energies

MDPI

Article

Efficient and Provably Secure Key Agreement for Modern Smart Metering Communications

An Braeken [1,*,†], Pardeep Kumar [2,†] and Andrew Martin [2]

1 Industrial Engineering INDI, Vrije Universiteit Brussel, 1050 Brussels, Belgium
2 Department of Computer Science, Oxford University, Oxford OX1 3QD, UK; pardeep.kumar@cs.ox.ac.uk (P.K.); andrew.martin@cs.ox.ac.uk (A.M.)
* Correspondence: an.braeken@vub.ac.be
† These authors contributed equally to this work.

Received: 18 September 2018; Accepted: 1 October 2018; Published: 6 October 2018

Abstract: Security in modern smart metering communications and in smart grid networks has been an area of interest recently. In this field, identity-based mutual authentication including credential privacy without active involvement of a trusted third party is an important building block for smart grid technology. Recently, several schemes have been proposed for the smart grid with various security features (e.g., mutual authentication and key agreement). Moreover, these schemes are said to offer session key security under the widely accepted Canetti-Krawczyk (CK) security model. Instead, we argue that all of them are still vulnerable under the CK model. To remedy the problem, we present a new provably secure key agreement model for smart metering communications. The proposed model preserves the security features and provides more resistance against a denial of service attack. Moreover, our scheme is pairing-free, resulting in highly efficient computational and communication efforts.

Keywords: smart metering network; authentication; canetti-krawczyk; ECQV certificates; anonymity

1. Introduction

Modern smart metering networks are one of the stepping stones in the evolution of smart grid (SG) networks and in renewable and distributed energy resource management systems. Unlike traditional metering systems, the infrastructure of modern smart metering relies typically on advanced information communication technologies (ICT). Smart metering networks will provide new opportunities for better control and management on energy production/consumption. Therefore, the future energy grid can operate, control and manage more efficiently and reliably.

In a smart metering network, a smart meter (SM) is an important enabler that utilises two-way communication to send/receive consumption data or commands from a service provider (SP) and then to various entities, e.g., energy supplier, distribution network operator (DNO) and others, as shown in Figure 1. Therefore, the increasing deployment of SMs in homes will provide many notable features, e.g., monitoring and control of energy consumption within the home area network, and communicating energy pricing information to consumers [1].

Smart metering provides various notable features, but it also poses various security and privacy challenges. For instance, recently three power companies experienced a massive cyberattack that disrupted the smooth functionality of the power grid and resulted in an energy blackout in a region for several hours [2]. Such verified attack incidents serve as prominent demonstrations of the significant risks to utility companies. Moreover, to meet the demands of energy consumption and generation, both customers and utility SPs need to exchange two-way information. In this scenario, an adversary can easily tamper with or capture the flow of information (i.e., wireless packets), which may affect

various use-cases, e.g., imbalance demand and supply systems, revenue losses, etc. Moreover, SMs can be captured physically as they are installed outside homes. As a SM usually collects and stores energy usages in its memory, the attacker can easily compromise a SM and dig out the information stored in it. This stored information may compromise the privacy of an individual, i.e., daily habits, sleeping patterns, and other activities. Moreover, as suggested in recent papers, under the Canetti-Krawczyk (CK) model [3,4], an attacker can even use the credentials that are stored in a meter and can impersonate other non-compromised entities. Therefore, security and privacy issues have been the main barriers recently in the success of smart metering and resistance.

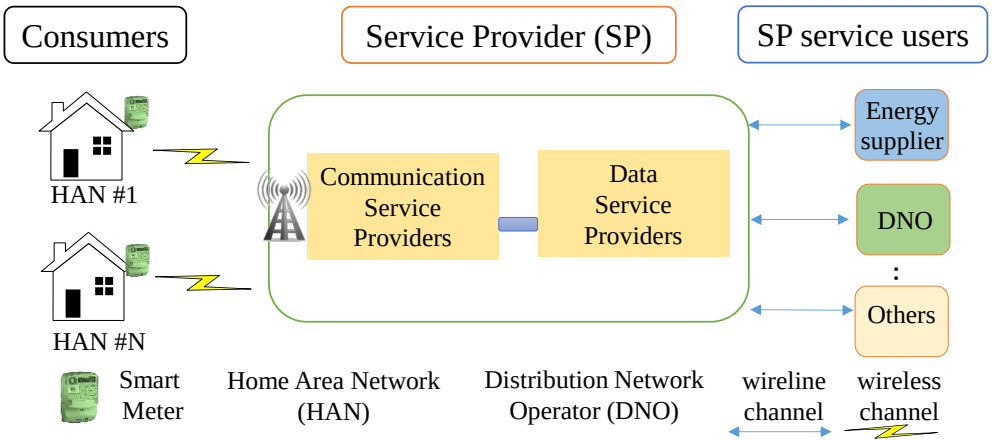

Figure 1. Smart Metering Architecture.

1.1. Related Work

To address security and privacy issues in smart metering networks, cryptography-based solutions play a significant role providing authentication, integrity and confidentiality of flow of information between the SM and utility SPs. Recently, several authentication and key agreement schemes have been proposed in SGs.

In [5,6], the authors proposed a secure key management and key distribution scheme, respectively, in the SG network. The authors utilized the involvement of a trusted third party (TTP) during the key negotiation phase and symmetric key operations between the SM and SP. This feature may not be always efficient as it requires the TTP to be online and enlarges the attack scope. Moreover, these schemes do not offer anonymity and are vulnerable to man-in-the-middle (MITM) attacks and impersonation attacks.

Considering public key infrastructure, Mahmood et al. and Mohammadali et al., proposed lightweight message authentication schemes for the SG in [7,8], respectively. In particular, Mahmood et al. [7] formally prove the resistance of the scheme in the CK security model. Unfortunately, both schemes do not provide anonymity of the SM's credentials.

In 2016, Tsai-Lo proposed the first anonymous key distribution scheme for SG communications [9]. The authors utilised an identity-based signature to achieve authentication and anonymity at low computational cost. Unfortunately, in [10], Odelu et al. reported that Tsai-Lo's scheme provides weak security to the session key and that leads to many other security attacks. Then in [10], the authors proposed another authentication and session key agreement scheme for SGs. The authors asserted that their scheme required low computational cost and provided security of the session keys in the CK security model. However, in [11], Chen et al. claimed that the registration phase of the scheme proposed in [10] may be vulnerable and may lead to attacks. They also proposed a new scheme, with session key security in the CK security model, but with a different registration phase. In [12], another anonymous key agreement scheme was proposed, which is claimed to be resistant in the CK security model. This scheme does not use computing-intensive pairing operations and is limited to elliptic curve multiplications and additions. We will show that the schemes [10–12], claiming to be secure

in the CK security model, do not satisfy this security feature. The advantage of resistance in the CK security model is that the scheme then also offers security attributes such as perfect forward secrecy, loss of information, known-key security, key-compromise impersonation, and unknown key-share, as proven in [13]. In addition, we also discuss their weakness with respect to resilience to denial of service (DoS).

In [14], Gope-Sikdar proposed an authenticated key agreement scheme for securing SG networks in the CK security model. The authors utilized a physically unclonable function (PUF) and claimed that their scheme can provide robust security against MITM attacks, and offers resilience to DoS. However, the PUF uses a fuzzy extractor, which has limitations as reported in [15].

Consequently, most of the proposed schemes are either vulnerable to security attacks or require high computation costs at resource-constrained SMs.

1.2. Contributions

First, we show how three recently proposed schemes [10–12] for authentication in the SG domain, aiming to establish session key security in the CK model, are still vulnerable. In addition, these schemes may suffer a MITM attack, and that may lead to a DoS attack. Moreover, the scheme proposed in [10] also experiences the key escrow problem.

Second, we propose a pairing-free scheme, able to establish the security features of [10–12], having identity-based mutual authentication, credential privacy, session key security, and resistance under the CK adversary model. Since the operations are limited to elliptic curve multiplication and addition, hashes and symmetric key encryption operations, both computational and communication performance largely outperforms the state of the art. In addition, the proposed scheme does not require a secure channel during the registration of the entities and is resilient to key escrow due to the usage of Elliptic Curve Qu Vanstone (ECQV) certificates. Moreover, the success of DoS attacks at the side of the SP is largely avoided as the key is derived in one single phase from the side of the SP.

Third, we formally prove the security of the scheme under the CK adversary model and random oracle model. The security verification simulations are performed using the AVISPA software tool [16]. In addition we compare the computational and communication cost with the other recently proposed anonymous, identity-based, mutual authentication schemes.

1.3. Outline

The paper is organised as follows. Section 2 deals with preliminaries. We show the weaknesses of the security scheme of [10–12] in Section 3. In Section 4, the proposed scheme is described. In Section 5, we give a formal proof of the security in the CK model. The computational complexity and the communication cost is explained in Section 6. Finally, conclusions are drawn in Section 7.

2. Preliminaries

We start with some background on bilinear pairings as it is required to understand the schemes and the attacks on [10–12]. Next, the CK security model is further elaborated. We also discuss in more detail the ECQV certificate scheme as it is an important building block in the registration phase of our proposed scheme and also used in [11].

2.1. Bilinear Pairings

Denote the additive cyclic group by G_1 and the multiplicative group by G_2, with both having high prime order q. Let P be a generator of G_1. Then, the bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$ should satisfy the followings:

- Bilinear: Given $P_1, P_2, Q, Q_2 \in G_1$, then $e(P_1 + P_2, Q_1) = e(P_1, Q_1)e(P_2, Q_1)$, $e(P_1, Q_1 + Q_2) = e(P_1, Q_1)e(P_1, Q_2)$ and $e(aP_1, bQ_1) = e(abP_1, Q_1) = e(P_1, abQ_1) = e(bP_1, aQ_1) = e(P_1, Q_1)^{ab}$ for any $a, b \in Z_q^*$.

- Nondegenerate: There exist $P, Q \in G_1$, such that $e(P, Q) \neq 1$, with 1 the identity element of G_2.
- Computable: For any $P, Q \in G_1$, the value $e(P, Q)$ is efficiently computed.

The following related mathematical problems are considered.

- The Elliptic Curve Discrete Logarithm Problem (ECDLP). This problem states that given two points R and Q of an additive group G, generated by an elliptic curve (EC) of order q, it is computationally hard for any polynomial-time bounded algorithm to determine a parameter $x \in Z_q^*$, such that $Q = xR$.
- The Elliptic Curve Diffie Hellman Problem (ECDHP). Given two points $R = xP, Q = yP$ of an additive group G, generated by an EC of order q with two unknown parameters $x, y \in Z_q^*$, it is computationally hard for any polynomial-time bounded algorithm to determine the EC point xyP.

2.2. Smart Metering Network Model

Following Figure 1, in the HAN, a SM collects the consumption data usage from the appliances within a home and then it sends the collected data to the SP, periodically. To send energy usage data, a SM can utilize IEEE 802.15.4 (e.g., ZigBee/Zwave), IEEE 802.11 (e.g., WiFi), and/or powerline communication technologies, as discussed in [17,18].

The SP aggregates consumption usage data from several SMs within the neighbouring area network (NAN), and provides data services to many other stakeholders, e.g., energy suppliers, DNO, etc. The SP can communicate to the other stakeholder via WiMAX, 4G/5G, etc. Note that the main focus of this research is to establish a secure communication between the SM and the SP.

2.3. Thread Model

Following the scheme proposed in [10], we assume the CK adversary model, as presented in [3], where the adversary cannot only eavesdrop on the channel or actively manipulate (insert, change, replay) the transmitted messages, but he/she can reveal session state specific information, session keys, or long-term private keys. For details, the interested readers may refer to [3,10]. The session specific information is defined as the local state of the session and its subroutines. Access is limited to either the locally stored information (local variables) or the long-term secret information, not the combination of both.

The ultimate goal of the attacker is to derive the common shared key.

2.4. ECQV Certificates

The ECQV certificate scheme [19] is a very efficient mechanism to construct a secret key pair together with a certificate for an entity in the scheme without the need for a secure channel between the TTP and the entity to share material for the generation of its secret private key. As a consequence, the TTP is also not able to derive the private key of the entity and so there are no key escrow problems. Its security has been formally proven in [20]. The ECQV scheme works as follows for an entity A requesting the generation of its secret key pair and corresponding certificate with the TTP.

Consider the curve $E_{p(a,b)}$ in Z_p with generator point P or order q. Denote the private and public key of the TTP by (k, P_{pub}) with $P_{pub} = kP$. Define the hash function $H_0 : \{0,1\}^* \rightarrow Z_q^*$. First the entity A with identity ID_A chooses a random value $r_A \in Z_p^*$ and computes $R_A = r_A P$. The message ID_A, R_A is sent to the TTP. Here, the TTP also selects a random value $r_T \in_R Z_p^*$ and computes $R_T = r_T P$. Next, it computes $cert_A = R_A + R_T$, and $r = H_0(cert_A \| ID_A) r_T + k$. The values $(cert_A, r)$ are sent to A over a public channel. Using these values, A now computes its private key as $d_A = H_0(cert_A \| ID_A) r_A + r$. It accepts the registration if its public key P_A satisfies the following equality

$$P_A = d_A P \quad = \quad H_0(cert_A \| ID_A) cert_A + P_{pub} \tag{1}$$

Consequently, given $ID_A, cert_A$, any legal entity is able to construct the corresponding public key of A using Equation (1). Thanks to the usage of the certificate, the other entity is assured of the relation between identity and public key.

3. Security Analysis of Authentication Schemes Recently Proposed in Literature

Based on weaknesses in the registration phase and key generation in [10], as also discussed later, Chen et al. [11] proposed a slightly different scheme. We argue that both schemes do not offer the required strength in the CK security model. In addition, we also describe some more practical issues with the registration phase in [10] and the resistance against DoS attacks. This last type of attack is also applicable for the scheme of [11].

3.1. Review of Odelu et al.

Description of the scheme [10]:

TTP setup phase: The TTP first chooses a bilinear map $e : G_1 \times G_1 \rightarrow G_2$, with G_1, G_2 be a cyclic additive and multiplicative group respectively, both of order q, and P a generator of G_1. Also five hash functions are identified: $H_1 : \{0,1\}^* \rightarrow Z_q^*, H_2 : G_2 \times \{0,1\}^* \rightarrow \{0,1\}^m, H_3 : \{0,1\}^* \rightarrow \{0,1\}^n$, $H_4 : G_2 \rightarrow Z_q^*$, and $H_5 : \{0,1\}^* \times G_1 \rightarrow Z_q^*$, with $m = n + w$ and w a constant determined by the security level.

Then the private key k is chosen and its corresponding public key is computed $P_{pub} = kP$. These system parameters are published by the TTP.

Registration phase: In this phase, both the SM and the SP need to undergo a different process.

For the SM, after receiving its identity ID_A, the TTP chooses a random number $r_A \in Z_q^*$ and computes $R_A = r_A P$ and $d_A = H_5(ID_A \| R_A)k + r_A$. The pair (d_A, R_A) is sent over a secure channel to the SM. The SM stores in its tamper-proof module the pair (d_A, R_A).

For the SP with identity ID_B, the TTP computes the private key as $d_B = \frac{1}{k+H_1(ID_B)}$ and sends d_B also over a secure channel. Denote $hd_b = H(ID_B \| d_B)$, then the SP stores the pair (d_b, hd_b) in its tamper-proof module.

Key negotiation phase: The steps in the key negotiation phase to derive the session key SK are summarised in Figure 2.

Weaknesses in [10]: Now we discuss the weaknesses of Odelu et al.'s scheme, as follows.

(A) Security in CK model: In the CK security model, a secure session reveal on the SM and SP, i.e., SSReveal(SM) and SSReveal(SP), is possible.
We can assume the leakage of $r_1 + d_A$ as it represents a local session state at the side of the SM. It is a local variable needed to be stored in order to be reused after reception of MSG_2. Similarly, we can assume $r_2 + hd_B$ as a local session state at the side of the SP. Again, this variable needs to be stored as it is used two times in the calculations of the SP. Since, $SK = H_4(g^{(r_1+d_A)(r_2+hd_B)})$, thus only depends on these two local session states, the session key can be retrieved and the scheme turns out to be vulnerable under the CK security model.

(B) Practical issues with registration phase: The generation of key material for both the SM and the SP from the TTP is not resistant against an honest but curious TTP, as the TTP derives the key material completely on its own and pushes it to the SM and SP. Consequently, the TTP can follow all the operations, and is able to track the SMs and SPs communications, and can collect the information.

Moreover, in the registration phase, a secure channel is required between the SM, SP and TTP. Secure channels are often difficult to establish, especially in the case of SMs, where they are mostly in practice established through physical contact. Therefore, secure channels are not always practical.

Finally, we also mention the weakness already noted by [11] regarding the key escrow problem. Once the private key of the TTP is revealed, all private keys of the SPs and SMs are easily derived. Once these private keys are revealed, impersonation attacks and many more attacks can start.

(C) Late detection of MITM: Lead to DoS: The scheme behaves very weak with regard to MITM attacks that may lead to DoS attacks on the server side. Assume that an attacker (Tom) acts as MITM. Tom can capture MSG_1 and sends it ($MSG_{1_{Tom}}$) to the SP. Note that the SP can only decide about the validity of a request ($MSG_{1_{Tom}}$) after completing the whole process, i.e., after receiving the message MSG_3. As a consequence, each request opens a buffer, where first several compute intensive pairings need to be computed, followed by the submission of a response. This buffer needs to be kept open until a response of the SM is received. Consequently, the memory can easily overflow by sending a huge number of invalid requests, where valid and invalid requests cannot be distinguished due to the late detection of forged messages.

SM		SP
Choose $r_1, n_1 \in_R Z_q^*$		
Compute $T_1 = (r_1 + d_A)(H_1(ID_B)P + P_{pub})$		
$g_1 = g^{r_1+d_A}$		
$C_1 = H_2(g_1 \| ID_B) \oplus (ID_A \| R_A \| n_1)$		
$A_1 = H_3(T_1 \| ID_A \| R_A \| n_1 \| g_1)$		
	$\xrightarrow{MSG_1=\{T_1,C_1,A_1\}}$	
		Compute $g_1 = e(T_1, d_B)$
		$(ID_A \| R_A \| n_1) = C_1 \oplus H_2(g_1 \| ID_B)$
		Check if $A_1 = H_3(T_1 \| ID_A \| R_A \| n_1 \| g_1)$
		Accept/Reject
		Choose $r_2 \in_R Z_q^*$
		$g_2 = e((r_2 + hd_B)P, H_5(ID_A \| R_A)P_{pub} + R_A)$
		$SK = H_4(g_1^{r_2+hd_B})$
		$A_2 = H_3(SK \| g_2 \| ID_B \| ID_A \| n_1 \| g_1)$
	$\xleftarrow{MSG_2=\{g_2,A_2\}}$	
Compute $SK = H_4(g_2^{(r_1+d_A)/d_A})$		
Check if $A_2 = H_3(SK \| g_2 \| ID_B \| ID_A \| n_1 \| g_1)$		
Accept/Reject		
Compute $A_3 = H_3(SK \| ID_A \| n_1 \| g_1 \| g_2 \| ID_B)$		
	$\xrightarrow{MSG_3=\{A_3\}}$	
		Check $A_3 = H_3(SK \| ID_A \| n_1 \| g_1 \| g_2 \| ID_B)$

Figure 2. Steps and computations in the key agreement phase of [10].

3.2. Review of Chen et al.

Description of the scheme [11]:

TTP setup phase: The setup phase is the same as in [10].

Registration phase: In this phase, both the SM and the SP are able to construct their key material according to the ECQV implicit certificate mechanism. As a result, the key pairs of SM and SP equal to (d_A, P_A) and (d_B, P_B) respectively.

Key negotiation phase: The steps in the key negotiation phase are summarised in Figure 3.

Weaknesses in [11]: Now we discuss the weaknesses of Chen et al.'s scheme, as follows.

(A) Security in CK model: A similar type of attack as in [10] is possible. In the CK security model, a secure session reveal on the SM and SP, i.e., SSReveal(SM) and SSReveal(SP), is possible. As a result, the local states $r_1 + d_A$ from SM and $r_2 + d_B$ from SP are leaked. Since, $SK = H_4((r_1 + d_A)(r_2 + d_B)P)$, the security of the session key is broken.

(B) Late detection of MITM: Lead to DoS: The same issue regarding the late detection of a MITM attack also holds in this scheme. After receiving MSG_1, the SP is unable to check if this request is coming from a registered SM. Only after receiving the last message MSG_3 and verifying the hash

value A_3, the SP can decide if the key negotiation has been successfully executed. As a result, this weakness could lead to DoS threat.

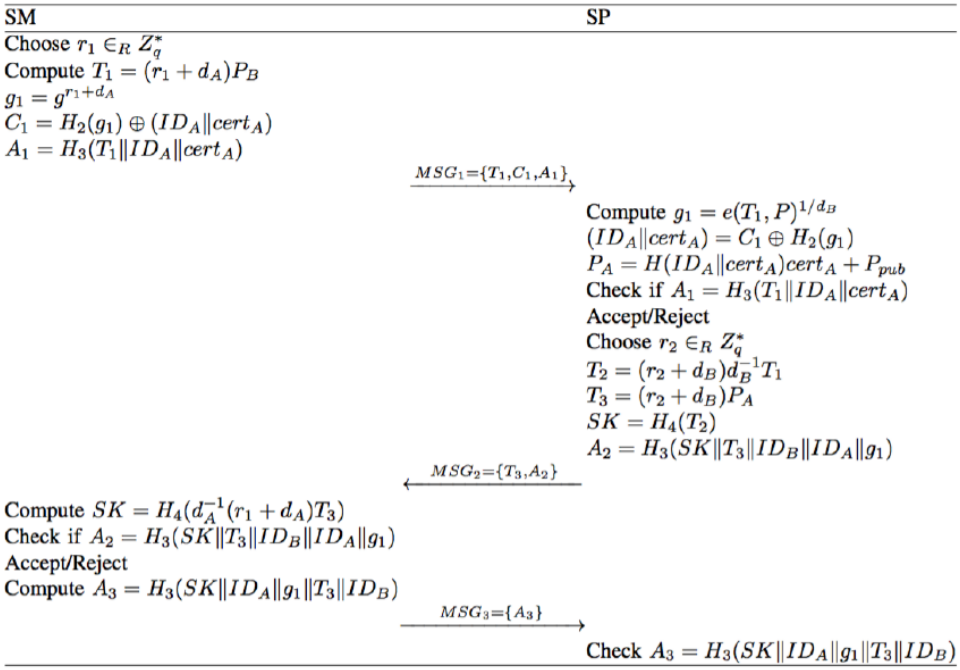

Figure 3. Review of key agreement phase of Chen et al.'s scheme [11].

3.3. Review of Abbasinezhad et al.

Description of the scheme [12]:

TTP setup phase: The TTP first chooses an elliptic curve (EC) $E_p(a, b)$ in Z_p with generator point P of order q. Also six hash functions are identified: $H_0, H_1, H_2, H_3, H_5 : \{0,1\}^* \rightarrow Z_q^*$, $H_4 : \{0,1\}^* \rightarrow \{0,1\}^{q+f}$, with f the size of the identity.

Then the private key k is chosen and its corresponding public key is computed $P_{pub} = kP$. These system parameters are published by the TTP.

Registration phase: Based on the ECQV-implicit certificate mechanism, both the SM and the SP are able to construct their key material. As a result, the key pairs of SM and SP equal to (d_A, P_A) and (d_B, P_B) with corresponding certificate $cert_A$ and $cert_B$ respectively.

Key negotiation phase: The steps in the key negotiation phase are summarised in Figure 4.

Weaknesses in [12]: Now we discuss the weaknesses of Abbasinezhad et al.'s scheme, as follows.

(A) Security in CK model: The scheme is not resistant in the CK security model if the private keys d_A and d_B of the SM and SP respectively are leaked. This follows from the fact that $C_A = C_B = d_B A_A$ and $D_A = D_B = d_A B_B$. As a result, the adversary is able to compute $SK = H(ID_A \| ID_B \| C_A \| D_A)$ and the security of the session key is broken.

(B) Late detection of MITM: Lead to DoS: The same issue regarding the late detection of a MITM attack also holds in this scheme. After receiving MSG_1, the SP is unable to check if this request is coming from a registered SM. Only after receiving the last message MSG_3 and verifying the hash value V_3, the SP can decide if the key negotiation has been successfully executed. As a result, this weakness could lead to a DoS threat.

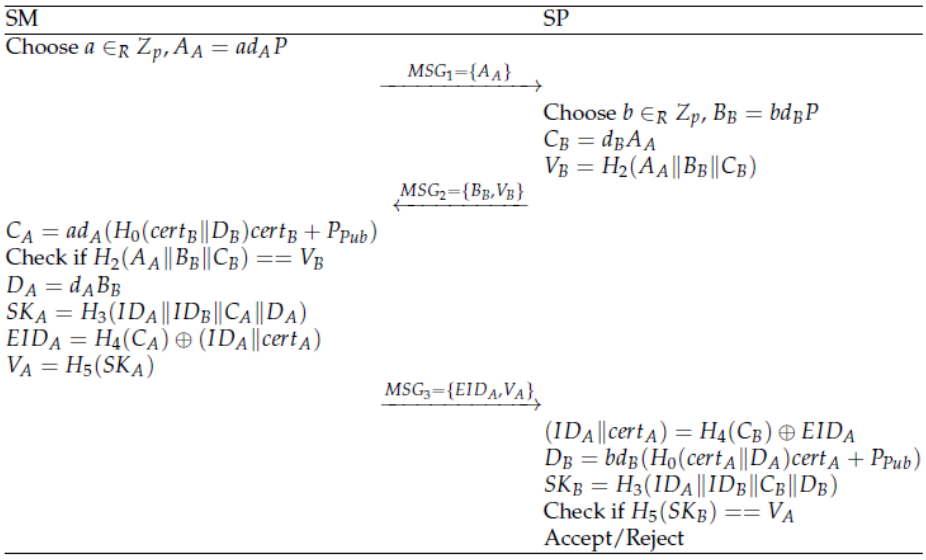

Figure 4. Steps and computations in the key agreement phase of Abbasinezhad et al.'s scheme [12].

3.4. Lessons Learned

We here summarise the two most important lessons which need to be taken into account when deriving a key agreement protocol, offering resistance in the CK model and being less vulnerable for DoS attacks.

- Resistance in CK model: Any possible definition of the SK should consist of a combination of both the local state variable and the private key of at least one entity. For instance, in [10,11], the SK can be written in a form only consisting of local state variables of the two entities and in [12], the SK is derivable to a form only consisting of the private keys of the two entities.
- Resistance to DoS attacks: In [10–12], the SP only retrieves the identity of the SMs after receiving a second message and is not able to detect malicious behaviour. Therefore, to avoid the SP keeping open a huge amount of (potential malicious) sessions, it is better to let the request initiated by the SP. The SM is less vulnerable since it can react when too many requests are sent as they are coming from only one entity, i.e., the SP.

4. Proposed Scheme

The proposed scheme consists of three main phases.

4.1. Setup Phase

In this phase, the TTP selects the EC $E_{p(a,b)}$ in Z_p with generator point P of order q. It determines six hash functions $H_0 : \{0,1\}^* \rightarrow Z_q^*$, $H_1 : Z_q \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow Z_q^*$, $H_3 : Z_q \rightarrow Z_q^*$, $H_4 : \{0,1\}^* \rightarrow Z_q^*$, and $H_5 : \{0,1\}^* \rightarrow Z_q^*$. Also a symmetric key encryption algorithm is chosen to encrypt a message M into the ciphertext C using the secret shared key k, $C = E_k(M)$, together with the corresponding decryption algorithm, $M = D_k(C)$. A random value k is set as the private key of the TTP. The corresponding public key P_{pub} is computed by $P_{pub} = kP$. Now, the public parameters $\{E_{p(a,b)}, P_{pub}, P, H_0, H_1, H_2, H_3, H_4, H_5, E_k(), D_k()\}$ are published.

4.2. Registration Phase

The registration phase for SMs and SPs are similar and follow the ECQV certificate scheme, refer to Section 2.3. As a result, each entity U is storing the public parameters $\{E_{p(a,b)}, P, H_0, H_1, H_2, H_3, E_k(), D_k(), P_{pub}\}$, its public key P_u, certificate $cert_U$ and identity ID_U, together with its private key d_U. Note that only the private key needs to be stored in the tamper resistant part of the memory.

Similar as in the other papers in literature on key agreement schemes, we assume that the SM is also storing the public key of the SP. If not, it needs to request before the key agreement phase the identity and certificate of the SP in order to compute the corresponding public key and verify the certificate, cf. Equation (1).

4.3. Key Agreement Phase

In the key agreement phase, the actual symmetric secret shared SK between SM and SP is established. We denote the SM by the entity with identity ID_A, key pair (d_A, P_A) and certificate $cert_A$. Similar, the SP is denoted by the entity with identity ID_B, key pair (d_B, P_B) and certificate $cert_B$. The detailed description is as follows.

SP broadcast: Every fixed period, the SP broadcasts the EC point $M_0 = \{R_2\}$. This point is computed by randomly choosing a variable r_2 and $R_2 = r_2P$.

SM response: For each SM that wants to do a key update or start a key negotiation process, the following steps are executed. First, the SM chooses a random value r_1 and computes the EC point $R_1 = (r_1 + d_A)P$. Next, the SM derives the symmetric shared key with the SP by $K = H_1((r_1 + d_A)P_B)$. This key is used to encrypt the identity information of the SM, $C = E_K(ID_A \| cert_A)$. Also the SK is computed by $SK = H_3(((r_1 + d_A)h_1 + d_A)(h_2R_2 + P_B))$, with $h_1 = H_2(ID_A \| ID_B \| R_1 \| R_2 \| P_A \| P_B)$ and $h_2 = H_2(ID_B \| ID_A \| R_2 \| R_1 \| P_B \| P_A)$. Finally, a hash value is computed, $S_1 = H_4(R_1 \| C \| P_A \| SK)$. The message $M_1 = \{R_1, C, S_1\}$ is sent to the SP.

SP response: Upon arrival of this message, the SP first computes the key, $K = H_1(d_BR_1)$ in order to decrypt C_1 and to derive $ID_A, cert_A$. Using these values, the public key P_A of the SM can be computed (Equation (1)), $P_A = H_0(cert_A \| ID_A)cert_A + P_{pub}$. At this point, the SK can be computed by the SP as $SK = H_3((r_2h_2 + d_B)(h_1R_1 + P_A))$, with again $h_1 = H_2(ID_A \| ID_B \| R_1 \| R_2 \| P_A \| P_B)$ and $h_2 = H_2(ID_B \| ID_A \| R_2 \| R_1 \| P_B \| P_A)$. Finally, using the received values R_1, C and the computed values P_A, SK, the hash function $H_4(R_1 \| C \| P_A \| SK)$ is computed and verified if it corresponds with the actual received S_1 value.

SP confirmation: If the verification of the SP is positive, a key confirmation is sent by the SP. Therefore, the hash $S_2 = H_5(ID_A \| ID_B \| R_1 \| R_2 \| P_A \| P_B \| SK)$ is computed and $M_2 = \{S_2\}$ is sent to the SM.

Finally, if also the SM verifies the correctness of the received hash, both SM and SP have successfully derived a common shared secret key, SK. The key agreement phase is summarised in Figure 5.

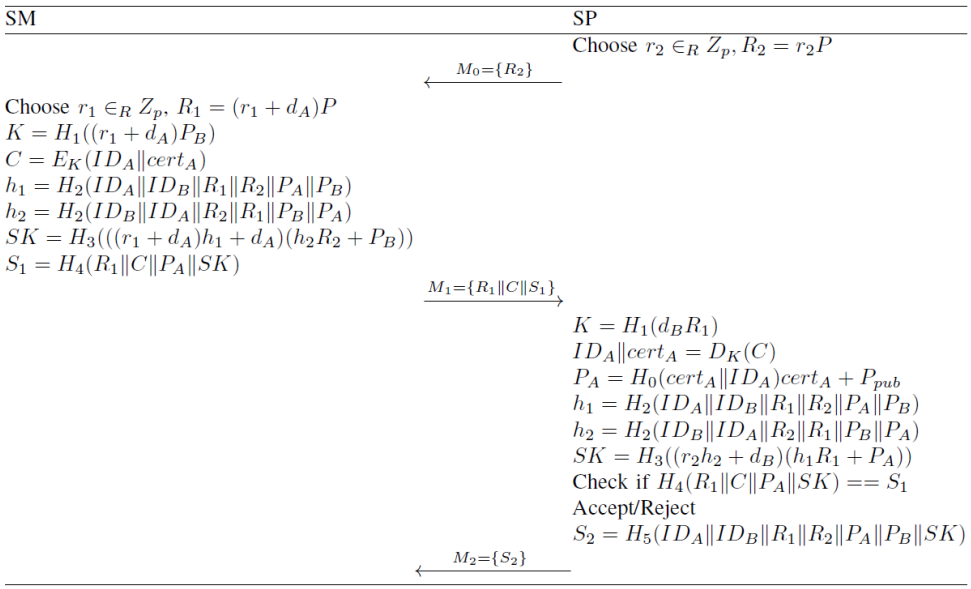

Figure 5. Steps and computations in the key agreement phase of proposed scheme.

Remark. Note that the SK is the result of a hash function, whose length is normally longer than the required key size of an encryption algorithm of the same security, due to the birthday problem. Therefore, in practice, only the first l bits of SK are considered.

5. Security Analysis

5.1. Formal Security Analysis

We now show that our key agreement scheme offers session key security under the CK adversary model [3,21] and in the random oracle model, following the method of [10,11,22]. The participants U in our scheme are the SM, SP, TTP or a random oracle O, i.e., $U = \{SM, SP, TTP, O\}$. Taking into account the CK adversary model, we assume that the attacker can run the following queries.

- Hash queries $H_i(m)$ with $i \in \{0, 1, 2, 3, 4, 5\}$. If m already exists in the list L_{H_i}, the value $H_i(m)$ will be returned. Otherwise, a random value will be generated, added to the list L_{H_i}, and returned.
- Send queries. These queries simulate active attacks, in which the adversary is able to modify the transmitted messages. As a result, a corresponding reply will be generated. Since there are three communication passes, four different Send queries need to be defined.
  - Send(0,SP). A random value r_2 is chosen to compute $R_2 = r_2P$. The output of the query is $M_0 = \{R_2\}$.
  - Send(M_0,SM). A random value r_1 is chosen to compute $R_1 = (r_1 + d_A)P$. Next, $K = H_1((r_1 + d_A)P_B)$ is determined, together with $C = E_K(ID_A \| cert_A)$. Then, $h_1 = H_2(ID_A \| ID_B \| R_1 \| R_2 \| P_A \| P_B)$ and $h_2 = H_2(ID_B \| ID_A \| R_2 \| R_1 \| P_B \| P_A)$ are computed to derive $SK = H_3(((r_1 + d_A)h_1 + d_A)(h_2R_2 + P_B))$. Finally, $S_1 = H_4(R_1 \| C \| P_A \| SK)$ is computed. The message $M_1 = \{R_1, C, S_1\}$ is returned.
  - Send(M_1,SP). First, $K = H_1(d_BR_1)$ is determined, leading to $ID_A \| cert_A = D_K(C)$. Then, $P_A = H_0(cert_A \| ID_A)cert_A + P_{TTP}$ is derived. Next, $h_1 = H_2(ID_A \| ID_B \| R_1 \| R_2 \| P_A \| P_B)$ and $h_2 = H_2(ID_B \| ID_A \| R_2 \| R_1 \| P_B \| P_A)$ are computed, to find $SK = H_3((r_2h_2 + d_B)(h_1R_1 + P_A))$ and check $H_4(R_1 \| C \| P_A \| SK)$ against S_1. If the verification is unsuccessful, the session can stop, otherwise $S_2 = H_5(ID_A \| ID_B \| R_1 \| R_2 \| P_A \| P_B \| SK)$ is computed and $M_2 = \{S_2\}$ is the output of the query.
  - Send(M_2,SP). If $S_2 = H_5(ID_A \| ID_B \| R_1 \| R_2 \| P_A \| P_B \| SK)$ is not valid, then the session is terminated. Otherwise, both SP and SM have successfully negotiated a common secret key SK.
- Execute queries. These queries simulate the passive attacks, in which the adversary can only eavesdrop onto the channel and is able to collect the transmitted messages. We can distinguish three different execute queries resulting from the first three Send queries, as defined above.
- Session specific state reveal queries (SSReveal). According to the CK adversary model, the attacker is able to retrieve session specific state information, derived by the SM and the SP, respectively. Note that no long-term private keys are revealed in this query.
  - SSReveal(SM). The output of this query results in $r_1 + d_A, h_1, h_2, R_1, C, S_1$.
  - SSReveal(SP). The output of this query results in r_2, R_2, h_1, h_2, S_2.
- Corrupt queries. These queries give the private key of the entity as result. Note that only Corrupt(SM) and Corrupt(SP) exist and no corrupt queries with regards to the TTP, which is considered a completely trusted entity. They are included to prove the perfect forward security of the scheme.
- Session key reveal query (SKReveal). In this query, the established symmetric SK between SP and SM is returned in case it has been successfully generated.
- Test query. In this query, either the established SK or a random value is returned, dependent on the output $c = 1$ or $c = 0$, respectively of a flipped coin c. Note that the test query cannot be issued

when the SKReveal query, the SSReveal(SM) and Corrupt(SM), or SSReveal(SP) and Corrupt(SP), have been executed.

In order to prove the semantic security of the scheme, we consider the following two definitions.

- The SP and SM are partners if they are exchanging messages directly and are the only entities able to successfully derive an authenticated common shared SK.
- The established shared secret key is said to be fresh if the SK has been established without SKReveal query by the adversary or Corrupt query of SM and SP.

The final goal of the adversary $\mathcal{A}$ is to distinguish the difference between a real secret session key or a random value, i.e., to successfully predict the output of the test query. If $Pr(succ)$ denotes the probability that the adversary succeeds in its mission, the advantage of the adversary in breaking the semantic security of the proposed scheme equals to $Adv(\mathcal{A}) = |2Pr[succ] - 1|$. Consequently, our scheme offers semantic security under the CK adversary and random oracle model if the advantage for $\mathcal{A}$ winning the game satisfies $Adv(\mathcal{A}) \leq \epsilon$, for any sufficiently small ϵ >0. The difference lemma [23] is used to prove the statement.

Lemma 1. *(Difference Lemma) Let E_1, E_2 be the events of winning game 1 and game 2. Denote an error event by E, such that $E_1|\neg E$ occurs if and only if $E_2|\neg E$. Then, $|Pr[E_1] - Pr[E_2]| \leq Pr[E]$.*

Theorem 1. *Let $\mathcal{A}$ be a polynomial-time adversary against the semantic security, which makes a maximum of q_s Send queries, q_e Execute queries and q_h Hash queries. The advantage of $\mathcal{A}$ is bounded by $Adv(\mathcal{A}) \leq \frac{O(q_s+q_e)^2}{2q} + \frac{O(q_h)^2}{2q} + \frac{O(q_s+q_h)}{q} + O(q_h T)$, with T the time to solve the ECDH problem.*

Proof. We proof the theorem by means of game hopping [21,23]. An attacker's success probability only increases by a negligible amount when moving between the games, as a consequence of Lemma 1. There are five games {GM0,GM1,GM2,GM3,GM4} to be defined. Denote by $succ_i$ the event that $\mathcal{A}$ wins the game GM_i, with $0 \leq i \leq 4$.

- Game GM0. This is the real game, as defined in the semantic security framework. From the definition, we have that

$$Adv(\mathcal{A}) = |2Pr[succ_0] - 1|. \tag{2}$$

- Game GM1. In this game, the oracles for the different queries are simulated and the resulting outputs of the queries are stored in the lists. In the random oracle model, it holds that

$$Pr[succ_1] = Pr[succ_0]. \tag{3}$$

- Game GM2. In this game, also all oracles are simulated, but collisions are avoided in the output of the hash functions and the selection of the random values r_1, r_2 among the different sessions. Due to the birthday paradox, the probability that these two events appear is bounded by $\frac{O(q_s+q_e)^2}{2q}$ and $\frac{O(q_h)^2}{2q}$ respectively. Consequently, it holds

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{O(q_s + q_e)^2}{2q} + \frac{O(q_h)^2}{2q}. \tag{4}$$

- Game GM3. In this game, the adversary $\mathcal{A}$ is able to find the hash values S_1, S_2 without input of the random oracle Send queries. In this case, the scheme is simply terminated. Consequently, GM3 and GM2 are indistinguishable, if and only if the SM rejects S_2 or the SP rejects S_1. Rejection only appears when the requested message belongs to the list of stored transcripts

and when the hash value is also present in the list of stored record, kept by the adversary $\mathcal{A}$. As a consequence, due to the difference lemma, we have that

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{O(q_s)^2}{2^l}. \tag{5}$$

- Game GM4. In this game, we consider the CK adversary model and assume that either the session state variables or the long-term secret variables are revealed at each of the involved participants. The goal of the adversary is to find the SK by performing Execute and Hash queries, with four possible combinations of SSReveal and Corrupt queries. In each of the four scenarios, we show that for successfully deriving the SK, one should be able to both solve the ECDHP and to perform a successful hash query of H_1.

  - SSReveal(SM) and SSReveal(SP). As a result of these queries, the adversary $\mathcal{A}$ possesses $r_1 + d_A, h_1, h_2, R_1, C, S_1$ and r_2, R_2, h_1, h_2, S_2, respectively. In order to compute the session key,

$$\begin{aligned} SK &= H_3(((r_1 + d_A)h_1 + d_A)(h_2R_2 + P_B)) \\ &= H_3((r_2h_2 + d_B)(h_1R_1 + P_A)) \end{aligned} \tag{6}$$

    either d_A or d_B is required. An impersonation attack is also not possible due to the usage of the ECQV implicit certificates scheme [20] and the presence of both P_A, P_B in the SK computation.
  - SSReveal(SM) and Corrupt(SP). Here, the adversary $\mathcal{A}$ receives the information $r_1 + d_A, h_1, h_2, R_1, C, S_1$ and d_B, respectively. For the derivation of the SK, cf. Equation (6), also r_2 or d_A is required.
  - Corrupt(SM) and SSReveal(SP). After performing these queries, the adversary $\mathcal{A}$ learns the information d_A and r_2, R_2, h_1, h_2, S_2, respectively. However, for the derivation of the SK, cf. Equation (6), also r_1 or d_B is required.
  - Corrupt(SM) and Corrupt(SP). In this case, as a result of these queries, the adversary $\mathcal{A}$ possesses the information d_A and d_B, respectively. However, for the derivation of the SK, cf. Equation (6), also r_1 or r_2 is required.

To conclude, the difference between GM3 and GM4 is negligible as long as the probability to solve the ECDHP and to perform a successful hash query of H_3 is small. Consequently,

$$|Pr[Succ_3] - Pr[Succ_2]| \leq O(q_h T) \tag{7}$$

with T the time to solve the ECDH problem and $Succ_3$ the event that $\mathcal{A}$ wins the game GM3.

Consequently, applying Lemma 1 on the games GM0, GM1, GM2, GM3 and GM4, taking into account Equations (2)–(5) and (7), results in the final proof of the theorem. □

5.2. Security Simulation Using AVISPA Tool [16]

In this subsection, we perform the security verification simulations for the proposed scheme using the AVISPA tool. It is a software tool and quite popular for performing automated verification of Internet security protocols and applications. The tool is widely used in industry and academia. For verification, the tool integrates backend servers, such as On-the-Fly Model Checker (OFMC) and Constraint-Logic-based Attack Searcher (Cl-AtSe). The AVISPA uses a high-level security protocol specification language (HLPSL). The HLPSL specifies the roles of each actor (e.g., SM and SP). These roles are as follows: (i) *Basic role* reports what initial information can be used by the SP and SM and how the conversions are being happening in the protocol; (ii) *Composition role* reports a session where the SP and SM are communicating together; (iii) Environment role provides details

the global parameters, sessions and an attacker knowledge in the key agreement phase in the proposed protocol. In security verification, the tool utilizes the Dolve-Yao model [24], which is represented as the *channel(dy)*.

We developed HLPSL script for the proposed scheme (i.e., key agreement phase). The SP and SM basic roles are shown in Figure 6. The results of the security verification presented in Figure 7 show that the proposed scheme is secure and *SAFE* from attacks using the (OFMC) backend. Precisely, Table 1 summaries the results from the AVISPA tool that the proposed key agreement achieved confidentiality and authentication. In addition, it is safe from MITM and replay attacks with a bounded number of sessions.

```
role ServiceProvider(SM, SP : agent,
        K, SK : symmetric_key,
        SPp : public_key,
        H       : hash_func,
        RCV, SND : channel (dy) )
played_by SM def=
local State                 : nat,
       idA, idB,            : text,
       M0,M1,M2             : message,
        H                   : hash_func
Const  SM_r1, SP_r2: protocol_id,
sub1, sub2: protocol_id,
init  State := 0
transition
1. State = 0 /\ RCV (start) =|>
   State′ := 1 /\ SM_r2′ := new ()
   State′ := 2 /\ SND (M0)
    State′ := 3 /\ RCV (R1',C',S1')
              …..
              …..
     State′ := 4 /\ SND (S2')

              …..
end role
```

HLSPL script for SP basic role

```
role SmartMeter (SM, SP : agent,
        K, SK : symmetric_key,
        SMp : public_key,
        H       : hash_func,
        RCV, SND : channel (dy) )
played_by SM def=
local State                 : nat,
       idA, idB,            : text,
       M0,M1,M2             : message,
        H                   : hash_func
Const  SM_r1, SP_r2: protocol_id,
sub1, sub2: protocol_id,
init  State := 0
transition
1. State = 0 /\ RCV (M0) =|>
   State′ := 1 /\ SM_r1′ := new ()
              /\ K′ := H((r1+dA)PB)
              …..
              …..
   State′ := 2 /\ SND (R1',C',S1')
              …..
              …..

end role
```

HLSPL script for SM basic role

Figure 6. HLPSL script for the SP and SM for their basic role.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/project/KeyAgreement.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 10.09s
  visitedNodes: 18 nodes
  depth: 10000 plies
```

Figure 7. Security simulation verification results of the proposed scheme.

Table 1. Summary of the results in AVISPA tool.

Security Services	Safe From
Confidentiality	MITM attack
Authentication	Replay attack

5.3. Informal Security Analysis

We now informally discuss the strength of the proposed protocol with respect to the required security features for an identity-based mutual authentication scheme, to be applied in a SG context [10,11].

- Resistance against replay attacks. There are two options, either M_1 is replayed in the same period of R_2 usage or it is replayed when a new R_2 is determined by the SP. In the first case, the same key as before is derived. However by capturing M_2, which is a hash value containing the SK, no additional information can be derived by the attacker. If the server keeps track of the parameters R_1 sent during the period in which R_2 remains constant, further action of the SP can be avoided.
  In the second case, a new session key is generated by the SP. However, when checking the hash value S_1 a contraction is found by the SP as the SK is different. The session is then stopped immediately.
- Resistance against impersonation attacks. There are again two options, impersonation of messages M_0 and M_1. First, it is impossible to impersonate the message M_0 as it is used to construct the SK by both the SM and the SP. Even if the SM is using R_2, sent from a malicious entity, the corresponding SK computed by the SM will not correspond with the SK computed by the SP and at the point where S_1 is validated. At that moment, the session will be terminated.
  Also impersonation of the message M_1 sent by the SM is impossible. This follows from the fact that M_1 consists of the parameter R_1. Only the SP is able to derive from R_1 the common shared key K with the SM in order to decrypt the ciphertext C for finding the identity and certificate of the SM. From these two parameters and the strength of the ECQV certificate mechanism, the SP can construct the corresponding public key P_A of the SM. The construction of the SK by the SP exploits the usage of this public key P_A and its own private key d_B, which is also derivable by the SM who is in possession of the correct corresponding private key d_A and the public key P_B of the SP. Consequently, it is impossible for an attacker to impersonate M_1 without knowledge of a valid private-public key pair of a SM or to impersonate M_2 without knowledge of the private key d_B of the SP.
- Resistance against MITM attacks. For the same reasons as explained in the replay and impersonation attacks, it is impossible to execute a MITM attack. Note that this resistance also strongly relies from the authentication feature established through the ECQV certificate mechanism.
- Anonymity. From the messages M_0, M_1, M_2 sent in the protocol, no information on the identity of the SM can be derived. The only identity related information is hidden in the message C, which is encrypted using a key only derivable by the SP.
- Resistance against DoS attacks. First of all, we here consider DoS attacks from the side of the server as resistance from the SM is easier to obtain by just blocking the amount of sent requests. Compared to the previous schemes in literature [10–12,14], our scheme is initiated by the SP with one single and common message to all interested SMs. Consequently, no separated buffers from received messages of different unknown SMs need to be saved by the SP. Upon arrival of a message M_1 from a particular SM, the SP can immediately check the validity and integrity of it in one single phase. If the check is not successful, it can drop the request and go to the next received message.

5.4. Comparison of Security Features

Table 2 compares the security features of our scheme with other schemes presented for the purpose of identity-based mutual authentication in the context of SGs. It must be in addition mentioned that the security strength of [14] with respect to the anonymity feature and the replay, impersonation and MITM attack is strongly dependent on the number of pre-stored security material, which is established with the need for a secure communication channel. Consequently, the process to update the security material is very impractical and by generating a DoS attack both from the side of the SM and the SP, this key material can be very easily exhausted.

Based on Table 2 and the fact that anonymity of the SM's credentials is an elementary security feature, we decided to perform our performance analysis on the schemes providing anonymity. These schemes correspond also with the most recent schemes. Consequently, we compare our scheme with the schemes [9–12,14].

Table 2. Comparison of security features of other identity-based mutual authentication schemes.

	2011 [5]	**2012 [6]**	**2016 [7]**	**2016 [8]**	**2016 [9]**	**2018 [10]**	**2018 [11]**	**2018 [12]**	**2018 [14]**	**Ours**
R1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
R2	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
R3	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
R4	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
R5	No	No	Yes	No	No	No	No	No	Yes	Yes
R6	No	No	No	No	No	No	Yes	Yes	No	Yes
R7	No	No	No	No	No	No	No	No	No	Yes

R1: Resistance replay attack; R2: Resistance impersonation attack; R3:Resistance MITM attack; R4: Anonymity; R5: SK security in CK model; R6: No secure channel in registration; R7: Resistance DoS attack.

6. Performance Analysis

The performance analysis is split into the computation and communication costs.

6.1. Computation Costs

The computation costs are measured by counting the number of most compute intensive operations and taking their corresponding computational time into account. We denote the timing for the bilinear pairing as T_b, the point multiplication T_{mp}, point addition T_{ap}, modular exponentiation T_e, a symmetric encryption/decryption T_s, and hash operation T_h.

The timings of these operations have been computed in [25] on a personal computer with a 2.5 GHz CPU, an 8 GB RAM and Windows 7 as OS for an 80-bit security level. This corresponds with a hash function resulting in a 160-bit output and an EC of order 160, i.e., $q = 160$. For the timing of the same operations on a more constrained device, mimicking a SM, a single core 798 MHz CPU and 256 MB of RAM is chosen. We refer to [14] for the corresponding timings. In addition, similar as in [14], the time to execute a 128-bit arbiter PUF call on an embedded device MSP430 micro controller with 798 MHz CPU is derived from [26]. For the time to execute a fuzzy extractor generation operation, $T_{FE.Gen}$, and a fuzzy extractor reconstruction operation, $T_{RE.Rec}$, the code offset mechanism using the Bose-Chaudhuri-Hocquenghem (BCH) code is considered, as in [27].

Table 3 shows the comparison of the computational overheads between our scheme and [9–11,14]. As it can be seen, our scheme is offering a better overall computational complexity compared to most of the other schemes. The complexity is very close to the scheme of [12]. However, the scheme of [11] is still slightly more efficient at the side of the SM. This cost in performance needs to be paid in order to offer resistance in the CK security model.

In addition, there is a huge difference in complexity with [14] at the SM, as the timing in [14] strictly depends on the efficiency of the PUF and the fuzzy extractor and no EC operations are computed. However, it should be noted that the fuzzy extractor of the PUF has several severe limitations [15].

Table 3. Comparison of computational complexity among other schemes.

Scheme	Cost at SM	ms	Cost at SP	µs
[9]	$4T_{mp} + 4T_{ap} + T_e + 5T_h$	≈31.59	$3T_{mp} + 2T_{ap} + 2T_b + T_e + 5T_h$	≈38.27
[10]	$3T_{mp} + T_{ap} + T_e + 6T_h$	≈25.72	$2T_{mp} + T_{ap} + 2T_b + T_e + 6T_h$	≈37.28
[11]	$2T_{mp} + T_e + 5T_h$	≈19.79	$3T_{mp} + T_b + T_e + 5T_h$	≈21.26
[12]	$4T_{mp} + T_{ap} + 5T_h$	≈23.80	$4T_{mp} + T_{ap} + 5T_h$	≈3.98
[14]	$5T_h + T_{PUF} + T_{FE.Rec}$	≈ 3.53	$6T_h + T_{RE.Gen}$	≈1170
Ours	$4T_{mp} + T_{ap} + T_s + 5T_h$	≈23.81	$4T_{mp} + 2T_{ap} + T_s + 5T_h$	≈3.99

6.2. Communication Costs

For the communication costs, we determine the number of bits transmitted by both the SM and the SP. In all the considered schemes, the number of communication phases equals to three, except in [14], where there are four phases required. Note that we consider the 80-bit security level. Therefore, the length of the hash function and the nonces/random numbers equals to 160 bits. For the schemes of [9–11], the sizes of the groups G_1 and G_2, related to the pairing operation, are 320 and 512 bits, respectively. When sending an EC point, it is sufficient to send the x-coordinate together with a sign bit. In addition, we assume that the length of identity related information equals to 32 bits. The length of the PUF responses equals to 128 bits. As can be concluded from Table 4, our scheme, together with [12], requires the smallest number of transmitted bits to be sent over the channel, compared to the other schemes.

Consequently, in our scheme messages M_0, M_2 from SP contain 161 and 160 bits respectively, resulting in a total message size sent by the SP equal to 321. The message M_1 consists of 1 hash (160 bit), 1 EC point (161 bit) and 1 ciphertext message (192 bits) containing the identity and an EC point. Therefore, the SM sends a message of size 511 to the SP.

Table 4. Comparison of communication complexity.

Scheme	No of Bits Sent by SM	No of Bits Sent by SP	Total No of Bits	No of Comm. Rounds
[9]	928	480	1408	3
[10]	1248	672	1920	3
[11]	1152	480	1632	3
[12]	512	320	832	3
[14]	576	768	1344	4
Ours	511	321	832	3

6.3. Other Simulations

Other simulations considering a real time smart metering communications are performed. The total traffic volume gain at the SP can be a concern due to the high number of packets received from the SMs. For instance, in order to manage the load balancing in the SG, the SP receives consumption usages data packets (i.e., P) from a number of SMs (i.e., N), periodically (every 15/30 min). Therefore, the total traffic volume gain at the SP will be very high, which is $N \times P$ for each session.

The results of the total traffic volume gain (in bits) at the SP presented in Figure 8 by varying the number of SMs. In our analysis, we consider one session per SM that sends consumption usages data to the SP. As shown in Figure 8, in the proposed scheme, the total traffic gain volume is gradually increasing as the number of SMs are increasing, which is quite obvious. Nevertheless, the increase in traffic volume is significantly higher in [9–11,14] as compared to our proposed scheme and [12]. Hence, the proposed scheme is more secure and efficient than the state of art schemes.

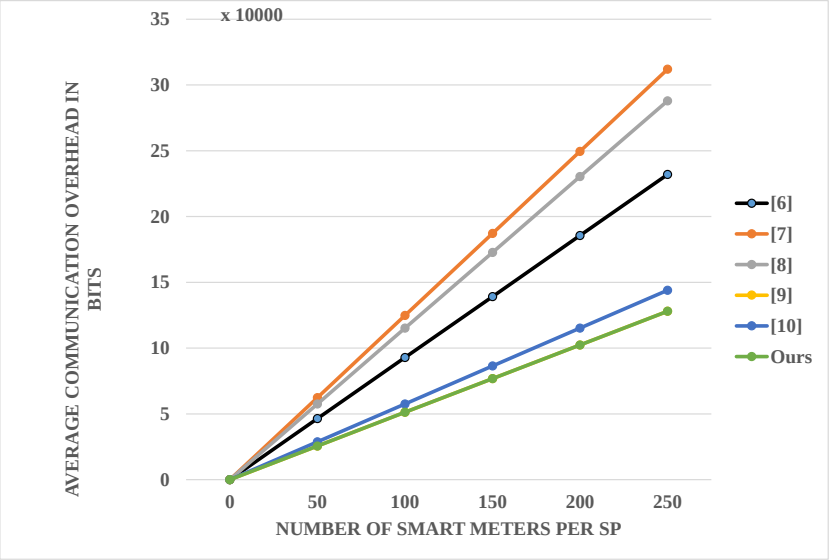

Figure 8. Total traffic volume at the SP from smart meters.

7. Conclusions

In this paper, we have shown weaknesses on three recently proposed authentication schemes in the context of SG, claiming to offer session key security in the CK security model. This security model has increasingly become the standard in common authentication protocols. From the lessons learned, we developed a new scheme which is able to provide effective session key security in the CK model and offered the same set of required security features in a smart metering setting and also in the SG. This set included identity-based mutual authentication, credential privacy, and session key security, as well as resistance against the well-known attacks of replay, MITM, and impersonation. Moreover, we also developed the scheme to offer maximum protection against DoS attacks by providing the possibility to the SP to derive from the first received message the validity of the request, thus avoiding the need for storing buffers coming from potential attackers, as required in the other schemes in the literature. Also from a computation and communication point of view, our proposed scheme behaves very well, if not excellently, compared to related work. Finally, we want to note that the application of our scheme goes beyond the domain of the SG, which will be purpose of future work.

Author Contributions: A.B. and P.K. have provided equal contribution in the paper. A.M. has provided the general comments.

Funding: The work of P. Kumar and A. Martin is supported by the UK EPSRC (Security and Privacy in Smart Grid Systems: Countermeasure and Formal Verifications) under Grant EP/NO20170/1. This is a joint research project with the National Research Foundation, Singapore (No. NRF2015NCR-NCR003-003). The authors would also like to acknowledge the contribution of the COST Action CA15127 (RECODIS).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ferrag, M.A.; Maglaras, L.A.; Janicke, H.; Jiang, J. A survey on privacy-preserving schemes for smart grid communications. *arXiv* **2016**, arxiv:1611.07722.
2. Liang, G.; Weller, S.R.; Zhao, J.; Luo, F.; Dong, Z.Y. The 2015 ukraine blackout: Implications for false data injection attacks. *IEEE Trans. Power Syst.* **2017**, *32*, 3317–3318. [CrossRef]
3. Canetti, R.; Krawczyk, H. Analysis of key-exchange protocols and their use for building secure channels. In *Advances in Cryptology EUROCRYPT 2001*; Springer: Berlin, Germany, 2001; pp. 453–474.
4. Boyd, C.; Mao, W.; Paterson, K. Key agreement using statistically keyed authenticators. In Proceedings of the Second International Conference on Applied Cryptography and Network Security, St. Petersburg, Russia, 21–23 May 2011; Springer: Berlin, Germany, 2021; Volume 3089, pp. 248–474.

5. Wu, D.; Zhou, C. Fault-tolerant and scalable key management for smart grid. *IEEE Trans. Smart Grid* **2011**, *2*, 375–381. [CrossRef]
6. Xia, J.; Wang, Y. Secure key distribution for the smart grid. *IEEE Trans. Smart Grid* **2012**, *3*, 1437–1443. [CrossRef]
7. Mahmood, K.; Chaudry, S.A.N.; Naqvi, H.; Shon, H.-T.; Ahmad, H.F. A lightweight message authentication scheme for Smart Grid Communications in power sector. *Comput. Electr. Eng.* **2016**, *52*, 114–124. [CrossRef]
8. Mohammadali, A.; Haghighi, M.S.; Tadayon, M.H.; Nodooshan, A.M. A novel identity-based key establishment method for advanced metering infrastructure in smart grid. *IEEE Trans. Smart Grid* **2016**, *9*, 2834–2842. [CrossRef]
9. Tsai, J.-L.; Lo, N.-W. Secure anonymous key distribution scheme for smart grid. *IEEE Trans. Smart Grid* **2016**, *7*, 906–914. [CrossRef]
10. Odelu, V.; Kumar, A.; Wazid, M.; Conti, M. Provably Secure Authenticated Key Agreement Scheme for Smart Grid. *IEEE Trans. Smart Grid*. **2018**, *9*, 1900–1910. [CrossRef]
11. Chen, Y.; Martinez, J.G.; Catellejo, P.; Lopez, L. An Anonymous Authentication and Key Establish Scheme for Smart Grid: FAuth. *Energies* **2018**, *10*, 1345. [CrossRef]
12. Abbasinezhad-Mood, D.; Nikoohgadam, M. Anonymous ECC-Based Self-Certified Key Distribution Scheme for Smart Grid. *IEEE Trans. Ind. Electron.* **2018**, *65*, 8. [CrossRef]
13. Li, X.; Ma, J.; Moon, S. On the Security of the Canetti-Krawczyk Model. In Proceedings of the Computational Intelligence and Security, Xi'an, China, 15–19 December 2005; Springer: Berlin, Germany, 2005; Volume 3802, pp. 356–363.
14. Gope, P.; Sikdar, B. Privacy-Aware Authenticated Key Agreement Scheme for Secure Smart Grid Communication. *IEEE Trans. Smart Grid* **2018**. [CrossRef]
15. Yasunag, K.; Yuzawa, K. On the Limitations of Computational Fuzzy Extractors. *ePrint Arch.* **2014**, *3*, 7–9.
16. Luca, V. Automated security protocol analysis with the AVISPA tool. *Electron. Notes Theor. Comput. Sci.* **2006**, *155*, 61–86.
17. Braeken, A.; Kumar, P.; Martin, A. Efficient and Privacy-Preserving Data Aggregation and Dynamic Billing in Smart Grid Metering Networks. *Energies* **2018**, *11*, 2085. [CrossRef]
18. Kumar, P.; Gurtov, A.; Sain, M.; Martin, A.; Ha, P. Lightweight Authentication and Key Agreement for Smart Metering in Smart Energy Networks. *IEEE Trans. Smart Grid* **2018**. [CrossRef]
19. Qu, M.; Vanstone, S.A. Implicit Certificate Scheme. US Patent 6,792,530, 14 September 2004.
20. Brown, D.R.L.; Scott, R.-G.; Vanstone, A. Provably Secure Implicit Certificate Schemes. In Proceedings of the 5th International Conferenc on International Conference on Financial Cryptography, Grand Cayman, Cayman Islands, 19–22 February 2001; Volume 2339, pp. 156—165.
21. Bellare, M.; Rogaway, P. The security of triple encryption and a framework for code-based game-playing proofs. In Proceedings of the 25th Eurocrypt Conference on Advances in Cryptology Eurocrypt 2006, St. Petersburg, Russia, 28 May–1 June 2006; Springer: Berlin, Germany, 2006; Volume 4004, pp. 409–426.
22. Pointcheval, D.; Zimmer, S. Multi-factor authenticated key exchange. In Proceedings of the 6th International Conference Applied Cryptography and Network Security, New York, NY, USA, 3–6 June 2008; Springer: Berlin, Germany, 2008; pp. 77–95.
23. Shoup, V. Sequences of Games: A Tool for Taming Complexity in Security Proofs. 2004. Available online: http://eprint.iacr.org/2004/332/ (accessed on 28 August 2018)
24. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
25. He, D.; Zeadally, S.; Wang, H.; Liu, Q. Lightweight Data Aggregation Scheme against Internal Attackers in Smart Grid Using Elliptic Curve Cryptography. *Wirel. Commun. Mob. Comput.* **2017**, *2017*, 3194845. [CrossRef]
26. Herder, C.; Yu, M.D.; Koushanfar, F.; Devadas, S. Physical Unclonable Functions and Applications: A Tutorial. *Proc. IEEE* **2014**, *102*, 1126–1141. [CrossRef]
27. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noise data. *SIAM J. Compt.* **2008**, *38*, 97–139. [CrossRef]