SComm: A Real-Time Mutually Authenticated Secure Communication Framework for Smart Grids

Abubakar Sadiq Sani, Ke Meng, and Zhao Yang Dong
School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia
Email: [sadiq.sani, ke.meng]@unsw.edu.au; zydong@ieee.org

2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT) | 978-1-7281-8897-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISGT49243.2021.9372211

***Abstract*—Motivated by recent Denial of Service (DoS) attacks at the control center and multiple remote power generation sites of a registered entity in the U.S. power grid, we seek to address the lack of sufficient real-time mutual authentication and secure communication between smart grid components. We introduce SComm, a real-time mutually authenticated secure communication framework that consists of a commitment-based enrolment protocol and a mutually authenticated key establishment protocol by which components can authenticate each other and carry out secure communication to prevent DoS attacks. Our framework applies a Zero-Knowledge Elliptic Curve Diffie-Hellman (ZK-ECDH) to establish a unique cryptographic session key for secure communication. We analysed our framework with respect to its security and performance, and the results show that our framework enhances the security of components and communication in real-time to efficiently deal with unauthentication and DoS attack. As proof of concept, we apply our framework to mitigate the DoS attacks at the registered entity.**

***Index Terms*—Smart grid, mutual authentication, secure communication, security, privacy**

I. Introduction

Many smart grid components such as field devices rely on remote connectivity for communication. These components are becoming more vulnerable to Denial of Service (DoS) attacks due to insufficient real-time mutual authentication and secure communication. In the recent DoS attacks at the control center and multiple remote power generation sites of a registered entity in the U.S. power grid, an unauthenticated attacker exploited a vendor's firewall firmware vulnerability to reboot internet-facing firewalls that controlled communication between field devices and between the control center and sites and caused communication outages [1]. While a firmware update has been offered to address the vulnerability, deploying a preventive solution such as mutual authentication and secure communication framework (with time-based authentication encryption) has not been widely deployed to simultaneously prevent unauthentication and DoS attacks in the smart grids.

The essential smart grid security requirements such as identification, authentication, authenticity, integrity, confidentiality, and availability are used to protect operations and communications in the smart grid. However, many operations and communications vulnerabilities arising from the smart grid are as a result of the lack of integrating all these requirements. Thus, smart grid operations and communications require security reinforcements that satisfy all the security requirements to make them resilient against unprecedented DoS attacks.

DoS attacks strike the smart grid control components or infrastructure (such as the firewalls in [1]) but do not usually target controlled components (such as the field devices), which are managed by the control components. This kind of attack does not require deep knowledge of the controlled components. Thus, we argue that the increase in the interconnectivity of the control and controlled components has introduced security uncertainties in the smart grid. This is because an unauthenticated attacker can disrupt communications between the control and controlled components by exploiting only the control component and triggering it into inconsistent or dangerous states to cause operation and communication disruptions between the controlled components.

Motivated by the U.S. power grid event, we propose SComm, a real-time mutually authenticated secure communication framework where all components are securely enrolled and can authenticate and carry out secure communications with each other based on the smart grid security requirements. The secure enrolment is derived using the Elliptic-Curve Pederson Commitment (EC-PCS) [2] and the mutually authenticated secure communication is established using a cryptographic session key based on a Zero-Knowledge Elliptic Curve Diffie-Hellman (ZK-ECDH) (supported by the ECDH [3]) with time-based authenticated encryption. The main contributions of our paper are as follows: i) We propose a commitment-based enrolment protocol for enrolling the components to the smart grid – note that the components are enrolled by smart grid trusted security authorities, say enrolment and identity issuing authorities $EIIs$, which can be seen as trusted servers with synchronised distributed databases for maintaining similar information; ii) We propose a mutually authenticated key establishment protocol for deriving a cryptographic session key used for secure communication between the components; iii) We mitigate an active Man-In-The-Middle Impersonation (MITMI) attack on the ECDH – note that we prevent this attack to provide strong assurance on the security of communications between components during key establishment; iv) We provide a formal security verification and a performance analysis of our framework; and v) We mitigate the DoS attacks in the U.S. power grid.

II. Background

In this section, we provide the main underlying security techniques used in our framework.

A. Elliptic Curve Pedersen Commitment Scheme

The EC-PCS is an efficient implementation of the PCS [4] which uses secure Elliptic Curve Cryptography (ECC) that is based on the algebraic structure of elliptic curves over finite fields. The security of the EC-PCS is based on the hardness of solving the elliptic curve discrete logarithm problem (ECDLP). The operation of the EC-PCS are provided as follows:

1) Setup: Let F_p be a group of elliptic curve points, where p is a large prime. Let $G \in F_p$ be a random generator point of order n and $H \in F_p$ be a chosen generator point of order n so that it is computationally difficult to find $H = x_H.G$ except if the ECDLP is solved, where x_H is a random secret and G and H are chosen by an EII. Then, the EII publishes the elliptic curve domain parameters (p, a, b, G, H, n, h), where a and b are curve parameters and h is a cofactor.

2) Commit: The committer S chooses $r_S \in Z_p$ at random and creates a commitment C of $x_S \in Z_p$ by computing $C(x_S, r_S) = x_S.G + r_S.H$.

3) Reveal: To verify the authenticity of C, S reveals x_S and r_S and the verifier T checks if $C = x_S.G + r_S.H$.

The EC-PCS and PCS have two similar properties: i) Unconditionally hiding, i.e., every possible value of x is equally committed in C; and ii) Computationally binding, i.e., one cannot open C with any $x_S^{'}$ since $x_S^{'} \notin x_S$, unless the ECDLP can be solved by finding x_H in $H = x_H.G$.

B. Elliptic Curve Diffie-Hellman

The ECDH is a key agreement protocol based on an elliptic curve that uses the elliptic curve domain parameters. In the ECDH, S and T generate their own public and private key pair. The private key of S is a randomly chosen value d_S from $\{1, ..., n-1\}$ and the public key is computed by $Q_S = d_S.G$. Similarly, T has d_T as a private key and Q_T as a public key. The operations of the ECDH are provided as follows: i) S chooses a private key d_S, computes a public key $Q_S = d_S.G$, and sends Q_S to T; ii) T chooses d_T, computes $Q_T = d_T.G$, and sends Q_T to S; iii) S computes a shared secret key $k_{ST} = Q_T.d_S = d_T.G.d_S = Q_S.d_T$; and iv) T computes the same secret key $k_{TS} = Q_S.d_T = d_S.G.d_T = Q_T.d_S$. The key property of the ECDH is completeness, i.e., the protocol succeeds if S and T are honest.

III. The SComm Framework

In this section, we present the two elements of our framework, namely: i) commitment-based enrolment protocol; and ii) mutually authenticated key establishment protocol. Our framework involves two main entities as follows: (I) $EIIs$: these are honest entities that enrol and issue a unique cryptographic identity to every component in the smart grid. Note that: a) every EII maintains a database, which can be utilised by enrolled components for data storage (in an encrypted manner to prevent compromised attack) and verification purposes; and b) every EII also maintains a discrete clock that is consistent with other $EIIs'$ clocks and increments in rounds and can be utilised by the components for setting current time in the smart grid. (II) Components: these are entities

Commitment-based Enrolment Protocol between *S* and *EII*

1. S selects two random secrets $x_S, r_S \in Z_p$, computes a commitment $C = x_S.G + r_S.H$, randomly select a private key d_S, computes a public key $Q_S = d_S.G$, and sends C and Q_S to EII over a secure channel.

2. Upon receiving C and Q_S, EII verifies that S exists in the distributed database, randomly selects a private key d_{EII}, computes a public key $Q_{EII} = d_{EII}.G$, computes a preshared key k_p via $k_{p2} = Q_S.d_{EII} = Q_{EII}.d_S$ and then $k_p = Hash(k_{p2}, C)$, computes a cryptographic identity $ID_S = Sig_{d_{EII}}(Hash_{k_p}(C))$, stores ID_S in the distributed database, computes messages $M_1 = Enc_{k_p}(ID_S, Q_S, Q_{EII})$ and $M_2 = MAC_{k_p}(M_1)$, and sends M_1, M_2, Q_{EII}, and t_{info} to S over a secure channel, where t_{info} represents information about the discrete clock.

3. Upon receiving M_1, M_2, Q_{EII}, and t_{info}, S computes k_p via $k_{p2} = Q_S.d_{EII}$ and then $k_p = Hash(k_{p2}, C)$, verifies M_2 using $MAC_{k_p}(.)$'s verification part, i.e., $VMAC_{k_p}(M_2, M_1) = 1$?. If the verification succeeds, S decrypts M_1 uses $Enc_{k_p}(.)$'s decryption part, i.e., $Dec_{k_p}(M_1) = ID_S, Q_S, Q_{EII}$. Furthermore, S stores t_{info} to set its time in the smart grid.

Fig. 1. Commitment-Based Enrolment Protocol.

that are enrolled by any EII and can authenticate each other, establish a cryptographic session key, and carry out secure communication.

A. Commitment-based Enrolment Protocol

The commitment-based enrolment protocol is executed between an EII and every component. In this protocol, the EII assigns to a component a unique cryptographic identity that encodes the cryptographic commitment of the component. Unlike the EC-PCS where a committer (i.e., the component) reveals its random secrets of the commitment to the verifier (i.e., the EII), the component's secrets are not revealed to the EII in this protocol to prevent privacy attack, which targets leakage of sensitive information. To prevent an adversary from executing this protocol, the EII has a secure encrypted list of all smart grid components in its distributed database.

The commitment-based enrolment protocol between the EII and a component S is presented in Fig. 1. In this protocol, we utilise the EC-PCS and show that the cryptographic commitment C and identity of the component ID_S provide privacy of the component's secrets while maintaining the properties of the EC-PCS as mentioned in Section II. We apply the ECDH to compute a preshared key that supports message security. Note that: a) we use lightweight cryptographic algorithms such as 160 $bits$ ECDSA digital signature algorithm $Sig_{d_{EII}}(.)$, 160 $bits$ keyed-Hash Message Authentication Code (HMAC) $MAC_k(.)$, 128 $bits$ Advanced Encryption Standard (AES) $Enc_k(.)$, and 256 $bits$ Secure Hash Algorithm 2 (SHA-2) $Hash_k(.)$ to provide authentication/non-repudiation, authenticity, confidentiality, and integrity, respectively, where d_{EII} is a private key of the EII and k is a preshared key; b) we use the distributed database to verify S; c) we use ID_S to provide identification and authentication; and d) the protocol provides availability since it is executed between the EII and

S. Thus, the protocol supports all the essential smart grid security requirements listed in Section I.

B. Mutually Authenticated Key Establishment Protocol

The mutually authenticated key establishment protocol is executed between two enrolled components, say S and T with ID_S and ID_T, respectively. In this protocol, S and T use their respectively cryptographic identity to authenticate each other and establish a cryptographic session key based on ZK-ECDH, which mitigates active MITMI attack that is carried out on ECDH by an attacker or malicious component. In active MITMI attack, the adversary can intercept and alter key establishment request from an honest component and can further initiate a key establishment request to some other honest components by claiming the component's identity. This type of attack is possible because authentication and verifying the time interval of received messages before establishing a shared secret session key do not take place in the ECDH. The solution to this attack is to integrate a zero-knowledge mechanism and time-based authenticated encryption to the ECDH via ZK-ECDH which serves two main purposes, namely: i) the zero-knowledge mechanism allows components to use their cryptographic commitments and public keys to authenticate each other without revealing their secrets; ii) the time-based authenticated encryption, i.e., encrypt then authenticate with the support of timestamps, provides additional authentication and allows components to verify the time interval of messages before establishing a cryptographic session key.

The mutually authenticated key establishment protocol between S and T is presented in Fig. 2. If the ZK-ECDH and time-based authenticated encryption are successful, S and T use the derived key k to establish a secure communication session by encrypting then MACing exchanged messages. Thus, the protocol prevents an unauthenticated attacker from rebooting S or T, carrying out a DoS attack, or causing communication outages in the smart grid. Note that since cryptographic session keys established from key establishment protocols expire within a short period, the window for any security attack or compromise is very negligible.

Mutually Authenticated Key Establishment Protocol between S and T

1. S verifies if ID_T exists in the distributed database and selects two random secrets $u_S, v_S \in Z_p$ and computes a commitment $d_S = u_S.G + v_S.H$ if this verification succeeds. Then, S computes a private key $d_{S.2}$ and a public key $Q_{S.2} = d_{S.2}.G$, and sends $d_S, Q_{S.2}$, and ID_S to T.

2. Upon receiving $d_S, Q_{S.2}$, and ID_S, T checks that ID_S exists in the distributed database and computes a private key $d_{T.2}$ and a public key $Q_{T.2} = d_{T.2}.G$ if the verification succeeds. Otherwise, T ends execution of this protocol and wait for authentication requests from other components. Then, T computes a preshared key k_{TS} via $k_{TS2} = Q_{S.2}.d_{T.2} = Q_{T.2}.d_{S.2}$ and then $k_{TS} = Hash(k_{TS}, d_S)$, messages $M_3 = Enc_{k_{TS}}(ID_T, ID_S, Q_{T.2})$ and $M_4 = MAC_{k_{TS}}(M_3)$, and a timestamp t_1, and sends $M_3, M_4, Q_{T.2}$, and t_1 to S, where t_1 is a timestamp of M_4. Maximum time to perform this step is t_T.

3. Upon receiving $M_3, M_4, Q_{T.2}$, and t_1, S first verifies that $t_1 < t_T$. If the verification succeeds, S computes k_{ST} via $k_{ST2} = Q_{T.2}.d_{S.2} = Q_{S.2}.d_{T.2}$ and then $k_{ST} = Hash(k_{ST2}, d_S)$. Then, S verifies M_4 using $VMAC_{k_{ST}}(M_4, M_3) = 1$?. If this verification succeeds, S computes $Dec_{k_{ST}}(M_3) = ID_T, ID_S, Q_{T.2}$. Then, S computes two values $a_S = u_S.Q_{T.2} + x_S$ and $b_S = v_S.Q_{T.2} + r_S$, and sends messages $M_5 = Enc_{k_{ST}}(a_S, b_S, t_2)$ and $M_6 = MAC_{k_{ST}}(M_5)$ to T, where t_2 is a timestamp of M_5. Maximum time to compute M_5 is t_S.

4. Upon receiving M_5 and M_6, T verifies if $VMAC_{k_{ST}}(M_6, M_5) = 1$?. If the verification succeeds, T computes $Dec_{k_{ST}}(M_5) = a_S, b_S, t_2$ and verifies that $|t_2 - t_1| \leq t_S$, which provides time-based authenticated encryption since $VMAC_{k_{ST}}$ and $Dec_{k_{ST}}$ are supported by timestamps. If any of the verifications fail, T ends this protocol execution and wait for authentication requests from other components. If the verification of $|t_2 - t_1|$ succeeds, T computes a cryptographic session key k via $k_2 = (a_S.G + b_S.H).(k_{TS}, ID_S, ID_T)$ and then $k = Hash(k_2, ID_S, ID_T, t_2)$. Thus, k is derived based on the time-based authentication encryption.

5. S computes the session key k via $k_2 = d_S.Q_B + C.(k_{TS}, ID_S, ID_T) = (a_S.G + b_S.H).(k_{TS}, ID_S, ID_T)$ and then $k = Hash(k_2, ID_S, ID_T, t_2)$.

6. S and T can carry out secure communication using k.

Fig. 2. Mutually Authenticated Key Establishment Protocol.

IV. Security and Performance Analyses

In this section, we present the security properties, formal security verification, and performance analysis of SComm.

A. Security Properties

SComm is a framework with the following security properties: i) Secure, i.e., the success probability of a computationally bounded adversary to break the framework and cause either a DoS attack or an active MITMI attack is negligible; ii) Privacy-preserving, i.e., sensitive information about the identity, commitment, and session key are not leaked from the component and transcripts of the enrolment and key establishment protocols, and the EII does not learn any information about the communications between components; iii) Complete. i.e., honest components can successfully execute the key establishment protocol; and iv) Sound, i.e., key establishment protocol does not allow an honest component to prove a false statement or derive a false cryptographic session key.

B. Formal Security Verification

We first provide the formal security verification of SComm using widely-accepted AVISPA tool [5], which uses On-the-fly Model Checker (OFMC) and Constraints Logic Based Attacker Searcher (CL-AtSe) backends to find security attacks on SComm and follows the widely-accepted Dolev-Yao threat model [6], where components communicate with each other over a public channel. We use the High-Level Protocol Specification Language (HPLSL) [7] supported in AVISPA to implement SComm. We implement three basic roles: EII (as shown in Fig. 1) and components S and T (as shown in Fig. 2). The HPLSL implementations show that: i) EII and S as well as EII and T send and receive messages according to the enrolment protocol; and ii) S and T send and receive messages according to the key establishment protocol. The simulation results presented in Fig. 3 show that SComm is resilient against replay and Man-In-The-Middle (MITM) attacks.

OFMC Backend	CL-Atse Backend
% OFMC	SUMMARY
% Version of 2006/02/13	SAFE
SUMMARY	DETAILS
SAFE	BOUNDED_NUMBER_OF_SESSI
DETAILS	ONS
BOUNDED_NUMBER_OF_SESSIONS	TYPED_MODEL
PROTOCOL	PROTOCOL
/home/span/span/testsuite/results/SComm.	/home/span/span/testsuite/results/SC
if	omm.if
GOAL	GOAL
as_specified	As Specified
BACKEND	BACKEND
OFMC	CL-AtSe
COMMENTS	STATISTICS
STATISTICS	Analysed : 0 states
parseTime: 0.00s	Reachable : 0 states
searchTime: 1.10s	Translation: 0.01 seconds
visitedNodes: 3068 nodes	Computation: 0.00 seconds
depth: 10 plies	

Fig. 3. SComm Simulation results in AVISPA.

Finally, we perform an active MITMI attack and a DoS attack in our simulation. We first consider the setting where T cannot verify the authenticity of S. As expected, the key establishment protocol stopped execution to mitigate the active MITMI attack. Then, we consider another setting where S and T generated a cryptographic session key k and then T receives messages $M_7 = Enc_k(data)$ and $M_8 = MAC_k(M_7)$ and cannot verify that $VMAC_k(M_8, M_7) = 1$. In this case, SComm also stopped execution to mitigate DoS attack since $VMAC_k(M_8, M_7) \neq 1$ and then wait to receive geuine messages.

We compare the security of SComm with a recent existing related scheme, LAKA [8], in terms of active MITMI and DoS attacks. In LAKA, components rely on issued security tokens that are vulnerable to these attacks (see Sections I and III for more details of the attacks). Furthermore, registering the components at only one gateway provides a single point of failure in the scheme. In SComm, we introduce many $EIIs$ that can register or enrol components.

C. Performance Analysis

We analyse the performance of SComm based on computational and communication costs and then compare it with LAKA. We use the following notions for our performance analysis. T_{pm}, T_{rn}, T_{hf}, T_{hm}, T_{sig}, T_{vsig}, T_{se}, and T_{sd} denote the cryptographic operations required for elliptic curve point multiplication (160 $bits$), random number (32 $bits$), hash function (256 $bits$), keyed-hash MAC (160 $bits$), digital signature (160 $bits$), verifying digital signature, symmetric encryption (128 $bits$), and symmetric decryption, respectively. The computational costs required for our commitment-based enrolment and mutually authenticated key establishment protocols in SComm are $2T_{rn} + 3T_{pm} + 2T_{hf} + T_{sig} + T_{vsig} + T_{se} + T_{sd} + 2T_{hm} \approx 1,952$ $bits$ and $7T_{rn} + 3T_{pm} + 4T_{hf} + 2T_{se} + 2T_{sd} + 4T_{hm} \approx 2,266$ $bits$, respectively. Furthermore, the communication cost required for the enrolment protocol is as follows: $S \rightarrow EII = (C, Q_S) = 160 + 160 = 320$ $bits$ and $EII \rightarrow S = (M_1, M_2, Q_{EII}, t_{info}) = 608 + 160 + 160 + 32 = 960$ $bits$, where $M_1 = Enc_{k_p}(ID_S, Q_S, Q_{EII})$ and $M_2 = MAC_{k_p}(M_1)$, thus the total communication bits required is $1,280$ $bits$ for six (6) messages. On the other hand, the communication cost required for the key establishment protocol is as follows: $S \rightarrow T = (d_S, Q_{S.2}, ID_S) = 160 + 160 + 160 = 480$ $bits$, $T \rightarrow S = (M_3, M_4, Q_{T.2}, t_1) = 608 + 256 + 160 + 32 = 1,056$ $bits$, and $S \rightarrow T = (M_5, M_6) = 288 + 256 = 544$ $bits$, where $M_3 = Enc_{k_{TS}}(ID_T, ID_S, Q_{T.2})$, $M_4 = MAC_{k_{TS}}(M_3)$, $M_5 = Enc_{k_{ST}}(a_S, b_S, t_2)$, and $M_6 = MAC_{k_{ST}}(M_5)$, thus the total communication bits required is $2,080$ $bits$ for nine (9) messages. We compare the computational and communication costs of the key establishment protocol with LAKA, which requires computational and communication costs of $6T_{pm} + 9T_{hf} + 4T_{hm} + 2T_{se} + 2T_{sd} \approx 4,416$ $bits$ and $\approx 2,368$ $bits$ (for twelve (12) messages), respectively for authentication and key agreement. It can be observed that SComm requires less computational and communication costs compared to LAKA due to the use of large random numbers and a reduced number of messages, respectively. Without loss of generality, we leave an investigation of scalability and latency of SComm for future work.

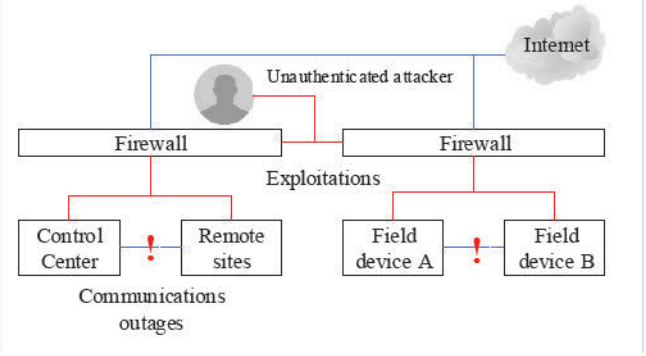

Fig. 4. A simple description of the recent DoS attack in the U.S. power grid.

V. Case Study

In this section, we carry out a case study on the recent issue at the registered entity [1] to demonstrate the usefulness of SComm. We analyse mutual authentication and secure communication between the internet-facing firewalls and field devices at the entity's control center and multiple remote generation sites. The firewalls are meant to provide authentication before access is granted, however, an unauthenticated attacker successfully exploited the firewalls and caused communication outages between the field devices and between the sites and control center. This shows that the firewalls, field devices, control center, and sites are not capable of providing sufficient real-time mutual authentication and secure communication.

A simple description of the issue is depicted in Fig. 4. This figure shows that the field devices, control center, and sites rely on the firewalls to mitigate security attacks and they do not meet the security properties of SComm. To see this, consider the following setting: S (i.e., the firewall) sends a message to T (i.e., the field device). T might have received a different message from another field device. Thus, we have no security

guarantees for the message and an attacker can (i) let T accept the message since S and T do not authenticate each other, and (ii) cause DoS attacks between S and T since no shared secret session key is established for secure communication between the components. Thus, the security capabilities of the firewall are not adequate to prevent unauthentication and DoS attacks against the firewall and field device that is controlled by the firewall.

To fix this problem, we equip all the components with SComm and show that SComm-supported entity provides real-time mutual authentication and secure communication to prevent unauthentication and DoS attacks. We prove the security of the SComm-supported entity by showing that a computationally bounded adversary's success probability to break SComm and carry out DoS attacks is negligible.

Theorem. *Let $RandValues$ be a random oracle for deriving two random secrets (x, r) used in an SComm-supported entity and there exists a computationally bounded adversary I that successfully authenticates and establishes a cryptographic session key with the key establishment protocol in SComm, then I can break the protocol and cause DoS attacks by solving the discrete logarithm problem with overwhelming probability.*

Proof Sketch: Assume such an adversary I exists, and an adversary J is given $(G, H, x_S.G, r_S.H)$. J is expected to send an encrypted message $M_3 = Enc_{k_{IJ}}(ID_J, ID_I, Q_J)$ and MACed message $M_4 = MAC_{k_{IJ}}(M_3)$ to I. Then, I is expected to output $a_I = u_I.Q_{J.2} + x_I$ and $b_I = v_I.Q_{J.2} + r_I$. To simulate $RandValues$, J selects two random values and add them to a set of tuples of random secrets, Set A, in the distributed database. When I queries $RandValues$ on a value y, J does the following. J checks if there exists a tuple (y, z) in Set A and then returns z. Otherwise, it chooses a random value $z^{'}$, adds $(y, z^{'})$ to Set A, and returns $z^{'}$ to I. We say that if I succeeds, then J always succeeds and can use y and $z^{'}$ to establish a cryptographic session key for secure communication with non-negligible probability. However, assuming the ECDLP is hard, then I does not exist.

We conclude that the SComm-supported entity prevents DoS attacks. We leave a detailed proof of the privacy of the SComm-supported entity for future work due to page limit.

VI. Related Work

Many cryptographic schemes for secure communication have been widely investigated in the smart grid [8], [9], [10]. We consider identification, authentication, authenticity, integrity, confidentiality, and availability as essential smart grid security requirements, but these requirements have not been widely integrated and applied to mitigate real-time unauthentication and DoS attacks in the smart grids. For example, Kumar et al. [8] uses ECC, symmetric encryption, hash function, and MAC algorithms to design a lightweight authentication and key agreement scheme in the smart grid. However, the scheme relies on a security token that is assigned to every device and further stored in the device's memory thereby leading to lack of security token secrecy, extra computational cost, active MITMI attack, compromised attack, and privacy attack. Furthermore, once the device is compromised and sensitive information have been leaked, an attacker can cause unauthentication and DoS attacks.

Recently, internet-facing firewalls of a registered entity in the U.S. power grid were exploited by an unauthenticated attacker to cause DoS attacks [1]. We argue that the key vulnerabilities at the entity are lack of sufficient real-time mutual authentication and secure communication between its components. In this paper, we introduce SComm to mitigate the vulnerabilities by providing real-time mutually authenticated secure communication between components.

VII. Conclusion and Future Work

We have presented SComm, a real-time mutually authenticated secure communication framework in which components can authenticate with each other using their cryptographic identities and further establish a cryptographic session key for secure communication and data exchange in the smart grid. SComm comprises of a commitment-based enrolment protocol and a mutually authenticated key establishment protocol, and focuses on preventing unauthentication and DoS attacks in the smart grids and overcoming the existing active MITMI attack on the ECDH. We analysed the security and performance of SComm and further compared the results with an existing related scheme. We illustrated the usefulness of SComm in a case study, which shows that the components of a registered entity in the U.S. power grid are not capable of providing sufficient real-time mutual authentication and secure communication, and we implemented SComm to overcome these shortcomings and prevent unauthentication and DoS attacks on the entity. In future work, we plan to extend SComm to capture access control, develop a prototype implementation of SComm, and apply SComm to real-world communication protocols in the smart grids.

References

[1] NERC. Lesson Learned: Risks Posed by Firewall Firmware Vulnerabilities. [Online]. Available: https://www.nerc.com/pa/rrm/ea/Pages/Lessons-Learned.aspx

[2] B. França, "Homomorphic mini-blockchain scheme," ed: April, 2015

[3] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," Designs, codes and cryptography, vol. 19, pp. 173-193, 2000.

[4] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in Annual International Cryptology Conference, 1991, pp. 129-140.

[5] AVISPA. Automation Validation of Internet Security Protocols and Applications. [Online]. Available: http://www.avispa-project.org/.

[6] D. Dolev and A. Yao, "On the security of public key protocols," IEEE Transactions on Information Theory, vol. 29, pp. 198-208, 1983.

[7] D. Von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in Proceedings of APPSEM 2005 workshop, 2005, pp. 1-17.

[8] P. Kumar, A. Gurtov, M. Sain, A. Martin, and P. H. Ha, "Lightweight authentication and key agreement for smart metering in smart energy networks," IEEE Transactions on Smart Grid, 2018.

[9] A. Mohammadali, M. Sayad Haghighi, M. H. Tadayon, and A. Mohammadi Nodooshan, "A Novel Identity-Based Key Establishment Method for Advanced Metering Infrastructure in Smart Grid," IEEE Transactions on Smart Grid, pp. 1-1, 2016.

[10] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," IEEE Transactions on Smart Grid, vol. 9, pp. 1900-1910, 2016.