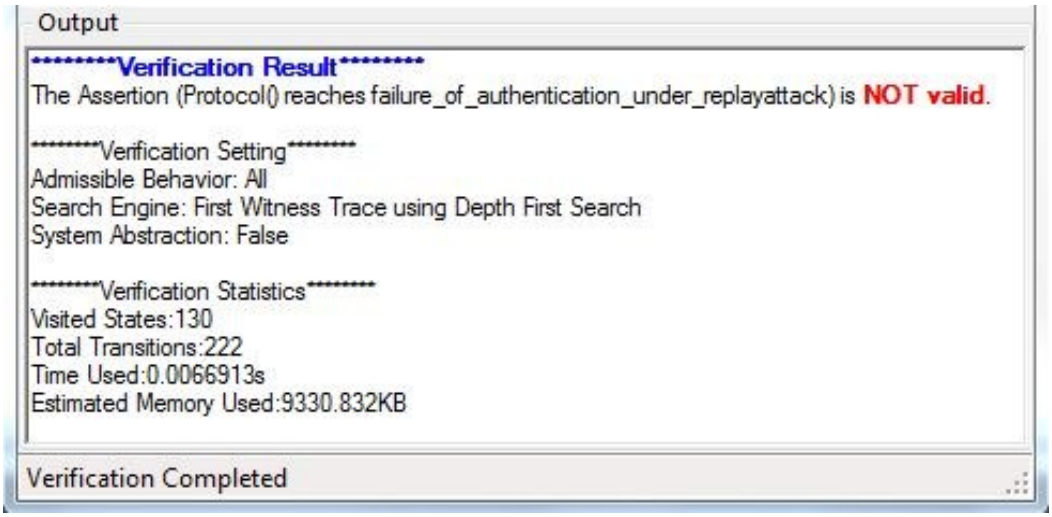

Figure 23: LCSNA under Replay Attack

4.4 Efficiency Analysis and Simulation Results

In this section, we perform the efficiency analysis of the proposed scheme and compare it to some of the other schemes in the literature to prove that our scheme provides the desired outcomes without overburdening the resource limited network devices to fulfill the required security requirements. First, we perform the efficiency analysis to compare the computation and storage costs of the proposed scheme. Then, we present the results of the simulation executions to compare the average end-to-end delay and the power consumption caused by our scheme to other proposed schemes. Then we list the major advantages provided by our scheme.

4.4.1 Efficiency Analysis

Here we evaluate the computation and storage costs of our proposed scheme. The costs for selection of rigid and big prime integers are considered small enough to be ignored.

The computation cost for the users during the initial sign up stage comes out to be $M_q + M_g + 2M_s$, where M_q is the computation cost for the modular exponentiation, M_g is the cost for logarithmic operation and M_s is the cost for square root derivation. The computation cost for the AM is 3 M_q and the cost to compute the system accumulation and partial accumulation values for each node. However, to improve the computational efficiency we can first evaluate $y = \sum y_i \ mod \ \varphi(n)$, where i is the number of users, and then by one modular exponentiation we can find the accumulation value $Z=x^y \ mod \ N$. Similarly, for partial accumulation, we can first find $1/y_i \ mod \ \varphi(n)$ and use one modular exponentiation to find the final value. Thus the total computation cost becomes $(4 + m)M_q$ for AM, where m is the total number of users. As the values M_g and M_s are much less than M_q, the computation cost for the users is much less compared to the AM. Hence the computation cost does not burden the resource limited users.

For the authentication phase the computation cost for the authenticator comes out to be $3M_q + 4M_k$ and the computation cost for the prover is $2M_q + 4M_k + M_s$, where M_k is the computation cost for encryption of the message using ENK. As we can observe the computation cost is low and almost similar for both the parties which are resource limited.

Let the storage cost required for storing the system accumulation value be S_{acc} and the storage cost for ENK is S_k. These are the only storage costs required for the users including the NAN and HAN gateways. However, the AM has to incur additional storage cost to maintain information and status of different users to be used in later processes such as membership revocation. However, the AM having higher computational and storage capability can accommodate these costs.

Table 3 depicts the comparison of the computation cost and the storage cost of the proposed scheme to the costs in scheme proposed in [73] using dynamic accumulator based authentication scheme called troupe based authentication (TBA). In TBA, the users undergo the authentication vis central computing authority (CCA). As we can observe that the proposed scheme costs significantly lesser and the security provided by the new scheme is also.

Table 3: Comparison of Computation and Storage Cost

Entity	**CCA[3]**	**User[3]**	**Prover**	**Authenticator**	**User**	**AM**
Computation Cost	mM_q	$10M_q$	$2M_q + 4M_k + M_s$	$3M_q + 4M_k$	$M_q + M_g + 2M_s$	$(4 + m)M_q$
Storage cost	$8S_{com}+S_{cha}$	$8S_{com}+S_{cha}$	S_k	S_k	S_{acc}	$S_{acc}+S_{info}$

4.4.2 Simulation Results

In this section, we investigate the impact on the delay performance and energy consumption of the participants of the LCSNA protocol and then compare the results to that of the scheme proposed in [73] i.e. TBA. To access the delay and energy consumption, the simulation experiment by using MATLAB has been conducted to simulate the authentication between the participants and that of the root controller. The lightweight encryption algorithm as mentioned in the cross authentication phase has been selected as AES-128 algorithm to encrypt and decrypt the messages. The message length has been fixed to 110 bytes and the big integers being generated by the parties are fixed to a minimum length of 1024 bits.

For calculating average delay between the two parties, we adopted the performance benchmarks for the AES-128, coded in C++, compiled with Microsoft visual C++ 2005 and ran on Intel Core 2, 1.83 GHz processor under windows vista in 32-bit mode. The delay performance analysis for modular exponentiation has been based on the results of [74]. The performance benchmark for the energy consumption analysis of the AES-128 protocol has been derived from the findings of [65]. The energy consumption analysis for modular exponentiation has been base on the findings of [74]. We have tailored this data to the LCSNA protocol and evaluated the energy consumption per node.

The algorithm is run under a set of 10,000 attack scenarios with increasing rate of unknown attacks. The unknown attacks account for the interruption of the execution of the proposed protocol due to undiscovered attacks on the algorithm or due to the hardware, channel or other unanticipated implications.

In Figure 24, we assess the delay performance of the execution of proposed LCSNA scheme and compare it with that of the TBA. As it can be observed, when the execution of the proposed scheme is under only the known attacks, the proposed scheme results in the minimum average delay between the nodes. However, as the fraction of unknown attacks increases, the average delays for both the LCSNA and the TBA increase as they can be randomly interrupted at any step of the protocol. The interrupted protocol session is reinitiated and hence the delay overhead is getting accumulated. However, the delay performance for our LCSNA scheme still remains better. Hence we can conclude that the proposed scheme has efficient delay performance while providing better security reliability.

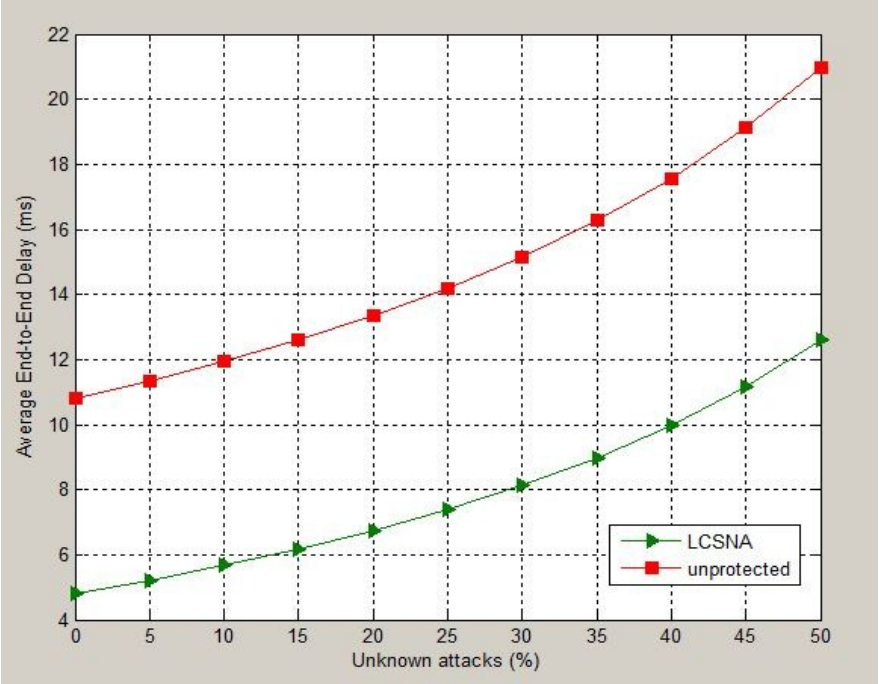

Figure 24: Average End-to-end Delay

Similarly, in Figure 25, we evaluate the impact on energy consumption when two protocols are under both known attacks and unknown attacks.

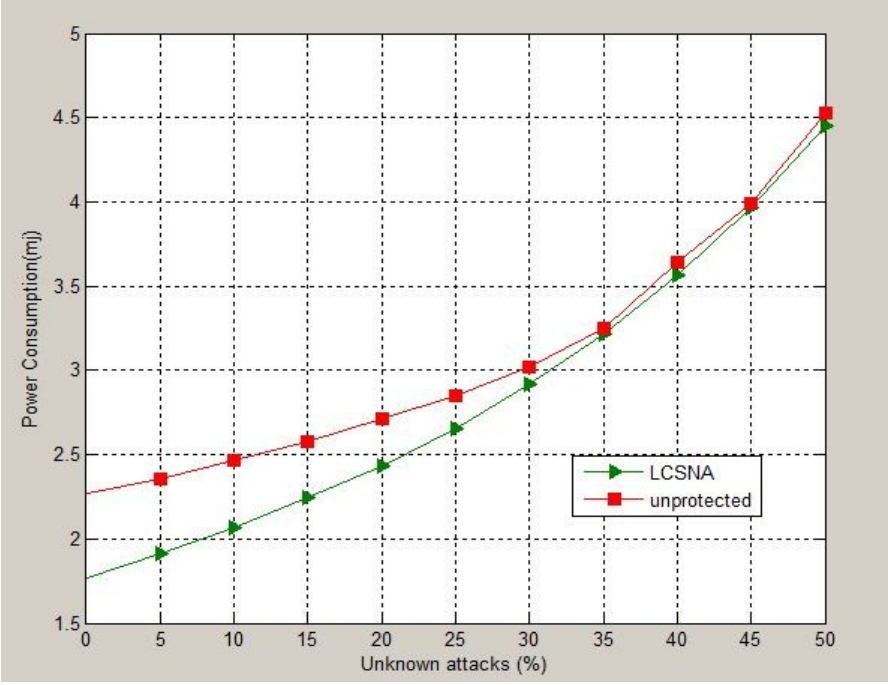

Figure 25: Average Energy Consumption per Node

It is clear when they are under the known attacks i.e. 0% unknown attacks, the LCSNA scheme has a lower power consumption. But as the number of unknown attacks increases, the energy consumed by the nodes increases for both of the LCSNA and the TBA scheme. As the percentage of unknown attacks increases the power consumption of LCSNA approaches TBA but still remains lower. The reason for that being the AES encryption and decryption being repeated for failed protocol. However, the security and power consumption tradeoff is still more efficient from TBA.

4.5 Summary

In this chapter, we present a novel authentication scheme, LCSNA for smart grid NAN domain by combining the dynamic one-way accumulators with simple zero-knowledge protocols and implement it on SDN architecture. The proposed scheme is suitable for mesh networks which are optimal for NAN domain and also easy addition and revocation of members make it suitable for large distributed networks. The formal verification of the proposed scheme proves that it is secure against man in the middle attack, forgery attacks or denial of service by replay attacks. The efficiency analysis shows that the computation and storage cost required for our scheme is optimal for resource constrained smart meters and HAN and NAN gateways and the simulation results demonstrate the effectiveness of the proposed strategy under the known and the unknown attacks. Finally, the implementation using distributed SDN network architecture makes deployment of new protocols and the future changes in our proposed scheme highly convenient and cost efficient.

Chapter 5. Conclusion and Future Work

In the thesis, we have first proposed two schemes: an improved dynamic key refreshment scheme and a SDN based authentication scheme. Both schemes aim to enhance the security in the NAN domain of the smart grid infrastructure. In the enhanced key refreshment scheme, a simple one-way hash based protection to the first message of the MKHSH protocol has been proposed. In the original key refreshment scheme, the message 1 was not encrypted and this message could be forged and replayed back in the consecutive key refreshment cycles to cause the DoS attack. Our proposed solution eradicates this issue while causing minimal computation expense over the resource sensitive network devices of the NAN domain. We also proposed an enhancement to the key refreshment scheme by performing multiple MKHSH protocol updates in one MSK session. This further improves the reliability and the efficacy of the key refreshment scheme. In the SDN based authentication scheme, a combination of strong RSA based dynamic one-way accumulators and zero- knowledge proofs have been used to device a cost efficient authentication scheme suitable for a mesh network architecture. Using SDN as the backbone of the architecture enables the proposed scheme to accommodate updates to the protocol. The use of one-way accumulators causes minimal computation expense and supports forward security. The first phase of the scheme signs-up all the members of the network to the root controller. In the second phase the network elements can perform cross authentication replicating a mesh network behavior. Our analysis shows that the proposed scheme not only withstands multitude of attacks but also maintains a good balance between the system security and efficiency.

The thesis focuses mainly on the authentication aspect of the NAN domain in the smart grid security. However significant amount of work is still required to make the deployment of smart grid technology a complete success so that the benefits of this fruitful innovation can be achieved to the fullest. Hence more efforts should be made for further security in the following aspects.

- WAN Security: As we move from the HAN domain to the higher domains the security requirements become more and more stringent. Hence the security solutions in the WAN domain require much complex and reliable outcomes. Although the computation power of devices in these domains are high and can accommodate already established complex cryptographic schemes, but the consideration of the smart grid architecture and the transition from NAN to WAN domain in terms of security aspects are also to be considered. Very limited work has been presented in the literature to tackle the security requirements of the WAN domain and most of the security reliability is assumed on the protocols that a specific utility will adopt. Hence unified security solutions are to be developed for the WAN domain also.
- The integration of renewable energy is also one of the major benefits of smart grid deployment. This will allow the consumer party to become the generator of the energy simultaneously. However, the legitimacy of those users and the authorization for certain users to generate and sell energy to other users and the security of this energy transaction are still areas where suitable research is pending. Also the power quality being fed to the smart grid will be influenced which should also be considered during the network automation of the smart grid.

In addition to the above mentioned potential research areas, there are several open-research issues related to smart grid security. The future research could focus on the following aspects:

- Compliance with future technologies: The use of communication resources and devices is increasing exponentially, the upcoming wireless technologies will be designed to accommodate these demands. Hence the smart grid infrastructure must also comply with these advances accordingly.
- Uniformity: Most of the solutions tend to focus on certain specific aspects of security requirements. However, the implementation of a combination of multiple complex mechanisms makes the process much more cumbersome. Hence, rather uniform security architecture should be devised that fulfills all the corresponding security requirements.
- Computational Complexity: The network devices, especially in the lower domains are resource limited and hence cannot sustain highly complex and computationally burdensome security mechanisms. Hence schemes enabling a good balance between computational cost and system security still need to be devised.
- Flexible: As the smart grids consists of highly distributed networks with large number of network devices, any changes to the security protocols will require changes down to the network device hardware which is very ineffective. However, as the communication technologies will advance, changes will be inevitable. So architecture that can accommodate such changes without requiring changes to network hardware every time can be devised.

BIBLIOGRAPHY

1. R. Yu, Y. Zhang, S. Gjessing, C. Yuen, S. Xie and M. Guizani, "Cognitive radio based hierarchical communications infrastructure for smart grid," IEEE Network, Vol. 25, No. 5, pp. 6-14, September-October 2011.

2. A. Zaballos, A. Vallejo and J. M. Selga, "Heterogeneous communication architecture for the smart grid," IEEE Network, Vol. 25, No. 5, pp. 30-37, September-October 2011.

3. A. Ipakchi,F. Albuyeh, "Grid of the future," *IEEE Power and Energy Magazine*, Vol.7, No.2, pp.52-62, March-April 2009.

4. B. G. Kim, Y. Zhang, M. van der Schaar and J. W. Lee, "Dynamic pricing for smart grid with reinforcement learning," IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 640-645, 2014.

5. Y. Wu, V. K. N. Lau, D. H. K. Tsang, L. Qian and Limin Meng, "Optimal exploitation of renewable energy for residential smart grid with supply-demand model," 7th International ICST Conference on Communications and Networking in China (CHINACOM), pp. 87-92, August 2012.

6. Jaeseok Choi, Jeongje Park, M. Shahidehpour and R. Billinton, "Assessment of CO2 reduction by renewable energy generators," Innovative Smart Grid Technologies (ISGT), 2010, pp. 1-5, January 2010.

7. R. Davies, “Hydro one’s smart meter initiative paves way for defining the smart grid of the future,” *Power & Energy Society General Meeting (PES ’09)*,pp. 1-2, 2009.

8. D. Sun, “The Utilization and Development Strategies of Smart Gridand New Energy,” In *Proceedings of Asia-Pacific Power and Energy Engineering Conference (APPEEC 2010),* pp.1-4, 2010.

9. R. Hassan, M. Abdallah and G. Radman, "Load shedding in smart grid: A reliable efficient Ad-Hoc broadcast algorithm for smart house," Proceedings of IEEE Southeastcon, pp. 1-5, March 2012.

10. U.S. Department of Commerce, National Institute of Standards and Technology (2010,January) NIST Special Publication 1108, NIST Framework and Roadmap for Smart Grid Interoperability Standards,

Release1.0[Online]Available:http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf

11. V. Namboodiri, V. Aravinthan, S. N. Mohapatra, B. Karimi and W. Jewell, "Toward a Secure Wireless-Based Home Area Network for Metering in Smart Grids," IEEE Systems Journal, Vol. 8, No. 2, pp. 509-520, June 2014.

12. W. Meng, R. Ma and H. H. Chen, "Smart grid neighborhood area networks: a survey," IEEE Network, Vol. 28, No. 1, pp. 24-32, January-February 2014.

13. E. Bou-Harb, C. Fachkha, M. Pourzandi, M. Debbabi and C. Assi, "Communication security for smart grid distribution networks," in IEEE Communications Magazine, vol. 51, no. 1, pp. 42-49, January 2013.

14. H. Farooq,L. T. Jung, "Choices available for implementing smart grid communication network," *2014 International Conference on Computer and Information Sciences (ICCOINS)*, pp.1-5, June 2014.

15. NIST, Smart grid cyber security strategy and requirements. Aug. 2010.[Online]. Available:http://csrc.nist.gov/publications/nistir/ir7628/nistir-7628_vol1.pdf

16. E. K. Lee, M. Gerla and S. Y. Oh, "Physical layer security in wireless smart grid," IEEE Communications Magazine, Vol. 50, No. 8, pp. 46-52, August 2012.

17. A. J. McBride,A. R. McGee, "Assessing smart Grid security," *Bell Labs Technical Journal*, Vol.17, No.3, pp.87-103, December 2012.

18. M. A. Lisovich,D. K. Mulligan,S. B. Wicker, "Inferring Personal Information from Demand-Response Systems," *IEEE Security & Privacy*, Vol.8, No.1, pp.11-20, 2010.

19. Z. Lu, X. Lu, W. Wang and C. Wang, "Review and evaluation of security threats on the communication networks in the smart grid," MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010, pp. 1830-1835, October-November 2010.

20. D. Grochocki et al., "AMI threats, intrusion detection requirements and deployment recommendations," IEEE Third International Conference on Smart Grid Communications (SmartGridComm), pp. 395-400, 2012.

21. D. Kushner, "The real story of stuxnet," *IEEE Spectrum,* Vol.50, No.3, pp.48-53, March 2013.

22. S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, pp.4490-4494, November. 2011

23. Z. Dehlawi and N. Abokhodair, "Saudi Arabia's response to cyber conflict: A case study of the Shamoon malware incident," IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 73-75, June 2013.

24. S. H. Seo, X. Ding and E. Bertino, "Encryption key management for secure communication in smart advanced metering infrastructures," IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 498-503, October 2013.

25. Fangming Zhao, Y. Hanatani, Y. Komano, B. Smyth, S. Ito and T. Kambayashi, "Secure authenticated key exchange with revocation for smart grid," 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), pp. 1-8, January 2012.

26. Sangji Lee, Jinsuk Bong, Sunhee Shin and Yongtae Shin, "A security mechanism of Smart Grid AMI network through smart device mutual authentication," The International Conference on Information Networking 2014 (ICOIN2014), pp. 592-595, February 2014.

27. N. Liu, J. Chen, L. Zhu, J. Zhang and Y. He, "A Key Management Scheme for Secure Communications of Advanced Metering Infrastructure in Smart Grid," IEEE Transactions on Industrial Electronics, Vol. 60, No. 10, pp. 4746-4756, Oct. 2013.

28. R. Lu, X. Lin, Zhiguo Shi and X. Shen, "EATH: An efficient aggregate authentication protocol for smart grid communications," 2013 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1819-1824, April 2013.

29. X. Duan and X. Wang, "Authentication handover and privacy protection in 5G hetnets using software-defined networking," IEEE Communications Magazine, Vol. 53, No. 4, pp. 28-35, April 2015.

30. H. Gharavi and Bin Hu, "4-way handshaking protection for wireless mesh network security in smart grid," *IEEE Global Communications Conference (GLOBECOM),* pp.790-795, December 2013.

31. M. Lu, Z. Shi, R. Lu, R. Sun and X. S. Shen, "PPPA: A practical privacy-preserving aggregation scheme for smart grid communications," IEEE/CIC International Conference on Communications in China (ICCC), pp. 692-697, August 2013.

32. A. Beussink, K. Akkaya, I. F. Senturk and M. M. E. A. Mahmoud, "Preserving consumer privacy on IEEE 802.11s-based smart grid AMI networks using data obfuscation," 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 658-663, May 2014.

33. S. C. Yip, K. Wong, R. C. W. Phan, S. W. Tan, I. Ku and W. P. Hew, "A Privacy-Preserving and Cheat-Resilient electricity consumption reporting Scheme for smart grids," 2014 International Conference on Computer, Information and Telecommunication Systems (CITS), pp. 1-5, July 2014.

34. W. Jia, H. Zhu, Z. Cao, X. Dong and C. Xiao, "Human-Factor-Aware Privacy-Preserving Aggregation in Smart Grid," IEEE Systems Journal, Vol. 8, No. 2, pp. 598-607, June 2014.

35. Y. Liu, P. Ning and M.K. Reiter, “False data injection attacks against state estimation in electric power grids,” ACM Transactions on Information and System Security (TISSEC), Vol. 14, No. 1, , May 2011.

36. L. Yang and F. Li, "Detecting false data injection in smart grid in-network aggregation," IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 408-413, October 2013.

37. K. Manandhar, Xiaojun Cao, Fei Hu and Y. Liu, "Combating False Data Injection Attacks in Smart Grid using Kalman Filter," International Conference on Computing, Networking and Communications (ICNC), pp. 16-20, February 2014.

38. J. H. Reed and C. R. A. Gonzalez, "Enhancing Smart Grid cyber security using power fingerprinting: Integrity assessment and intrusion detection," Future of Instrumentation International Workshop (FIIW), pp. 1-3, October 2012.

39. M. Esmalifalak; L. Liu; N. Nguyen; R. Zheng; Z. Han, "Detecting Stealthy False Data Injection Using Machine Learning in Smart Grid," *IEEE Systems Journal* , Vol. PP, No.99, pp.1-9, 2014.

40. H. Sedghi and E. Jonckheere, "Statistical structure learning of smart grid for detection of false data injection," IEEE Power & Energy Society General Meeting, pp. 1-5, July 2013.

41. N. Beigi-Mohammadi, J. Mišić, H. Khazaei and V. B. Mišić, "An intrusion detection system for smart grid neighborhood area network," IEEE International Conference on Communications (ICC), pp. 4125-4130, June 2014.

42. Y. Zhang, L. Wang, W. Sun, R. C. Green and M. Alam, "Artificial immune system based intrusion detection in a distributed hierarchical network architecture of smart grid," IEEE Power and Energy Society General Meeting, pp. 1-8, July 2011.

43. Z. Lu, W. Wang and C. Wang, "Camouflage Traffic: Minimizing Message Delay for Smart Grid Applications under Jamming," IEEE Transactions on Dependable and Secure Computing, Vol. 12, No. 1, pp. 31-44, Jan.-Feb. 2015.

44. Hongbo Liu, Yingying Chen, Mooi Choo Chuah and Jie Yang, "Towards self-healing smart grid via intelligent local controller switching under jamming," IEEE Conference on Communications and Network Security (CNS), pp. 127-135, October 2013.

45. S. Ruj and A. Nayak, "A Decentralized Security Framework for Data Aggregation and Access Control in Smart Grids," IEEE Transactions on Smart Grid, Vol. 4, No. 1, pp. 196-205, March 2013.

46. Qinghai Gao, "Biometric authentication in Smart Grid," 2012 International Energy and Sustainability Conference (IESC), pp. 1-5, March 2012.

47. J. Choi, I. Shin, J. Seo and C. Lee, "An Efficient Message Authentication for Non-repudiation of the Smart Metering Service," 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering (CNSI), pp. 331-333, May 2011.

48. Z. Xiao, Y. Xiao and D. H. c. Du, "Non-repudiation in neighborhood area networks for smart grid," IEEE Communications Magazine, Vol. 51, No. 1, pp. 18-26, January 2013.

49. A. Irfan, N. Taj and S. A. Mahmud, "A Novel Secure SDN/LTE Based Architecture for Smart Grid Security," IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), pp. 762-769, 2015.

50. IEEE Standard for Information Technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking, *IEEE Std 802.11s-2011*, September 2011, pp.1,372.

51. IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks, *IEEE Std 802.15.4g-2012 (Amendment to IEEE Std 802.15.4-2011)*, April 2012, pp.1,252.

52. M. Weixiao, Ma Ruofei and C. Hsiao-Hwa, "Smart grid neighborhood area networks: a survey," *IEEE Network* , Vol.28, No.1, February 2014, pp.24,32.

53. K. H. Chang, "Interoperable Nan Standards: a path to cost-effective smart grid solutions," *IEEE Wireless Communications* , Vol.20, No.3, June 2013, pp.4,5.

54. T. Braskich, W. S. Conner, J. Kruys, S. Emeott , J. Walker, M. Zhao, R. Falk, “Efficient mesh security and link establishment”, November 2006, doc.: IEEE 802.11-06/1470r3.

55. B. Hu and H. Gharavi, "Smart Grid Mesh Network Security Using Dynamic Key Distribution With Merkle Tree 4-Way Handshaking,", *IEEE Transactions on Smart Grid*, Vol.5, No.2, March 2014, pp.550,558.

56. T. Braskich and S. Emeott. “Mesh key holder protocol improvements”, June 2007, doc.:IEEE 11-07/1987r1.

57. Secure Hash Standard, SHA-1, FIPS PUB 180-1 [Online]. Available: http://www.itl.nist.gov/fipspubs/fip180-1.html

58. Secure Hash Standard, SHA-2, FIPS PUB 180-2 [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html

59. A. Datta, A. Derek, J. C. Mitchell, and A. Roy. “Protocol composition logic (PCL)”. *Electronic Notes in Theoretical Computer Science*, Vol.172, April 2007, pp.311–358.

60. N. Durgin, J. Mitchell, and D. Pavlovic.,“A compositional logic for proving security properties of protocols”. *Journal of Compuer Security*, Vol.11, No.4, July 2003, pp.677-721.

61. A. Roy, A. Datta, A. Derek, J. C. Mitchell, and J-P. Seifert. “Secrecyanalysis in protocol composition logic,” *Proceedings of 11th Annual AsianComputing Science Conference*, Vol.4435, December 2006, pp.197-213.

62. A. Datta, A. Derek, J. C. Mitchell, and D. Pavlovic. “A derivation system and compositional logic for security protocols.” *Journal of Computer Security*, Vol.13, No.3, May 2005, pp.423-482.

63. A. Datta, A. Derek, J.C.Mitchell, and B. Warinschi, “Computationally Sound Compositional Logic for Key Exchange Protocols,”*Proceedings of 19th IEEE Computer Security Foundations Workshop*, 2006, pp. 321-334.

64. D. Kuhlman , “A Proof of Security of a Mesh Security Architecture,” *IEEE Press*, 2007, tech. report 802.11-07/2436r0.

65. N. R. Potlapally, S. Ravi, A. Raghunathan, N. K. Jha, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols," *IEEE Transactions on Mobile Computing* , Vol.5, No.2, February 2006, pp.128-143.

66. J. G. Andrews *et al*., "What Will 5G Be?," *IEEE Journal on Selected Areas in Communications*, Vol. 32, No. 6, June 2014, pp. 1065-1082.

67. J. Camenisch and A. Lysyanskaya, “Dynamic accumulators and applications to efficient revocation of anonymous credentials”, *In the proceedings of Advances in Cryptology—Crypto’02, LNCS*, Vol. 2442, Springer-Verlag, 2002, pp. 61–76.

68. B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 3, 2014, pp. 1617-1634.

69. S. H. Yeganeh and Y. Ganjali. “Kandoo: a framework for efficient and scalable offloading of control applications”. *In the proceedings of 1st workshop on Hot topics in software defined networks*, *HotSDN ’12*, 2012, pp. 19-24.

70. J. Benaloh and M. de Mare, “One-way accumulators: a decentralized alternative to digital signatures”, *In the proceedings of Advances in Cryptology—Eurocrypt’93, LNCS*, Vol. 765, 1993, pp. 274–285.

71. R. Gennaro, D. Micciancio, and T. Rabin, “An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products”, *In the proceedings of 5th ACM Conference on Computer and Communications Security*, 1998, pp.67-72.

72. Ren Yu and Li Yue. “Zero-Knowledge Proof of Big Integer Factorization”, *Computer Science*, Vol.33, No.9, 2006, pp.298-300.

73. S. Gokhale and P. Dasgupta, "Distributed authentication for peer-to-peer networks," *In the proceedings of 2003 Symposium* on *Applications and the Internet Workshops,* 2003, pp. 347-353.

74. L. Zhong, "Modular Exponentiation Algorithm Analysis for Energy Consumption and Performance," *Technical Report CE-01-ZJL*, 2001, Dept. of Electrical Engineering, Princeton University.

Appendix

A. Proof of security goals of SEDKR

A.1 Matching conversations, MKD:

AA1, ARP, AA4, Θ_{MKHSH}

$[MKHSH : MKD]_T$

(Rcve (T, "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Send (T, "MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) <

Rcve (T, "MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) <

Send (T, "MKH4", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2)) ..... (9)

ARP, VER, Θ_{MKHSH}

$[MKHSH : MKD]_T$

Rcve (T, "MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) ⊃

∃Z. Computes (Z, HASH $_{mptkX,T}$ ("MKH3", x, t, , $\hat{X}$, $\hat{T}$, $INFO_X$)) ∧

Send (Z, HASH $_{mptkX,T}$ ("MKH3", x, t, , $\hat{X}$, $\hat{T}$, $INFO_X$))<

Rcve (T, "MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) ..... (10)

$\Gamma_{MKHSH, SI, 1}$, HASH1

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃ Computes (Z, HASH $_{mptkX,T}$ ("MKH3", x, t, , $\hat{X}$, $\hat{T}$, $INFO_X$)) ⊃

Has (Z, $mptk_{X,T}$) ⊃ $\hat{Z}=\hat{X} \vee \hat{Z}=\hat{T}$ ..... (11)

(10), (11), AA1, $\Gamma_{MKHSH, 1}$, Θ_{MKHSH}

$[MKHSH : MKD]_T$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Send (Z, HASH $_{mptkX,T}$ ("MKH3", x, t, , $\hat{X}$, $\hat{T}$, $INFO_X$)) ⊃ $\hat{Z}=\hat{X}$ ..... (12)

(10), (12), Θ_{MKHSH}

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Computes (X, HASH $_{mptkX,T}$ ("MKH3", x, t, , $\hat{X}$, $\hat{T}$, $INFO_X$)) ∧

Send (X, HASH $_{mptkX,T}$ ("MKH3", x, t, , $\hat{X}$, $\hat{T}$, $INFO_X$)) ..... (13)

(13), ϕHONESTY, Θ_{MKHSH}

$[MKHSH : MKD]_T$

KOHonest ($mptk_{X,T}$ $\{mkdk_{X,T}\}$)⊃

Computes (X, HASH $_{mptkX,T}$ (“MKH2”, x, t, , $\hat{X}$, $\hat{T}$, $INFO_T$)) ..... (14)

ARP, HASH3, Θ_{MKHSH}

Rcve (T, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) ⊃

∃Z. Computes (Z, HASH (x, $\hat{X}$, $\hat{T}$, $mkdk_{X,T}$)) ∧ Send (Z, HASH (x, $\hat{X}$, $\hat{T}$, $mkdk_{X,T}$)) ..... (15)

$\Gamma_{MKHSH, 1}$, HASH1

KOHonest ($mptk_{X,T}$ $\{mkdk_{X,T}\}$)⊃ Computes (Z, HASH (x, $\hat{X}$, $\hat{T}$, $mkdk_{X,T}$)) ⊃

Has (Z, $mkdk_{X,T}$) ⊃ $\hat{Z}$=$\hat{X}$ ⋁ $\hat{Z}$=$\hat{T}$ ..... (16)

(15), (16), AA1, $\Gamma_{MKHSH, 1}$, Θ_{MKHSH}

KOHonest ($mptk_{X,T}$ $\{mkdk_{X,T}\}$)⊃ Send (Z, HASH (x, $\hat{X}$, $\hat{T}$, $mkdk_{X,T}$)) ⊃ $\hat{Z}$=$\hat{X}$ ..... (17)

(15), (17), Θ_{MKHSH}

KOHonest ($mptk_{X,T}$ $\{mkdk_{X,T}\}$) ⊃ Computes (X, HASH (x, $\hat{X}$, $\hat{T}$, $mkdk_{X,T}$)) ⊃

Send (X, HASH (x, $\hat{X}$, $\hat{T}$, $mkdk_{X,T}$)) ..... (18)

(13), (14), (18), HASH1, Θ_{MKHSH}

$[MKHSH : MKD]_T$

KOHonest ($mptk_{X,T}$ $\{mkdk_{X,T}\}$) ⊃

Has (X, $mptk_{X,T}$) ∧ Has (X, x, t, $INFO_X$, $INFO_T$, $\hat{X}$, $\hat{T}$) ..... (19)

(10), (13), (19), ϕHONESTY, Θ_{MKHSH}

$[MKHSH : MKD]_T$

KOHonest ($mptk_{X,T}$ $\{mkdk_{X,T}\}$)⊃

Send (X, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Rcve (X, “MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) <

Send (X, “MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) ..... (20)

(10), (20), Θ_{MKHSH}

$[MKHSH : MKD]_T$

$KOHonest\ (mptk_{X,T}\ \{mkdk_{X,T}\}) \supset Send\ (X, \text{“MKH3”}, x, t, \hat{X}, \hat{T}, INFO_X, mic_1) <$

$Rcve\ (T, \text{“MKH3”}, x, t, \hat{X}, \hat{T}, INFO_X, mic_1)$ ..... (21)

ARP, VER, Θ_{MKHSH}

$[MKHSH : MKD]_T$

$KOHonest\ (mptk_{X,T}\ \{mkdk_{X,T}\}) \wedge Rcve\ (X, \text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0) \supset$

$\exists Z.\ Computes\ (Z, HASH_{mptkX,T}\ (\text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0)) \wedge$

$Send\ (Z, HASH_{mptkX,T}\ (\text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0)) \wedge$

$Send\ (Z, HASH_{mptkX,T}\ (\text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0)) <$

$Rcve\ (X, \text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0)$ ..... (22)

$\Gamma_{MKHSH, SI, 1}$, HASH1

$KOHonest\ (mptk_{X,T}\ \{mkdk_{X,T}\}) \supset$

$Computes\ (Z, HASH_{mptkX,T}\ (\text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0)) \supset$

$Has\ (Z, mptk_{X,T}) \supset \hat{Z}=\hat{X} \vee \hat{Z}=\hat{T}$ ..... (23)

(22), (23), AA1, $\Gamma_{MKHSH, 1}$, Θ_{MKHSH}

$[MKHSH : MKD]_T$

$KOHonest\ (mptk_{X,T}\ \{mkdk_{X,T}\}) \supset$

$Send\ (Z, HASH_{mptkX,T}\ (\text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0)) \supset \hat{Z}=\hat{T}$ ..... (24)

(22), (24), Θ_{MKHSH}

$[MKHSH : MKD]_T$

$Send\ (T, \text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0) <$

$Rcve\ (X, \text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0)$ ..... (25)

FS1, AN3, Θ_{MKHSH}

$[MKHSH : MKD]_T$

$FirstSend\ (T, t, \text{“MKH2”}, x, t, \hat{X}, \hat{T}, INFO_T, mic_0)$ ..... (26)

(20), (26), FS2, Θ_{MKHSH}

$[MKHSH : MKD]_T$

Send (T, "MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) < Rcve (X, "MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) ..... (27)

FS1, AN3, Θ_{MKHSH}

$[MKHSH : MKD]_T$

Honest ($\hat{X}$) ⊃ FirstSend (X, x, "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) ..... (28)

(20), (28), FS2, Θ_{MKHSH}

$[MKHSH : MKD]_T$

Honest ($\hat{X}$) ⊃ Send (X, "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Rcve (T, "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) ..... (29)

(20), (21), (27), (29), Θ_{MKHSH}

Send (X, "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Rcve (T, "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Send (T, "MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) <

Rcve (X, "MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) <

Send (X, "MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) <

Rcve (T, "MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) <

Send (T, "MKH4", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2) ..... (30)

A.2 Matching conversations, MA:

AA1, ARP, AA4, Θ_{MKHSH}

$[MKHSH : MA]_X$

(Send (X, "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Rcve (X, "MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) <

Send (X, "MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) <

Rcve (X, "MKH4", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2)) ..... (31)

ARP, HASH3, Θ_{MKHSH}

$[MKHSH : MA]_X$

Rcve (X, “MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2) ⊃

∃Z. Computes (Z, HASH $_{mptkX,T}$ (“MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ∧

Send (Z, HASH $_{mptkX,T}$ (“MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$))<

Rcve (X, “MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2) ..... (32)

$\Gamma_{MKHSH, SI, 1}$, HASH1

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Computes (Z, HASH $_{mptkX,T}$(“MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ⊃

Has (Z, $mptk_{X,T}$) ⊃ $\hat{Z}$=$\hat{X}$ ∨ $\hat{Z}$=$\hat{T}$ ..... (33)

(32), (33), AA1, $\Gamma_{MKHSH, 1}$, Θ_{MKHSH}

[MKHSH : MA]$_X$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Send (Z, HASH $_{mptkX,T}$ (“MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ⊃ $\hat{Z}$=$\hat{T}$ ..... (34)

(32), (34), Θ_{MKHSH}

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Computes (T, HASH $_{mptkX,T}$ (“MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ∧

Send (T, HASH $_{mptkX,T}$ (“MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ..... (35)

(35), ϕHONESTY, Θ_{MKHSH}

[MKHSH : MA]$_X$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Computes (T, HASH $_{mptkX,T}$ (“MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$)) ..... (36)

ARP, HASH3, Θ_{MKHSH}

Rcve (X, “MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) ⊃

∃Z. Computes (Z, HASH $_{mptkX,T}$ (“MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ∧

Send (Z, HASH $_{mptkX,T}$ (“MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$))<

Rcve (X, “MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) ..... (37)

$\Gamma_{MKHSH,\,1}$, HASH1

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Computes (Z, HASH $_{mptkX,T}$ (“MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ⊃

Has (Z, $mptk_{X,T}$) ⊃ $\hat{Z}$=$\hat{X}$ ∨ $\hat{Z}$=$\hat{T}$ ..... (38)

(37), (38), AA1, $\Gamma_{MKHSH,\,1}$, Θ_{MKHSH}

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Send (Z, HASH $_{mptkX,T}$ (“MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ⊃ $\hat{Z}$=$\hat{T}$ ..... (39)

(37), (39), Θ_{MKHSH}

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Computes (T, HASH $_{mptkX,T}$ (“MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ⊃

Send (T, HASH $_{mptkX,T}$ (“MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$)) ..... (40)

(35), (36), (40), HASH1, Θ_{MKHSH}

[MKHSH : MA]$_X$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃ Has (T, $mptk_{X,T}$) ∧ Has (T, x, t, $INFO_X$, $INFO_T$, $\hat{X}$, $\hat{T}$) ..... (41)

(32), (35), (41), ϕHONESTY, Θ_{MKHSH}

[MKHSH : MA]$_X$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Rcve (T, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Send (T, “MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) <

Rcve (T, “MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) <

Send (T, “MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2) ..... (42)

(32), (42), Θ_{MKHSH}

[MKHSH : MA]$_X$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Send (T, “MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2) <

Rcve (X, “MKH4”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2) ..... (43)

ARP, VER, Θ_{MKHSH}

$[MKHSH : MA]_X$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})∧

Rcve (T, “MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) ⊃

∃Z. Computes (Z, HASH $_{mptkX,T}$ (“MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1)) ∧

Send (Z, HASH $_{mptkX,T}$ (“MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1)) ∧

Send (Z, HASH $_{mptkX,T}$ (“MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1)) <

Rcve (T, “MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) ..... (44)

$\Gamma_{MKHSH, SI, 1}$, HASH1

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃ Computes (Z, HASH $_{mptkX,T}$ (“MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1)) ⊃

Has (Z, $mptk_{X,T}$) ⊃ $\hat{Z}$=$\hat{X}$ ∨ $\hat{Z}$=$\hat{T}$ ..... (45)

(44), (45), AA1, $\Gamma_{MKHSH, 1}$, Θ_{MKHSH}

$[MKHSH : MA]_X$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Send (Z, HASH $_{mptkX,T}$ (“MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1)) ⊃ $\hat{Z}$=$\hat{X}$ ..... (46)

(44), (46), Θ_{MKHSH}

$[MKHSH : MA]_X$

Send (X, “MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) < Rcve (T, “MKH3”, x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) ..... (47)

ARP, VER, Θ_{MKHSH}

$[MKHSH : MA]_X$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})∧ Rcve (T, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) ⊃

∃Z. Computes (Z, HASH $_{mptkX,T}$ (“MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$)) ∧

Send (Z, HASH $_{mptkX,T}$ (“MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$)) ∧

Send (Z, HASH $_{mptkX,T}$ (“MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$)) <

Rcve (T, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) ..... (48)

$\Gamma_{MKHSH, SI, 1}$, HASH1

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Computes (Z, HASH $_{mptkX,T}$ (“MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$)) ⊃

Has (Z, $mptk_{X,T}$) ⊃ $\hat{Z}=\hat{X} \vee \hat{Z}=\hat{T}$ ..... (49)

(48), (49), AA1, $\Gamma_{MKHSH, 1}$, Θ_{MKHSH}

[MKHSH : MA]$_X$

KOHonest ($mptk_{X,T}$ {$mkdk_{X,T}$})⊃

Send (Z, HASH $_{mptkX,T}$ (“MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$)) ⊃ $\hat{Z}=\hat{X}$ ..... (50)

(48), (50), Θ_{MKHSH}

[MKHSH : MA]$_X$

Send (X, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) < Rcve (T, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) ..... (51)

FS1, AN3, Θ_{MKHSH}

[MKHSH : MA]$_X$

FirstSend (T, t, “MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) ..... (52)

(42), (52), FS2, Θ_{MKHSH}

[MKHSH : MA]$_X$

Send (T, “MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) < Rcve (X, “MKH2”, x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) ..... (53)

FS1, AN3, Θ_{MKHSH}

[MKHSH : MA]$_X$

Honest ($\hat{X}$) ⊃ FirstSend (X, x, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) ..... (54)

(42), (54), FS2, Θ_{MKHSH}

[MKHSH : MA]$_X$

Honest ($\hat{X}$) ⊃ Send (X, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Rcve (T, “MKH1”, x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) ..... (55)

(42), (43), (53), (55), Θ_{MKHSH}

Send (X, "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Rcve (T, "MKH1", x, $\hat{X}$, $\hat{T}$, $INFO_X$, $hash_0$) <

Send (T, "MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) <

Rcve (X, "MKH2", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_0) <

Send (X, "MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) <

Rcve (T, "MKH3", x, t, $\hat{X}$, $\hat{T}$, $INFO_X$, mic_1) <

Send (T, "MKH4", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2) <

Rcve (X, "MKH4", x, t, $\hat{X}$, $\hat{T}$, $INFO_T$, mic_2) ..... (56)

B. Security proof of LCSNA

B.1 Matching conversations, Prover:

AA1, ARP, AA4, Θ_{CA}

$[CA : PROVER]_X$

(Send (*X,* "MSG1"*, r, s, t,* $\hat{X}$*,* $\hat{T}$*,* enc_0) <

Rcve (*X,* "MSG2"*,* $\hat{X}$*,* $\hat{T}$*,* enc_1) <

Send (*X,* "MSG3", $\hat{X}$*,* $\hat{T}$*,* enc_2) <

Rcve (*X,* "MSG4", $\hat{X}$*,* $\hat{T}$*,* enc_3)) ..... (57)

ARP, HASH3, Θ_{CA}

$[CA : PROVER]_X$

Rcve (*X,* "MSG4", $\hat{X}$*,* $\hat{T}$*,* enc_3) ⊃

∃Z. Computes (Z, $HASH_{K\lambda}$ ("MSG4", $\hat{X}$*,* $\hat{T}$*,* enc_3)) ∧

Send (Z, $HASH_{K\lambda}$ ("MSG4", $\hat{X}$*,* $\hat{T}$*,* enc_3)) <

Rcve (*X,* "MSG4", $\hat{X}$*,* $\hat{T}$*,* enc_3) ..... (58)

$\Gamma_{CA,\,2}$, HASH1

KOHonest (K_λ,*{ ENK}*)⊃

Computes (Z, $HASH_{K\lambda}$ ("MSG4", $\hat{X}$*,* $\hat{T}$*,* enc_3)) ⊃

Has (Z, K_λ) ⊃ $\hat{Z}$=$\hat{X}$ ⋁ $\hat{Z}$=$\hat{T}$ ..... (59)

(58), (59), AA1, $\Gamma_{CA,\,1}$, Θ_{CA}

$[CA : PROVER]_X$

KOHonest (K_λ,{*ENK*})$\supset$ Send (Z, HASH $_{K\lambda}$ (“MSG4”, $\hat{X}$, $\hat{T}$, enc_3)) $\supset \hat{Z}=\hat{T}$ ..... (60)

(58), (60), Θ_{CA}

KOHonest (K_λ,{*ENK*})$\supset$

Computes (T, HASH $_{K\lambda}$ (“MSG4”, $\hat{X}$, $\hat{T}$, enc_3)) $\wedge$ Send (T, HASH $_{K\lambda}$ (“MSG4”, $\hat{X}$, $\hat{T}$, enc_3)) ..... (61)

(61), ϕHONESTY, Θ_{CA}

$[CA : PROVER]_X$

KOHonest (K_λ,{*ENK*})$\supset$ Computes (T, HASH $_{K\lambda}$ (“MSG3”, $\hat{X}$, $\hat{T}$, enc_2)) ..... (62)

ARP, HASH3, Θ_{CA}

Rcve (X, “MSG2”, $\hat{X}$, $\hat{T}$, enc_1) $\supset$

$\exists$Z. Computes (Z, HASH $_{K\lambda}$ (“MSG2”, $\hat{X}$, $\hat{T}$, enc_1)) $\wedge$

Send (Z, HASH $_{K\lambda}$ (“MSG2”, $\hat{X}$, $\hat{T}$, enc_1)) < Rcve (X, HASH $_{K\lambda}$ (“MSG2”, $\hat{X}$, $\hat{T}$, enc_1)) ..... (63)

$\Gamma_{CA,\,1}$, HASH1

KOHonest (K_λ,{*ENK*})$\supset$

Computes (Z, HASH $_{K\lambda}$ (“MSG2”, $\hat{X}$, $\hat{T}$, enc_1)) $\supset$ Has (Z, K_λ) $\supset \hat{Z}=\hat{X} \vee \hat{Z}=\hat{T}$ ..... (64)

(63), (64), AA1, $\Gamma_{CA,\,1}$, Θ_{CA}

KOHonest (K_λ,{*ENK*})$\supset$ Send (Z, HASH $_{K\lambda}$ (“MSG2”, $\hat{X}$, $\hat{T}$, enc_1)) $\supset \hat{Z}=\hat{T}$ ..... (65)

(63), (65), Θ_{CA}

KOHonest (K_λ,{*ENK*})$\supset$

Computes (T, HASH $_{K\lambda}$ (“MSG2”, $\hat{X}$, $\hat{T}$, enc_1)) $\supset$ Send (T, HASH $_{K\lambda}$ (“MSG2”, $\hat{X}$, $\hat{T}$, enc_1)) ..... (66)

(61), (62), (66), HASH1, Θ_{CA}

$[CA : PROVER]_X$

KOHonest (K_λ,{*ENK*})$\supset$ Has (T, K_λ) $\wedge$ Has (T, r, s, t, ENK, $\hat{X}$, $\hat{T}$) ..... (67)

(58), (61), (67), ϕHONESTY, Θ_{CA}

$[CA : PROVER]_X$

KOHonest (K_λ,{*ENK*})⊃ Rcve (T, "MSG1",*r, s, t,* $\hat{X}$, $\hat{T}$, enc_0) <

Send (T, "MSG2", $\hat{X}$, $\hat{T}$, enc_1) <

Rcve (T, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) <

Send (T, "MSG4", $\hat{X}$, $\hat{T}$, enc_3) ..... (68)

(58), (68), Θ_{CA}

$[CA : PROVER]_X$

KOHonest (K_λ,{*ENK*})⊃ Send (T, "MSG4", $\hat{X}$, $\hat{T}$, enc_3) < Rcve (X, "MSG4", $\hat{X}$, $\hat{T}$, enc_3) ..... (69)

ARP, VER, Θ_{CA}

$[CA : PROVER]_X$

KOHonest (K_λ,{*ENK*}) ∧ Rcve (T, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) ⊃

∃Z. Computes (Z, HASH $_{K\lambda}$ ("MSG3", $\hat{X}$, $\hat{T}$, enc_2)) ∧

Send (Z, HASH $_{K\lambda}$ ("MSG3", $\hat{X}$, $\hat{T}$, enc_2)) < Rcve (T, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) ..... (70)

$\Gamma_{CA,\,2}$, HASH1

KOHonest (K_λ,{*ENK*})⊃

Computes (Z, HASH $_{K\lambda}$ ("MSG3", $\hat{X}$, $\hat{T}$, enc_2)) ⊃ Has (Z, K_λ) ⊃ $\hat{Z}$=$\hat{X}$ ⋁ $\hat{Z}$=$\hat{T}$ ..... (71)

(70), (71), AA1, $\Gamma_{CA,\,1}$, Θ_{CA}

$[CA : PROVER]_X$

KOHonest (K_λ, {*ENK*}) ⊃ Send (Z, HASH $_{K\lambda}$ ("MSG3", $\hat{X}$, $\hat{T}$, enc_2)) ⊃ $\hat{Z}$=$\hat{X}$ ..... (72)

(70), (72), Θ_{CA}

$[CA : PROVER]_X$

Send (X, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) < Rcve (T, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) ..... (73)

ARP, VER, Θ_{CA}

$[CA : PROVER]_X$

KOHonest (K_λ, {*ENK*}) ∧ Rcve (T, "MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_1) ⊃

$\exists$Z. Computes (Z, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$)) $\wedge$

Send (Z, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$)) < Rcve ("MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$) ..... (74)

$\Gamma_{CA,\,2}$, HASH1

KOHonest (K_λ, *{ENK}*) $\supset$

Computes (Z, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$)) $\supset$ Has (Z, K_λ) $\supset \hat{Z}=\hat{X} \vee \hat{Z}=\hat{T}$ ..... (75)

(74), (75), AA1, $\Gamma_{CA,\,1}$, Θ_{CA}

[CA : PROVER]$_X$

KOHonest (K_λ, *{ENK}*) $\supset$ Send (Z, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$)) $\supset \hat{Z}=\hat{X}$ ..... (76)

(74), (76), Θ_{CA}

[CA : PROVER]$_X$

Send (X, "MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$) < Rcve (T, "MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$) ..... (77)

FS1, AN3, Θ_{CA}

[CA : PROVER]$_X$

Honest ($\hat{X}$) $\supset$ FirstSend (X, r, s, t, "MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$) ..... (78)

(68), (78), FS2, Θ_{CA}

[CA : PROVER]$_X$

Honest ($\hat{X}$) $\supset$ Send (X, "MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$) < Rcve (T, "MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$) ..... (79)

(68), (69), (79), Θ_{CA}

Send (X, "MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$) <

Rcve (T, "MSG1", *r, s, t,* $\widehat{X}$*,* $\widehat{T}$*, enc*$_0$) <

Send (T, "MSG2", $\hat{X}$*,* $\hat{T}$*, enc*$_1$) <

Rcve (X, "MSG2", $\hat{X}$*,* $\hat{T}$*, enc*$_1$) <

Send (X, "MSG3", $\hat{X}$*,* $\hat{T}$*, enc*$_2$) <

Rcve (T, "MSG3", $\hat{X}$*,* $\hat{T}$*, enc*$_2$) <

Send (T, "MSG4", $\hat{X}$*,* $\hat{T}$*, enc*$_3$) <

Rcve (X, "MSG4", $\hat{X}$*,* $\hat{T}$*, enc*$_3$) ..... (80)

B.2 Matching conversations, Authenticator:

AA1, ARP, AA4, Θ_{CA}

$[CA : AUTH]_T$

(Rcve (T, "MSG1", $r, s, t, \hat{X}, \hat{T}, enc_0$) <

Send (T, "MSG2", $\hat{X}, \hat{T}, enc_1$) <

Rcve (T, "MSG3", $\hat{X}, \hat{T}, enc_2$) <

Send (T, "MSG4", $\hat{X}, \hat{T}, enc_3$)) ..... (81)

ARP, VER, Θ_{CA}

$[CA : AUTH]_T$

Rcve (T, "MSG3", $\hat{X}, \hat{T}, enc_2$) $\supset$

$\exists Z$. Computes (Z, HASH $_{K\lambda}$ ("MSG3", $\hat{X}, \hat{T}, enc_2$)) $\wedge$

Send (Z, HASH $_{K\lambda}$ ("MSG3", $\hat{X}, \hat{T}, enc_2$)) < Rcve (T, "MSG3", $\hat{X}, \hat{T}, enc_2$) ..... (82)

$\Gamma_{CA, 2}$, HASH1

KOHonest (K_λ, {ENK}) $\supset$

Computes (Z, HASH $_{K\lambda}$ ("MSG3", $\hat{X}, \hat{T}, enc_2$)) $\supset$Has (Z, K_λ) $\supset \hat{Z}=\hat{X} \vee \hat{Z}=\hat{T}$ ..... (83)

(82), (83), AA1, $\Gamma_{CA, 1}$, Θ_{CA}

$[CA : AUTH]_T$

KOHonest (K_λ, {ENK}) $\supset$Send (Z, HASH $_{K\lambda}$ ("MSG3", $\hat{X}, \hat{T}, enc_2$)) $\supset \hat{Z}=\hat{X}$ ..... (84)

(82), (84), Θ_{CA}

KOHonest (K_λ, {ENK}) $\supset$

Computes (X, HASH $_{K\lambda}$ ("MSG3", $\hat{X}, \hat{T}, enc_2$)) $\wedge$

Send (X, HASH $_{K\lambda}$ ("MSG3", $\hat{X}, \hat{T}, enc_2$)) ..... (85)

(85), ϕHONESTY, Θ_{CA}

$[CA : AUTH]_T$

KOHonest (K_λ, {ENK}) $\supset$

Computes (X, HASH $_{K\lambda}$ ("MSG2", $\hat{X}, \hat{T}, enc_1$)) ..... (86)

ARP, HASH3, Θ_{CA}

Rcve (T, "MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_0) $\supset$

$\exists$Z. Computes (Z, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_0)) $\wedge$

Send (Z, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_0)) ..... (87)

$\Gamma_{CA,\,1}$, HASH1

KOHonest (K_λ, *{ENK}*) $\supset$

Computes (Z, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_0)) $\supset$ Has (Z, K_λ) $\supset \hat{Z}=\hat{X} \vee \hat{Z}=\hat{T}$ ..... (88)

(87), (88), AA1, $\Gamma_{CA,\,1}$, Θ_{CA}

KOHonest (K_λ, *{ENK}*) $\supset$ Send (Z, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_0)) $\supset \hat{Z}=\hat{X}$ ..... (89)

(87), (89), Θ_{CA}

KOHonest (K_λ, *{ENK}*) $\supset$ Computes (X, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_0)) $\supset$

Send (X, HASH $_{K\lambda}$ ("MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_0)) ..... (90)

(85), (86), (90), HASH1, Θ_{CA}

$[CA : AUTH]_T$

KOHonest (K_λ, *{ENK}*) $\supset$ Has (X, K_λ) $\wedge$ Has (X, r, s, t, ENK, $\hat{X}$, $\hat{T}$) ..... (91)

(82), (85), (91), ϕHONESTY, Θ_{CA}

$[CA : AUTH]_T$

KOHonest (K_λ, *{ENK}*) $\supset$

Send (X, "MSG1", *r, s, t,* $\hat{X}$, $\hat{T}$, enc_0) < Rcve (X, "MSG2", $\hat{X}$, $\hat{T}$, enc_1) <

Send (X, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) ..... (92)

(82), (92), Θ_{CA}

$[CA : AUTH]_T$

KOHonest (K_λ, *{ENK}*) $\supset$ Send (X, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) < Rcve (T, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) ..... (93)

ARP, VER, Θ_{CA}

$[CA : AUTH]_T$

KOHonest (K_λ, *{ENK}*) ∧ Rcve (X, "MSG2", $\hat{X}$, $\hat{T}$, enc_1) ⊃

∃Z. Computes (Z, HASH $_{K\lambda}$ ("MSG2", $\hat{X}$, $\hat{T}$, enc_1)) ∧ Send (Z, HASH $_{K\lambda}$ ("MSG2", $\hat{X}$, $\hat{T}$, enc_1)) <

Rcve (X, "MSG2", $\hat{X}$, $\hat{T}$, enc_1) ..... (94)

$\Gamma_{CA,\,2}$, HASH1

KOHonest (K_λ, *{ENK}*) ⊃

Computes (Z, HASH $_{K\lambda}$ ("MSG2", $\hat{X}$, $\hat{T}$, enc_1)) ⊃ Has (Z, K_λ) ⊃ $\hat{Z}=\hat{X} \vee \hat{Z}=\hat{T}$ ..... (95)

(94), (95), AA1, $\Gamma_{CA,\,1}$, Θ_{CA}

[CA : AUTH]$_T$

KOHonest (K_λ, *{ENK}*) ⊃ Send (Z, HASH $_{K\lambda}$ ("MSG2", $\hat{X}$, $\hat{T}$, enc_1)) ⊃ $\hat{Z}=\hat{T}$ ..... (96)

(94), (96), Θ_{CA}

[CA : AUTH]$_T$

Send (T, "MSG2", $\hat{X}$, $\hat{T}$, enc_1) < Rcve (X, "MSG2", $\hat{X}$, $\hat{T}$, enc_1) ..... (97)

FS1, AN3, Θ_{CA}

[CA : AUTH]$_T$

Honest ($\hat{X}$) ⊃ FirstSend (X, r, s, t, "MSG1", *r, s, t,* $\widehat{X}$, $\hat{T}$, enc_0) ..... (99)

(92), (99), FS2, Θ_{CA}

[CA : AUTH]$_T$

Honest ($\hat{X}$) ⊃ Send (X, "MSG1", *r, s, t,* $\widehat{X}$, $\hat{T}$, enc_0) < Rcve (T, "MSG1", *r, s, t,* $\widehat{X}$, $\hat{T}$, enc_0) ..... (100)

(92), (93), (100), Θ_{CA}

Send (X, "MSG1", *r, s, t,* $\widehat{X}$, $\hat{T}$, enc_0) <

Rcve (T, "MSG1", *r, s, t,* $\widehat{X}$, $\hat{T}$, enc_0) <

Send (T, "MSG2", $\hat{X}$, $\hat{T}$, enc_1) <

Rcve (X, "MSG2", $\hat{X}$, $\hat{T}$, enc_1) <

Send (X, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) <

Rcve (T, "MSG3", $\hat{X}$, $\hat{T}$, enc_2) <

Send (T, "MSG4", $\hat{X}$, $\hat{T}$, enc_3) ..... (101)