Journal of Computer and System Sciences 83 (2017) 84–100

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Secure and efficient protection of consumer privacy in Advanced Metering Infrastructure supporting fine-grained data analysis

Vitaly Ford *, Ambareen Siraj, Mohammad Ashiqur Rahman

Tennessee Tech University, P.O. Box 5101, Cookeville, TN 38505, USA

ARTICLE INFO

Article history:
Received 11 October 2015
Received in revised form 26 April 2016
Accepted 9 June 2016
Available online 7 July 2016

Keywords:
Security
AMI
Privacy preservation
Trusted third party
Fine-grained data

ABSTRACT

The Advanced Metering Infrastructure (AMI) plays a critical role in the Smart Grid. In regarding the usage of smart meters in AMI, there is a primary concern about how utility companies manage energy consumption data, particularly with respect to consumer privacy. This research presents a novel protocol for secure and efficient communication of energy consumption data, protecting its confidentiality, integrity, and privacy while utilizing the existing Grid infrastructure. The protocol supports time-of-use billing and data mining for advanced fine-grained data analysis. We report on the empirical results of the theoretical, experimental, and comparative analyses of the proposed protocol.

1. Introduction

Constantly evolving revolutionary technologies in power grid systems provide a variety of opportunities for making energy distribution more efficient and reliable. Advanced power meters (smart meters) enable efficient close monitoring of energy usage and capturing of fine-grained Energy Consumption (*EC*) readings, which facilitate advanced intelligent analysis. Examples of such analysis include (but not limited to) identifying opportunities to reduce *EC* and cost [1,2], detecting energy fraud [3–5], and providing flexible billing and load monitoring. A major concern of customers with this reporting of fine-grained *EC* data is a potential revelation of sensitive and private information about their daily life [6–8]. To alleviate this concern, several zero-knowledge proof and homomorphic encryption-based privacy-preserving communication protocols [9–11] have been proposed. However, while preserving the privacy of consumers, these protocols sacrifice the reporting of granular data for advanced analysis. Therefore, the advantage of using smart meters is lost to a certain extent. Although protecting the privacy of consumer's *EC* data is very important, it is all the more essential to retain the capability of advanced analysis for efficient energy management.

Therefore, we introduce a three-tier model for secure smart meter communication that enables consumer privacy preservation as well as retention of fine-grained data analysis capability. The model comprises of Smart Meters (SMs), a Utility Company (UC), and a Trusted Third Party (TTP) managing a cloud-based storage system. TTP has direct access to fine-grained consumer data and has the capability to include additional advanced analysis features, such as fraud detection. TTP can manage meter data from several different electricity providers and thus release the Advanced Metering Infrastructure (AMI) from

* Corresponding author.
E-mail addresses: vford42@students.tntech.edu (V. Ford), asiraj@tntech.edu (A. Siraj), marahman@tntech.edu (M. Ashiqur Rahman).

http://dx.doi.org/10.1016/j.jcss.2016.06.005
0022-0000/

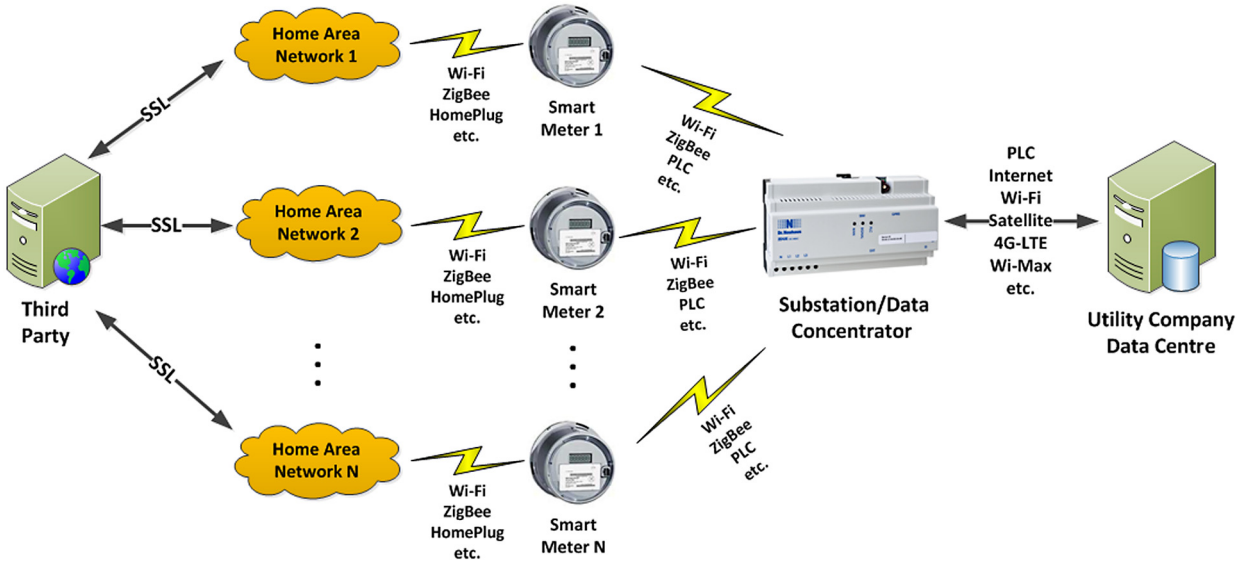

Fig. 1. AMI architecture.

unnecessary computations, such as aggregation, fraud detection, and *EC* analysis. In the proposed model, SMs encrypt all the *EC* data and send the encrypted traffic to TTP through separate collectors in the UC's smart metering network.

Existing solutions in AMI propose different dedicated protocols for authentication, load monitoring, aggregation, billing, and fraud detection [12–17]. Instead of using separate ones, the proposed architecture utilizes a comprehensive protocol with the capability to accomplish all of the above with minimal overhead to SM. The features of the proposed protocol are as follows:

- Fine-grained data are encrypted by SM and anonymized by UC before transmission to TTP.
- TTP cannot identify real consumers because of anonymization.
- Lightweight efficient encryption is used for securing all communication. This makes it computationally viable for implementation in resource-restrained AMI.
- For time-of-use billing, UC can provide TTP with the energy billing rates and acquire consumer bills calculated by TTP.
- UC cannot ask a consumer to pay a fee different from the one that was produced by TTP for billing. UC has neither the privilege of changing the energy measurements stored at TTP nor the ability to read consumer's fine-grained data.

The contribution of this work can be summarized in the following:

- Consumer privacy preservation as well as data confidentiality and integrity protection, utilizing the existing Grid infrastructure.
- Access to the fine-grained data, supporting advanced intelligent analysis like fraud detection.
- Support of time-of-use billing and a *retail choice* as described in [18]. Consumers can select an energy provider based on their electricity consumption behaviors and needs.
- Feasible computing by SMs, which are under sole control of UC.
- Utilization of a cloud-based Trusted Third Party outside of the Grid infrastructure.

This paper is structured as follows. Section 2 introduces the Advanced Metering Infrastructure. Section 3 outlines the proposed model. Section 4 discusses the trust model, research goals of this work, and assumptions. Section 5 describes the proposed protocol. Section 6 evaluates the proposed protocol's performance and its capabilities in consumer privacy preservation and secure communication. Section 7 describes the related work done in the area of secure and privacy-preserving AMI. Section 8 concludes with future work.

2. Advanced Metering Infrastructure

The Smart Grid is a modernized power grid incorporating various alternative energy sources, sensors, smart meters, and other advanced technologies for providing more efficient and reliable energy distribution and opportunities to effectively manage *EC* [19].

One of the main components of the Smart Grid is an Advanced Metering Infrastructure (AMI) consisting of smart meters, communication networks, home area networks, and other devices supporting energy usage monitoring (Fig. 1 [20]).

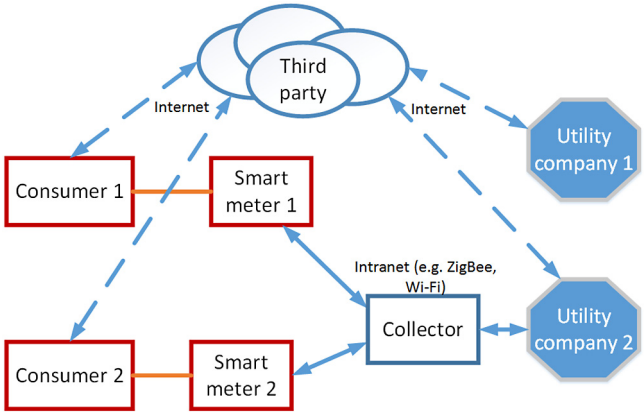

Fig. 2. Proposed architecture.

Smart meters allow a two-way communication with utility companies. It is crucial to keep meter data secure because malicious actors can carry out attacks such as compromising meters to send false energy measurements, disconnecting consumers from electricity supply, or profiling consumers' behavior for unauthorized purposes [7,21,22].

3. Approach

The following describes the proposed model for AMI. The three-tier system consists of Smart Meters (SMs), a Utility Company (UC), and a cloud-based Trusted Third Party (TTP) storage system. TTP is an independent private organization, whose service is purchased by UC. TTP can be represented as a set of decentralized clusters connected with each other over the Internet to support scalability. There is also a collector installed by UC, which facilitates collection of *EC* data from various SMs. Fig. 2 shows the high-level architecture (solid lines correspond to an internal UC network and dashed lines correspond to the Internet connectivity).

In the proposed model (Fig. 2), TTP is connected to UC via the wide area network. SMs are not directly connected with TTP and instead connected through UC. This is because SMs connect with their UC via an internal network (e.g., ZigBee/Wi-Fi/Ethernet) to decrease the possibility of attacks that are common in the Internet. In the event that the communication between SMs and TTP via UC is disrupted, UC can store the consumers' encrypted data locally and after restoring the connection, forward them to TTP.

In this model, UC deploys SMs and has limited control over them. The control is restricted for preserving consumer privacy and, therefore, UC is only allowed to provide administrative support for AMI, such as verifying SMs' availability or updating their firmware.

To protect against physical attacks, every SM can be equipped with a Trusted Platform Module (TPM). TPM is a secure storage that can generate pseudorandom numbers and perform cryptographic operations. It provides tamper-resistant hardware for keeping cryptographic keys safe from memory snooping attacks [23].

When UC deploys SM, it generates a random identification number (ID) for SM in the household and assigns a pre-shared secret key. SM and TTP initiate a certificateless public key exchange protocol [24] where UC serves as a Key Generation Center. Once public/private keys are distributed to both parties, TTP generates a session key for securely communicating *EC* data from SM to TTP and sends it via an encrypted connection (using public/private keys) to SM. SM stores the session key and uses it for sending energy readings to TTP via UC.

We use Advanced Encryption Standard (AES) 128-bit keys for securing communication between SM and UC as well as SM and TTP. In addition, we utilize a "CertificateLess Public Key Encryption without Bilinear Pairing" (CLPKE) [24] only on the initial stage, when a new SM is deployed or a key renewal is needed. CLPKE provides a means to securely establish public key communication without the need of certificate authorities. This type of cryptography has a better performance in comparison to the original "Certificateless Public Key Encryption using Bilinear Pairing" [25].

SM encrypts *EC* measurements and sends them to the collector. The main responsibility of the collector is to temporarily store the encrypted *EC* data and send them to UC periodically in a predefined time interval, for example, every 5 minutes. UC verifies and forwards the encrypted *EC* data to TTP. Without knowledge of a key, UC cannot decrypt the data and thus, privacy is preserved for their clients. TTP decrypts all received data and stores them in its local database.

At the time of billing, UC sends TTP a request for *EC* readings to be billed, including the anonymized meter ID and price ranges for different periods of time. TTP authenticates UC, queries the requested data from the database, and aggregates energy on a daily/monthly basis depending on the policy and bill calculation requirements. Afterwards, TTP sends UC the calculated bill.

When a consumer receives the bill from UC, he/she can check the correctness of the billing computations. Consumers can connect to the TTP web-service via the Internet, authenticate without revealing their real identity, and gain access to their fine-grained data as well as the billing rates. More details follow in Section 5.

4. Trust and threat models, goals, and assumptions

This section introduces the trust and threat models of the proposed architecture as well as security and usability goals.

4.1. Trust model

Following other researchers in this area, for instance [14], both TTP and UC are *honest-but-curious* in the proposed model, meaning that while they will follow the proposed protocol, if opportunity arise, they may attempt to gain as much knowledge as possible about consumers' information. UC also trusts TTP in accumulating fine-grained data and providing information to UC for billing the consumers.

4.2. Threat model

There are several security and privacy threats that exist in the AMI. Attackers can target a broad spectrum of the AMI components including SMs, data collectors, communication medium, UC, and TTP. We focus on protecting confidentiality, integrity, and privacy of consumer data against Man-In-The-Middle (MITM) attacks and rogue SMs that can attempt to impersonate legitimate SMs. At the same time, the proposed scheme also guarantees consumer privacy protection from *honest-but-curious* UC and TTP that may attempt to learn consumers' behaviors based on the energy consumption data.

4.3. Security and privacy goals

Several security and privacy goals are associated with the proposed architecture to protect consumer privacy and provide UC with precise billing information.

1. Provide confidentiality and integrity of communication among TTP, UC and SMs – transmitted data have to be protected against unauthorized reading and modification.
2. Offer identity privacy for consumers – it has to be impossible for TTP to trace or link real identity of consumers based on the stored data.
3. Offer data privacy for consumers – UC can only access the aggregated data.
4. Provide mutual authentication between TTP and consumers in such a way that only legitimate consumers can create personal accounts for storing and accessing their data at TTP and, at the same time, their personal identities are not exposed.

4.4. Usability goals

In addition to protecting consumer privacy and providing confidentiality and integrity of communication, the usability goals for UC and consumers are as follows.

1. Allow UC to bill consumers using time-of-use tariffs so that the energy bills are calculated depending on the time of energy consumption.
2. Support fine-grained data analysis for UC without violating consumer privacy.
3. Provide consumers with the opportunity to compare energy providers' prices to select the service provider that best satisfies their consumption profile. At the same time, ensure that consumers can always access their historical fine-grained data.
4. Process billing in such a way that UC cannot claim a different fee other than that reported by TTP.

4.5. Assumptions

Although we assume that UC and TTP are *honest-but-curious*, if attackers gain access to TTP, they cannot obtain any sensitive information about a particular consumer due to the fact that in this architecture TTP only stores anonymized data that are linked to real identities known only to UC. To use service offered by TTP, UC has to securely authenticate at TTP using two-factor authentication. The communication between UC and TTP is assumed to be secure by utilizing protocols like *TLS*.

We assume that the keys used by SM are secured by installation of a tamper-resistant device, such as a Trusted Platform Module. We also assume that UC and TTP do not collude as some other researchers in this area suppose [14].

The final assumption we make is that UC has access to SM before its deployment, meaning that UC can *privately* install a shared symmetric key for communication between UC and SM.

5. Protocol design

This section describes the proposed protocol as well as UC and consumer authentication at TTP. Table 1 shows the protocol notations used throughout the paper.

Table 1
Notations.

$S_{SM\text{-}UC}$	Encrypted with a pre-shared key $S_{SM\text{-}UC}$ between UC and SM
$S_{SM\text{-}TTP}$	Encrypted with a symmetric key $S_{SM\text{-}TTP}$ shared between TTP and SM
TLS	Encrypted over a *TLS* (Transport Layer Security) connection between UC (authenticated at TTP) and TTP
P_{SM}	Encrypted with a public key P_{SM} of SM
P_{TTP}	Encrypted with a public key P_{TTP} of TTP
HMAC	A keyed-hash message authentication code that uses $S_{SM\text{-}UC}$ as a shared secret key according to the NIST Keyed-Hash Message Authentication Code standard [26]
S_{SM}	Private (secret) key of SM
S_{TTP}	Private (secret) key of TTP
$R_{SM\text{-}TTP}$	A message containing a random number as well as a request to share $S_{SM\text{-}TTP}$
$A \mid\mid B$	Concatenation of B with A
ID_{SM}	Identification number of SM
$an\text{-}ID_{SM}$	Anonymized ID_{SM}
$\oplus$	XOR, *exclusive or* operation

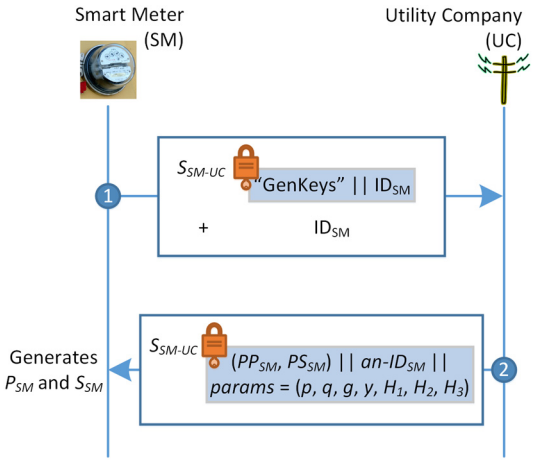

Fig. 3. Registration phase for SM.

5.1. Protocol

Initially, a Utility Company (UC) installs a pre-shared key ($S_{SM\text{-}UC}$) on a new Smart Meter (SM) before deploying it in a household. This key can be stored in SM's trusted platform module (TPM). $S_{SM\text{-}UC}$ is known only to UC and SM, therefore it is used for any communication between UC and SM to secure data transmission. Such communication can include (but not limited to) voltage readings and maintenance mode operations.

The initial key exchange algorithm between SM and TTP is built on an efficient Certificateless Public Key Cryptography (CPKC) without bilinear pairing [24]. This type of cryptography allows interested parties to authenticate among each other without the authenticity of their public keys. CPKC is proved in [24] to be secure against two types of adversaries described in [25] as:

"Type I Adversary $\mathcal{A}_1$, who does not have access to *masterKey* (derived below at the *Registration phase*). However, $\mathcal{A}_1$ may request public keys and replace public keys with values of its choice, extract partial private and private keys and make decryption queries, all for identities of its choice" [25].

"Type II Adversary $\mathcal{A}_2$, who does have access to *masterKey* but may not replace public keys of entities. Adversary $\mathcal{A}_2$ can compute partial private keys for itself, given master-key. It can also request public keys, make private key extraction queries and decryption queries, both for identities of its choice" [25].

There are three main phases in the proposed protocol: *registration phase*, *session key exchange phase*, and *data transmission phase*. The registration phase describes the steps that SM and TTP have to follow for receiving their public/private key pairs from UC. Those keys will allow SM and TTP to establish a secure and private connection for exchanging a *session key* used for further communication between SM and TTP at the *data transmission phase*.

5.1.1. Registration phase

At this phase, UC serves as a Key Generation Center. UC and TTP have to establish a *TLS* connection before sending messages to each other, meaning that UC has to go through the authentication process described in Section 5.2. SM and

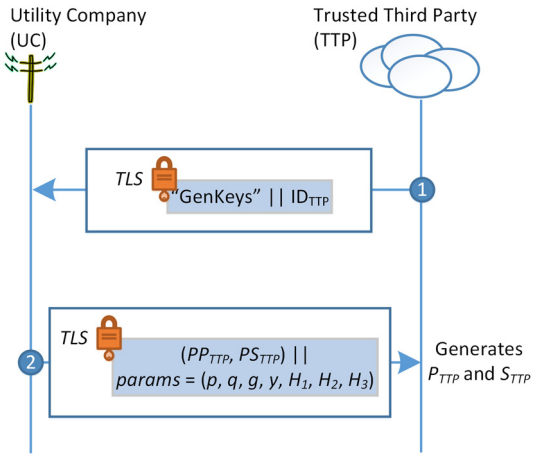

Fig. 4. Registration phase for TTP.

TTP communicate with UC in order to obtain public/private keys (Figs. 3 and 4). Any communication between SM and UC is encrypted with the pre-shared key $S_{SM\text{-}UC}$. When SM sends UC an encrypted (with $S_{SM\text{-}UC}$) message, it concatenates its ID_{SM} so that UC can identify the meter upon receiving the message and decrypt it accordingly. The message "GenKeys" consists of a request to generate keys and a timestamp against replay attacks. The following are the steps both SM and TTP have to fulfil to obtain their keys.

- As a one-time operation, UC has to establish the necessary cryptographic primitives and its private/public keys. UC generates two primes p and q such that $q|p-1$. UC picks a generator g of $\mathbb{Z}_p^*$ of order q. UC picks uniformly at random $x \in \mathbb{Z}_q^*$ as its private key and computes $y = g^x \, mod \, p$ as its public key. UC selects hash functions $H_1 : \{0,1\}^* \times \mathbb{Z}_p^* \rightarrow \mathbb{Z}_q^*$, $H_2 : \{0,1\}^{l_0} \times \{0,1\}^{l_1} \rightarrow \mathbb{Z}_q^*$ and $H_3 : \mathbb{Z}_p^* \times \mathbb{Z}_p^* \rightarrow \{0,1\}^l$, where $l = l_0 + l_1 \in \mathbb{N}$. UC keeps secret its master key, where $masterKey = (p, q, g, x, H_1, H_2, H_3)$.
- Given an identification ID_R (R stands for either SM or TTP) from the requester, UC generates requester's *partial* public and private keys. UC picks $s \in \mathbb{Z}_q^*$ at random and computes a partial public key PP_R as $PP_R = g^s mod \, p$ and a *partial* private key PS_R as $PS_R = s + xH_1(ID_R, PP_R) \, mod \, q$. UC returns (PP_R, PS_R) to the requester R (SM or TTP) as well as the public information parameters *params*, where $params = (p, q, g, y, H_1, H_2, H_3)$.
- Upon receiving the message from UC comprising of (PP_R, PS_R) and *params*, the requester verifies it by computing $g^{PS_R} = PP_R \cdot y^{H_1(ID_R, PP_R)} \, mod \, p$. If the equation holds, then the requester knows that the message comes from UC and can continue following the protocol's steps. The *correctness* property of the above equation is presented below:

$$g^{PS_R} = g^{s+xH_1(ID_R,PP_R)} = g^s g^{xH_1(ID_R,PP_R)} = PP_R \cdot y^{H_1(ID_R,PP_R)}.$$

  After substituting PS_R with $s + xH_1(ID_R, PP_R)$ and g^x with y, we can observe that the equation holds only if UC has correctly generated the partial public/private key pair.
- The requester generates its private key based on the *partial* private key PS_R received from UC. It picks $z_R \in \mathbb{Z}_q^*$ at random. Then, it sets its private key $S_R = (z_R, PS_R)$.
- The requester generates its public key based on publicly available *params* and the *partial* public key PP_R obtained from UC. The requester computes $\mu_R = g^{z_R} \, mod \, p$ and sets its public key $P_R = (PP_R, \mu_R)$.

It should be noted that when SM sends its ID_{SM} to UC, UC randomly selects an anonymized ID ($an\text{-}ID_{SM}$) and links it with ID_{SM}. Afterwards, UC generates partial public and private keys for SM based on that $an\text{-}ID_{SM}$ so that TTP cannot learn the real ID_{SM} while communicating with SM. UC sends SM the $an\text{-}ID_{SM}$ and SM temporarily stores it to verify the correctness of the *partial* public/private keys (PP_R, PS_R) received from UC. The $an\text{-}ID_{SM}$ preserves privacy of SM because TTP cannot trace the real ID_{SM} and can store data associated with only $an\text{-}ID_{SM}$ in its database. UC keeps a mapping table linking the real identity of SM (ID_{SM}) to the $an\text{-}ID_{SM}$ and, therefore, UC works as a mediator between SM and TTP.

5.1.2. Session key exchange phase

When SM and TTP complete the registration phase, they initiate a session key exchange in order to share a secret key used for encrypting/decrypting fine-grained meter readings (Fig. 5). SM generates a message M including $P_{SM} \, || \, R_{SM\text{-}TTP}$. $R_{SM\text{-}TTP}$ contains a random number so that when TTP sends it back along with the secret key (at the end of this phase), SM will be able to verify if $R_{SM\text{-}TTP}$ is the same as it is supposed to be (that way SM will attest the origin integrity of the secret key).

SM splits M into several parts $M_1, M_2, ..., M_n$ so that the bit-length of every sub-message M_i, $i = (1, 2, ..., n)$, is l_0 and the concatenation of all parts holds $M_1||M_2||...||M_n = M$. If the bit-length of the message M is not divisible by l_0, then the

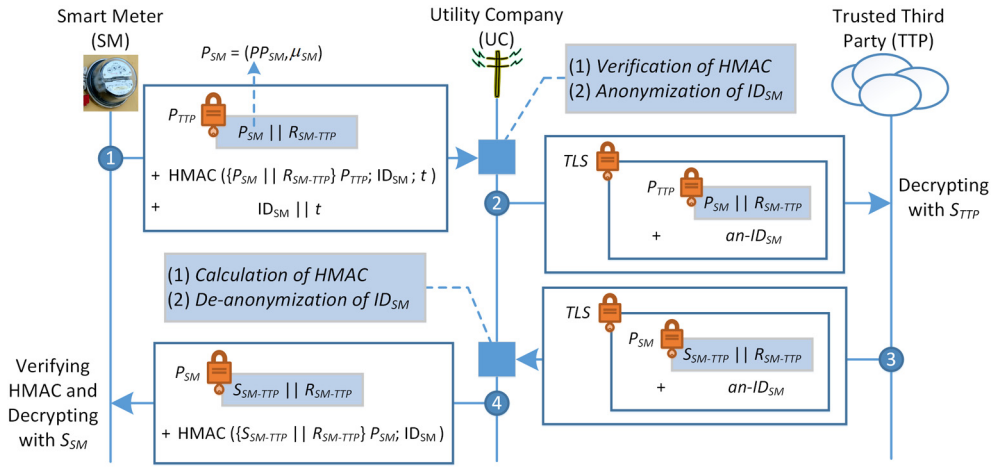

Fig. 5. Session key exchange phase.

padding algorithm PKCS#5 [27] can be used in order to split it into n even parts. Each sub-message M_i is encrypted as follows.

- SM uses TTP's public key $P_{TTP} = (PP_{TTP}, \mu_{TTP})$ and computes $\gamma_{TTP} = PP_{TTP} \cdot y^{H_1(ID_{TTP}, PP_{TTP})} \bmod p$.
- SM picks $\sigma \in \{0,1\}^{l_1}$ at random and computes $r = H_2(M_i, \sigma)$.
- SM calculates an encrypted $C = (c_1, c_2)$ such that $c_1 = g^r \bmod p$ and $c_2 = H_3(\mu_{TTP}^r \bmod p, \gamma_{TTP}^r \bmod p) \bigoplus (M_i||\sigma)$. The bit length of $(M_i||\sigma)$ is equal to $l = l_0 + l_1$.
- SM computes *HMAC* of $C = (c_1, c_2)$, its ID_{SM}, and a timestamp t (against replay attacks).
- SM concatenates the encrypted $C = (c_1, c_2)$ with the previously computed *HMAC*, its ID_{SM}, and t and sends them to UC.
- UC verifies *HMAC* received from SM, concatenates *an-ID*$_{SM}$ corresponding to ID_{SM}, and forwards the encrypted C with *an-ID*$_{SM}$ to TTP over the *TLS* channel.
  TTP decrypts the message $C = (c_1, c_2)$ as follows.
- TTP uses its private key $S_{TTP} = (z_{TTP}, PS_{TTP}) = (z, w)$ and $C = (c_1, c_2)$ to compute $(M_i||\sigma) = H_3(c_1^z \bmod p, c_1^w \bmod p) \bigoplus c_2$. The *correctness* property of the above-mentioned decryption procedure that accurately inverts the encryption is presented below:

$$
\begin{aligned}
&H_3(c_1^z, c_1^w) \bigoplus c_2 = H_3(g^{rz}, g^{rw}) \bigoplus H_3(\mu_{TTP}^r, \gamma_{TTP}^r) \bigoplus (M_i||\sigma) = \\
&H_3(g^{rz}, g^{rw}) \bigoplus H_3(g^{rz}, (g^s g^{xH_1(ID_{TTP}, PP_{TTP})})^r) \bigoplus (M_i||\sigma) = \\
&H_3(g^{rz}, g^{rw}) \bigoplus H_3(g^{rz}, (g^{s+xH_1(ID_{TTP}, PP_{TTP})})^r) \bigoplus (M_i||\sigma) = \\
&H_3(g^{rz}, g^{rw}) \bigoplus H_3(g^{rz}, (g^w)^r) \bigoplus (M_i||\sigma) = (M_i||\sigma).
\end{aligned}
$$

  As a result, TTP can correctly compute $(M_i||\sigma)$ by performing an *exclusive or* operation: $H_3(c_1^z \bmod p, c_1^w \bmod p) \bigoplus c_2$.
- TTP verifies the equation $g^{H_2(M_i,\sigma)} \bmod p = c_1$ and, if it holds, retrieves M_i from $(M_i||\sigma)$ as it knows that the bit-length of M_i is l_0 and the bit-length of $(M_i||\sigma)$ is $l = l_0 + l_1$.

After obtaining the information contained in M, TTP generates a session key $S_{SM\text{-}TTP}$, appends $R_{SM\text{-}TTP}$, and sends it to SM by encrypting it with P_{SM}, following similar steps as it is done at the beginning of the *session key exchange phase*. In addition, TTP concatenates *an-ID*$_{SM}$. When UC receives the message from TTP, it de-anonymizes *an-ID*$_{SM}$ and substitutes it with ID_{SM}. Subsequently, UC relays the message from TTP to SM along with ID_{SM} and *HMAC* of the encrypted (with P_{SM}) $S_{SM\text{-}TTP} \,||\, R_{SM\text{-}TTP}$. SM verifies data and origin integrity of the message from UC by recalculating *HMAC*. SM performs identical steps as TTP for decrypting and retrieving the session key $S_{SM\text{-}TTP}$ by using its private key S_{SM}. Having established a shared secret with TTP, SM deletes *an-ID*$_{SM}$ and its private $S_{SM} = (z_{SM}, PS_{SM})$ and public $P_{SM} = (P_{SM}, \mu_{SM})$ keys as there is no need for using them in future communication.

5.1.3. Data transmission phase

The session key established at the *session key exchange phase* is used for sending meter readings from SM to TTP (Fig. 6). Thus, only SM and TTP can decrypt the fine-grained measurements. SM sends UC the energy consumption (EC) along with a timestamp t by encrypting the data with $S_{SM\text{-}TTP}$. It also concatenates *HMAC* to the message by hashing $EC \,||\, t$ and its ID_{SM}. UC verifies *HMAC* and forwards the received data to TTP, replacing ID_{SM} with *an-ID*$_{SM}$. TTP decrypts $EC \,||\, t$ by using $S_{SM\text{-}TTP}$ and retrieves the data.

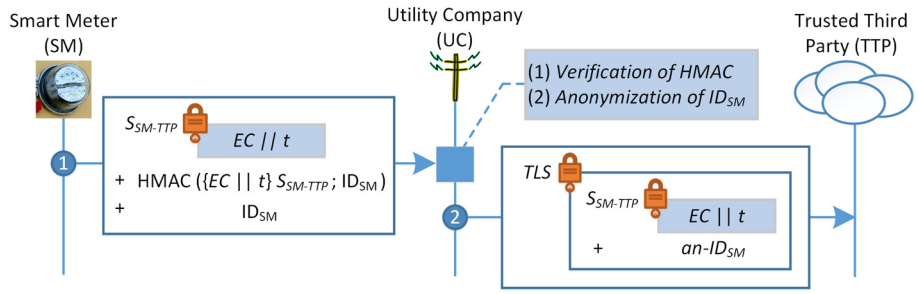

Fig. 6. Data transmission phase.

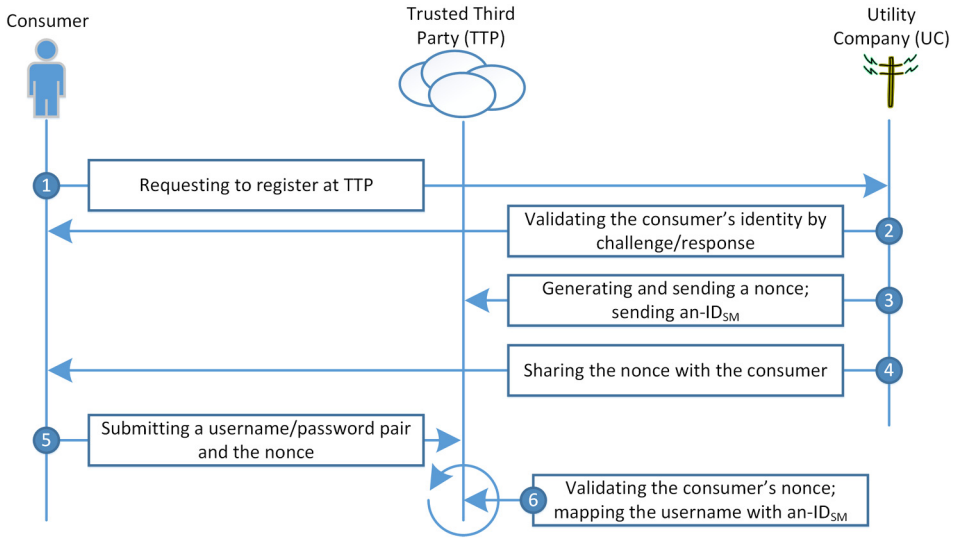

Fig. 7. Consumer authentication at TTP.

5.2. Authentication

Consumers, UC, and TTP are all involved in the authentication procedures. Consumers have to be provided with access to their fine-grained data at TTP. Therefore, they have to authenticate at TTP before accessing their energy measurements. Also, UC needs to bill consumers and have a capability to receive different reports from TTP including, for instance, an energy fraud analysis. Thus, UC has to be authenticated at TTP in order to fulfil its goals.

5.2.1. Utility company authentication at TTP

As part of the initialization process, UC has to create an account and register at TTP's web-service allowing UC to retrieve consumers' bills and fine-grained data analysis results. To register, UC has to provide all the necessary paperwork to TTP and sign a contract agreement. This is required by UC not only to receive billing information and energy readings analysis but also to relay messages between SM and TTP. In order to access TTP's web-service, UC has to pass a two-factor authentication process including verification of credentials and, for example, providing a physical token or smart-card for sending any requests via HTTPS connection in the Internet. An access control mechanism can be used at TTP for restricting UC's access to energy consumption data.

5.2.2. Consumer authentication at TTP

As SM data are stored at TTP's database, TTP has to provide consumers with access to those data and *EC* analysis tools. At the same time, consumer privacy has to be protected in such a way that TTP cannot link real consumer identity with *EC* data. Fig. 7 presents our scheme for consumers to authenticate at TTP.

In this scheme, consumers obtain different username/password pairs for UC and TTP. We assume that the communication among consumers, UC, and TTP is secured, using *TLS* or some other security protocol.

Consumer's authentication information is configured at the time of opening an account and signing a contract with UC. After successfully logging in at UC, the consumer enters his/her account preferences and requests UC to generate a nonce (a random number). The nonce needs to be shared between the consumer and TTP so that TTP could validate the consumer without knowing his/her real identity information. UC generates a nonce and stores it in the consumer's

account. UC authenticates at TTP with its own credentials via *TLS* and sends a message containing the consumer's nonce, its expiration date, timestamp, and $an\text{-}ID_{SM}$ linked with the consumer's SM. Consequently, the consumer sets up a secure connection with TTP over a wide area network and registers at TTP with a new username/password pair along with the nonce. If the nonce sent by the consumer matches the one sent by UC and it is not expired, TTP validates the consumer, adds him/her to the database, and maps the username with $an\text{-}ID_{SM}$ sent before by UC. After that, TTP removes the nonce from its storage. In this scenario, UC does not know the consumer's username/password pair used for authentication at TTP. Also, TTP cannot map any energy consumption information to the real consumer identity. However, TTP can validate the consumer by trusting UC, who authenticates the consumer's identity in a challenge/response form. If a consumer forgets his/her password, he/she has to repeat the steps of the proposed "consumer authentication at TTP" scheme because TTP is unaware of any information about the consumer except his/her username, password, and $an\text{-}ID_{SM}$.

5.3. Key management

Key revocation and renewal can be demonstrated with a few different scenarios. Let us assume, SM is compromised by a physical attack (the board with the TPM is removed in order to access it). In this case, UC will detect SM's physical penetration when that SM discontinues sending any parameters to UC. UC will send technicians to investigate SM's breach. When SM is replaced and the line is secure, SM can re-establish a connection with UC as described in Section 5.1.1. Afterwards, the initial session key exchange occurs between SM and TTP (Section 5.1.2).

Let us assume that $S_{SM\text{-}TTP}$ or $S_{SM\text{-}UC}$ needs to be renewed according to a security policy of UC or TTP (for example, the key must be renewed every 6 months). In this scenario, UC sends a request for key exchange to SM, encrypted with $S_{SM\text{-}UC}$. SM fulfils the same steps as described in Section 5.1 in order to update the necessary keys. A more detailed key management scheme is a part of our future work.

6. Protocol analysis

In this section, we analyze the proposed protocol in terms of security, privacy, and performance characteristics.

6.1. Compliance with secure design principles

The proposed model complies with the fundamental *secure design* principles introduced by Saltzer and Schroeder [28] for protection of systems. These principles include economy of mechanism, fail-safe defaults, complete mediation, open design, separation of privilege, least privilege, least common mechanism, and psychological acceptability. The proposed architecture follows the economy of mechanism as it deploys a simple three-tier infrastructure consisting of SMs, UC, and TTP. The fail-safe defaults principle is followed in the model by denying access to the fine-grained EC by default and allowing only authorized access to sensitive information. Every access to fine-grained data requires a two-factor authentication mechanism according to the complete mediation principle. The proposed protocol has an open design as it is publicly available. Separation of privilege is implemented in the proposed scheme as a required two-factor authentication. Also, separation of privilege is accomplished by separating access to the fine-grained data in such a way that consumers can retrieve their EC measurements and UC can request only the aggregated results. According to the least privilege principle, when a consumer or UC authenticates at TTP, he/she acquires a minimal set of access rights in accordance with his/her account type. Least common mechanism is followed by limiting shared resources among SMs, UC, and TTP to only the cryptographic keys necessary for secure communication. Last, psychological acceptability is attained by providing consumers with a simple yet secure way to access their fine-grained EC data.

6.2. Compliance with an ideal data collecting protocol

According to Jawurek et al. [29], an ideal protocol should allow consumers to choose arbitrary aggregation functions. Our proposed protocol provides a flexible way of defining aggregation functions at TTP. For example, consumers are allowed to gain access to their fine-grained data upon request. However, if a consumer desires to obtain the report about his/her EC history, he/she can query that information using the built-in functionality in accordance with the TTP's access control mechanism. On the other hand, UC has restricted access to consumer's meter readings and is only allowed to request daily/monthly aggregated EC data.

In the ideal situation, smart meters should operate asynchronously [29]. That is, each consumer should send EC to the aggregator independently from one another. In the proposed protocol, smart meters send energy readings absolutely independently. Requesting aggregated results for billing or other purposes does not require synchronous gathering.

6.3. Analysis of privacy and security goals

There are privacy, confidentiality, and integrity requirements for the proposed protocol. First, UC should not have access to the individual fine-grained meter readings but can only receive the aggregated values, computed bills, and results of the

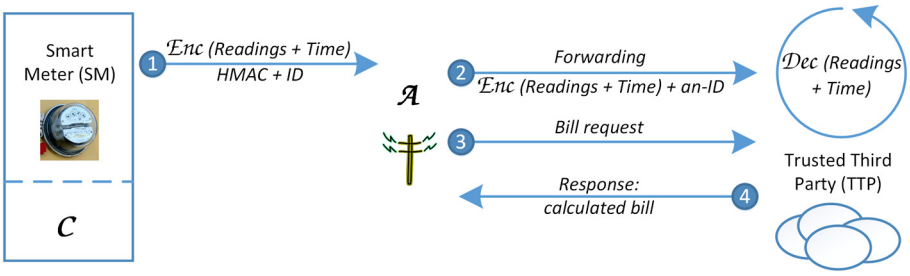

Fig. 8. UC attempts to learn private consumption data.

advanced *EC* analysis. Second, TTP should not learn whom the fine-grained data belong to, when it decrypts the meter readings forwarded by UC. Third, confidentiality has to be protected against man-in-the middle attacks and message integrity needs to be preserved.

The following theorem demonstrates that our proposed protocol satisfies the first requirement.

Theorem 1. *The proposed protocol preserves consumer privacy against a utility company.*

Proof. Let $\mathcal{A}$ (an adversary) be a utility company and $\mathcal{C}$ be a consumer. $\mathcal{A}$ is honest-but-curious, i.e., it always tries to learn as much as possible about $\mathcal{C}$. Let us now follow the steps of the proposed data transmission phase and trace the actions that $\mathcal{A}$ can impose against $\mathcal{C}$. SM gathers the meter readings of $\mathcal{C}$ and encrypts them with a session key $S_{SM\text{-}TTP}$, it forwards the encrypted message along with *HMAC* and ID_{SM} to $\mathcal{A}$ (Fig. 8, step 1). At this point, $\mathcal{A}$ verifies *HMAC* and tries to retrieve any other information from the message. Since SM encrypted (by using AES 128) the message with $S_{SM\text{-}TTP}$, which is known only to SM and TTP, $\mathcal{A}$ cannot decrypt the message and learn anything from it. As a result, $\mathcal{C}$'s privacy is preserved for this part of the protocol.

At the next step (Fig. 8, step 2), $\mathcal{A}$ forwards the message to TTP. After receiving the message, TTP decrypts it and stores the *EC* in the database. When $\mathcal{A}$ requests the bill from TTP (Fig. 8, step 3), it may attempt to masquerade as $\mathcal{C}$ and bypass the policy at TTP, which restricts access for $\mathcal{A}$. However, $\mathcal{A}$ cannot learn the credentials of $\mathcal{C}$ because $\mathcal{C}$ is anonymously registered at TTP according to the authentication procedure described in Section 5.2.2. Therefore, $\mathcal{A}$'s attempt to impersonate $\mathcal{C}$ fails. □

Another privacy requirement states that TTP cannot identify which consumer sent the fine-grained measurements for the analysis and storage. This requirement is satisfied as demonstrated in Theorem 2.

Theorem 2. *The proposed protocol preserves consumer privacy against a trusted third party.*

Proof. Let $\mathcal{A}$ (an adversary) be a trusted third party and $\mathcal{C}$ be a consumer. $\mathcal{A}$ is honest-but-curious, i.e., it always tries to learn as much as possible about $\mathcal{C}$. The first time when $\mathcal{A}$ may attempt to learn information about $\mathcal{C}$ is when it decrypts the data received from UC at the data transmission phase of the protocol (Fig. 9, step 3). Regardless of the fact that the fine-grained data are directly accessed by $\mathcal{A}$, SM's ID is anonymized by UC in advance (Fig. 9, step 2). As a result, rather than being linked to a particular consumer, the fine-grained measurements are associated with an anonymized ID, *an-ID*. Thus, $\mathcal{A}$ cannot learn any real information about $\mathcal{C}$ and therefore, the privacy of $\mathcal{C}$ is preserved.

There is one more scenario when $\mathcal{A}$ may attempt to link $\mathcal{C}$ to a particular set of fine-grained measurements. This scenario can be described as follows. Suppose that $\mathcal{C}$ authenticates at UC and signs into his/her personal account (Fig. 9, step 4). At this point, $\mathcal{C}$ has an opportunity to initiate registration at TTP for accessing the fine-grained data by making an appropriate request to UC. According to the authentication protocol described in Section 5.2.2, UC generates a nonce N and shares it between $\mathcal{C}$ and $\mathcal{A}$ (Fig. 9, step 6). In addition, UC shares an anonymized ID, $an\text{-}ID_{SM}$, with $\mathcal{A}$. Thus, up to this, point $\mathcal{A}$ does not know any information about $\mathcal{C}$ except that there is a nonce linked to the $an\text{-}ID_{SM}$. $\mathcal{C}$ continues following the protocol and provides $\mathcal{A}$ with the registration information containing username/password and the nonce received from UC (Fig. 9, step 7). $\mathcal{A}$ verifies the nonce and accepts the registration if it matches N and is not expired. After this authentication, $\mathcal{C}$ can access the fine-grained data linked to his/her account. As we followed the steps of the protocol, there is no sensitive information involved in the authentication process at $\mathcal{A}$. Therefore, the privacy of $\mathcal{C}$ is preserved as no private data are exposed to $\mathcal{A}$. □

Confidentiality and integrity of fine-grained readings is as important to protect as consumer privacy. Theorem 3 demonstrates that the proposed protocol supports these security requirements against a man-in-the-middle attacker.

Theorem 3. *The proposed protocol preserves confidentiality and data/origin integrity of fine-grained measurements.*

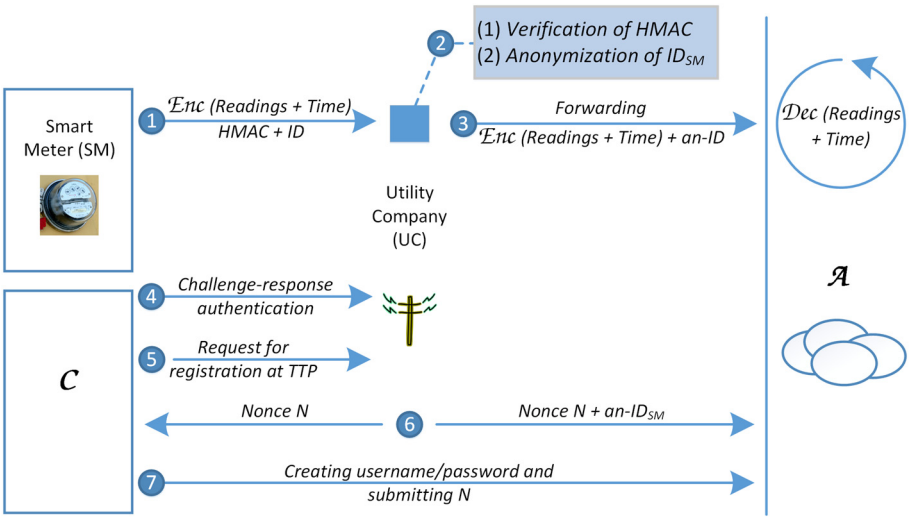

Fig. 9. TTP attempts to learn private consumption data. $Enc()/Dec()$ – AES 128 encryption/decryption functions using the $S_{SM\text{-}TTP}$ session key.

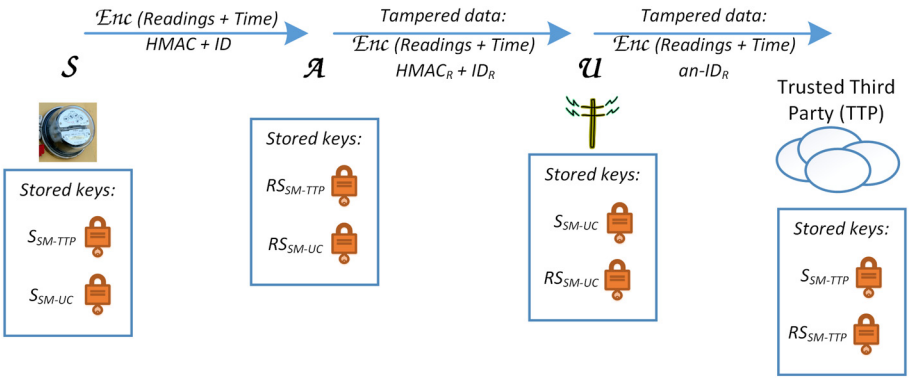

Fig. 10. Man-in-the-middle attack.

Proof. There are two cases where a Man-In-The-Middle (MITM) attack is possible: (1) MITM between SM and UC; and (2) MITM between UC and TTP. The second case can be prevented with utilization of a secure protocol like *TLS*. Therefore, the first case is of a primary concern.

Let us assume that an MITM adversary $\mathcal{A}$ attempts to tamper with the communication between a smart meter $\mathcal{S}$ and utility company $\mathcal{U}$ (Fig. 10). The confidentiality and consumer privacy are preserved by encrypting the energy measurements with a session key $S_{SM\text{-}TTP}$, therefore $\mathcal{A}$ cannot decrypt the energy readings because $\mathcal{A}$ does not possess the key.

Scenario 1 $\mathcal{S}$ sends $\mathcal{U}$ a message M containing *encrypted readings + timestamp*, *HMAC*, and ID. If $\mathcal{A}$ replaces ID with another legitimate but fake ID_F that is linked with a pre-shared key *fake-*$S_{SM\text{-}UC}$, then $\mathcal{U}$ will not be able to verify *HMAC* of the message M from $\mathcal{S}$ because the key associated with the fake ID_F is different from $S_{SM\text{-}UC}$. Verification of *HMAC* preserves message data/origin integrity. Therefore, the ID replacement attack fails.

Scenario 2 $\mathcal{A}$ has successfully registered a rogue smart meter $\mathcal{R}$ (Fig. 10), whereas $\mathcal{U}$ stores $\mathcal{R}$'s ID_R and $R_{SM\text{-}UC}$ (a pre-shared key between $\mathcal{R}$ and $\mathcal{U}$). At this point, $\mathcal{A}$ can substitute ID with ID_R and recalculate *HMAC* of M sent from $\mathcal{S}$. In this scenario, the message tampered by $\mathcal{A}$ will pass the verification by $\mathcal{U}$ because $\mathcal{U}$ will be able to find the key $R_{SM\text{-}UC}$ associated with ID_R. However, after $\mathcal{U}$ verifies the tampered message, $\mathcal{U}$ substitutes ID_R with a corresponding anonymized *an-*ID_R. Consequently, when TTP receives the tampered message from $\mathcal{U}$, TTP uses $R_{SM\text{-}TTP}$ for decrypting M sent from $\mathcal{S}$ and fails to do that because M is encrypted with $S_{SM\text{-}TTP}$ and thus cannot be decrypted with $R_{SM\text{-}TTP}$. Therefore, this type of an impersonation attack fails. □

The notion of using one symmetric key The proposed protocol uses the same key $S_{SM\text{-}UC}$ in both *HMAC* and AES encryption only to reduce the storage cost for SM. We believe that this is secure for the following reasons. First, both AES and *HMAC* are assumed to be intractable in terms of deriving a key given the encrypted message. Second, AES and *HMAC* do not

Table 2
Smart meter performance analysis summary.

Storage cost		96 bytes
Computational cost	Registration phase:	625 ms
	Session key exchange phase:	1,050 ms
Communication cost		Encrypted message
	+	ID_{SM}
	+	*HMAC*
	+	Timestamp
		$16 + 4 + 32 + 6 = 58$ bytes

interact with each other in any way and, therefore, it is secure to use the same key for two different purposes: integrity and confidentiality preservation. If two different keys are used, then if the smart meter is compromised, both keys will be compromised as well since they are stored in the same place and linked to the same SM. Therefore, it is clear that using two different keys does not make the scheme cryptographically stronger.

6.4. Performance analysis

This section evaluates the computational and communication costs as well as the storage cost for SM. The performance metrics for UC and TTP are not evaluated in this paper as modern computers can efficiently compute modular exponentiations and multiplications (UC and TTP are not restricted with computational power). Our primary focus is the performance of SM as its computing capabilities are very limited. The results of SM's overheads are summarized in Table 2.

6.4.1. Storage cost

Storage cost can be computed as a summation of the session key $S_{SM\text{-}TTP}$ (32 bytes), $S_{SM\text{-}UC}$ (32 bytes) and two Initial Vectors (each 16 bytes) used in AES. Both keys have to be stored all the time at SM to conduct normal operations.

6.4.2. Communication cost

The communication cost is based on an encrypted meter measurement M that is *16* bytes (*for padding*) + *length*(M) bytes total, timestamp (6 bytes), *HMAC* (32 bytes), and ID_{SM} (4 bytes). To calculate *length*(M), the average *EC* in the U.S. will be taken into account. From the statistics of U.S. Energy Information Administration [30], the average power consumption per household per month is 1,273 kWh. In other words, it is 30 Wh per minute or 450 Wh per 15 minutes. Let us assume that SM must report the energy readings in the time interval varying from 1 to 15 minutes. Therefore, in order to have enough memory to store *EC* value M gathered during several time units, we need to allocate up to 16 bits assuming that in peak hours the consumption of a household/business may raise to several kWh. As a result, the total communication cost will be 58 bytes.

6.4.3. Computational cost

SM uses computationally efficient symmetric encryption (AES 128) to secure its meter readings. NISTIR 7628 Guidelines for Smart Grid Cyber Security [19] states that AES 128 is acceptable for use in the Smart Grid. In our architecture, computationally expensive public key exchange is only performed at the initial stage of the proposed protocol.

At the registration phase, SM obtains a *partial* public/private key pair from UC without experiencing any computational overhead. For verifying the UC's message and generating its full public/private keys, SM has to fulfil three modular exponentiations, one hash operation, one multiplication, and one random number generation.

At the session key exchange phase, SM has to encrypt the message containing a request for TTP to generate a shared secret key. For performing this operation, SM performs four hash operations and modular exponentiations, one multiplication, and two random number generations. Upon receiving an encrypted secret key $S_{SM\text{-}TTP}$ from TTP, SM decrypts it by utilizing two hash operations and three modular exponentiations.

Due to the fact that we apply efficient AES 128 encryption, the computational cost for regular energy measurement encryption at the data transmission phase is minimized.

Let us calculate the computational expenses based on the aforementioned cost breakdown. According to NIST Special Publication 800-56A [31], the private key length for the public key cryptography should be 256 bits and the public key length – 2,048 bits. TPM can compute a 2,048-bit RSA signature in 200 ms [32]. Also, generating a 2,048-bit random number is no slower than generating a 2,048-bit RSA signature. Thus, a 256-bit random number can be generated in no more than 25 ms. As a result, SM can fulfil the operations at the registration phase in no more than 625 ms. The SM's total computational cost for the session key exchange phase is up to 1,050 ms. Registration and session key exchange phases are only one-time operations and they do not introduce more overhead into SM's computing cost at the data transmission phase.

6.4.4. Implementation of the proposed scheme

We implemented the proposed privacy-preserving protocol and measured its performance on Intel(R) Core(TM) i5-2430M at 2.40 GHz, 6 GB RAM, utilizing a free cryptographic library Crypto++ 5.6.3 [33]. We modeled core functionality of UC, SM,

Table 3
Implementation analysis of the proposed protocol.

Protocol's phase	Computational cost (ms)	Storage cost (Bytes)
Registration phase	SM: 4 ms UC: 618 ms TTP: 4 ms	SM: 1376 B UC: 832 B TTP: 1376 B
Session key exchange	SM: 3.3 ms UC: – TTP: 4.7 ms	SM: 16 B UC: – TTP: 16 B
Data transmission phase	SM: 2.7 ms UC: – TTP: 1.1 ms	SM: 58 B UC: – TTP: 10 B

and TTP units in C++ classes in order to determine their storage and computational costs during all phases of the protocol. Most importantly, we focused on SM's overheads as it is the bottleneck of the AMI with regard to its low computational capabilities. As we can observe in Table 3, SM can fulfil the necessary steps of the one-time initialization and session key exchange phases within 7.3 ms. The key to efficiency analysis of the protocol are the encryption and *HMAC* generation of the data transmission phase, which altogether take only 2.7 ms for SM to finish due to the efficient AES 128 encryption.

SM's computational capabilities are several factors inferior to the system utilized for the above-mentioned performance analysis. However, we believe that SM can efficiently perform the operations according to the computational analysis in Section 6.4.3.

6.5. Analysis of usability goals

For consumers, the proposed protocol offers flexibility in using services of UC. Suppose that a consumer changes his/her home address from house A (energy is provided by UC_1) to house B (energy is provided by UC_2). First, the consumer obtains $an\text{-}ID_{SM}$ from UC_1. Then, B's ID_{SM} is updated with a new ID, ID_{new}, which is generated by UC_2 providing energy to B. UC_2 sends TTP a request to replace $an\text{-}ID_{SM}$ (associated with the consumer) with ID_{new}. When TTP receives the request from UC_2, it creates a note in consumer's account that a new UC requested to change his/her ID. When the consumer enters his/her account, he/she will be given a choice to accept the change. After accepting it, the consumer can continue using the service and UC_2 can fulfil its responsibilities, whereas UC_1 can no longer request information about consumer's *EC*.

Consumers can compare multiple utility companies' prices (*retail choice*) based on their personal *EC* in order to select the best available UC according to their needs. This is achieved by TTP having direct access to the fine-grained data. TTP can gather information about prices of different utility companies and generate a report based on consumer's *EC* data.

UC can bill consumers using time-of-use tariffs. In order to establish time-of-use billing, UC can send the price ranges for different time periods to TTP that can calculate the precise bill based on consumer fine-grained energy readings.

6.6. Protocol analysis discussion

The following is a discussion of additional attack vectors and how the proposed protocol addresses them.

Scenario 1 Suppose that a rogue smart meter SM_R attempts to transfer all *EC* to victim's SM_V. SM_R sends the energy readings to TTP via UC and changes its ID_R to ID_V of SM_V. When the encrypted message arrives at UC, UC verifies *HMAC* of the message and fails to do that with the key linked to SM_V because *HMAC* is computed by using ID_R. Therefore, the rogue SM cannot transfer its *EC* to another SM. In case if SM_R recalculates *HMAC* based on ID_V, then TTP will detect the tampered message as described in the man-in-the-middle attack of Section 6.3, Theorem 3.

Scenario 2 Suppose that UC attempts to overbill consumers. In the proposed protocol, the consumers will detect such a fraudulent activity due to their ability to independently verify the bill calculations at TTP. Consumers can anonymously authenticate at TTP as described in Section 5.2.2 and receive their bills by requesting TTP to calculate them based on the time-of-use rates. Then consumers can compare the bills received from TTP with the bills received from UC and any mismatch will indicate tampering by UC.

Scenario 3 For the intention of committing forgery and stealing energy, a consumer might intercept the data between his/her assigned SM and UC to implement a replay attack where the same small amount of consumed energy is reported all the time. To thwart this, SM includes a timestamp, capturing the time when the *EC* measurement was gathered.

Scenario 4 UC may attempt to execute a new type of attack that we define as a *wait-for-response man-in-the-middle* attack (Fig. 11). UC tries to learn the fine-grained readings by sequentially requesting bills from TTP after each measurement and reversing those bills to calculate the consumed energy as the following. Let us assume that UC forwarded the energy readings to TTP up until time t_0. SM sends the data to UC at time t_1. UC forwards it to TTP. SM sends the data at time t_2.

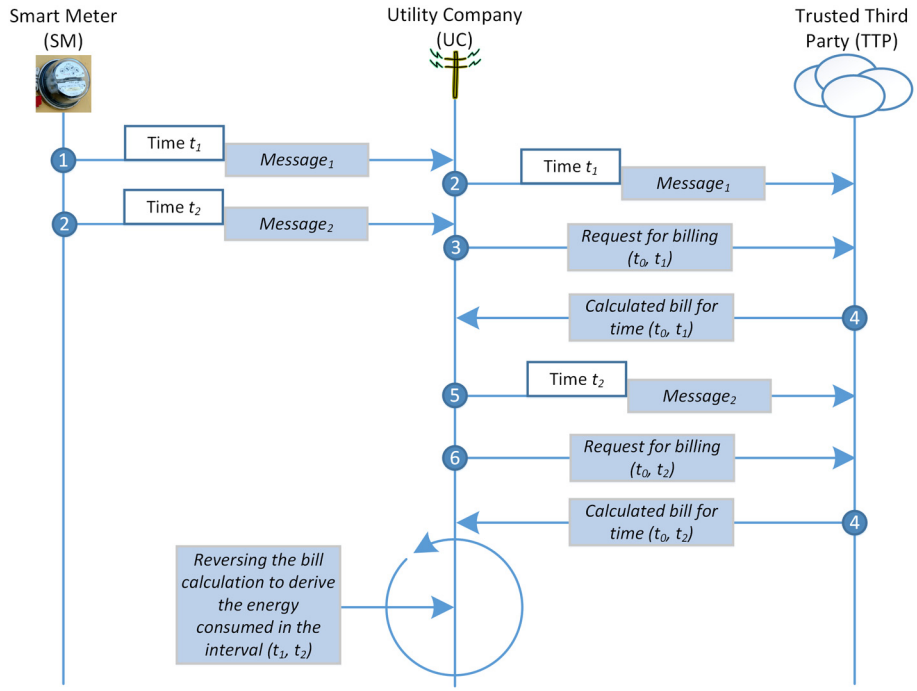

Fig. 11. Wait-for-response attack.

UC holds the data and sends a request to TTP to bill the consumer for the period from t_0 to t_1. TTP responds back with the bill for that time interval. UC forwards TTP the data for time t_2, requests the bill again, and waits for another billing response from TTP before it sends the next readings. TTP responds back to UC with the bill for the time period from t_0 to t_2 (or from t_1 to t_2, depending on UC's request). UC receives several consecutive bills for short time intervals and learns the fine-grained measurements by reversing the calculation of the bills obtained from TTP. To mitigate this kind of attack, TTP can set the minimum billing period requirements, for example, 1 week or more. The aggregated data over a 1-week period will not reveal fine-grained measurements and consumer privacy will be preserved.

7. Related work

This section introduces related work as well as comparison of the proposed model with the existing approaches.

7.1. Privacy-preserving solutions

Privacy-preserving protocols have drawn a lot of attention in recent AMI research work. However, many of those protocols do not consider storing and using the fine-grained data for an advanced analysis. Others do not take into account different facts, such as (i) price ranges can be different for different time intervals, (ii) consumers may need the opportunity to compare UCs for selecting the best choice and reducing their energy consumption cost, and (iii) consumers may need to have access to their fine-grained energy consumption history. In addition, as required by some of these existing solutions, a smart meter connectivity to the Internet opens up new opportunities for malicious activities from the outside world.

Jawurek et al. [13] proposed a privacy-preserving protocol, which enables billing with time-of-use tariffs without disclosing the actual consumption profile to the supplier. A plug-in device intercepts the meter readings and, upon receiving the tariff information over the Internet, provides the calculated bill to an energy service provider. This approach uses a zero-knowledge proof based on the Pedersen Commitment. However, it does not provide access to stored fine-grained data for advanced analysis.

Lin et al. [14] proposed a solution for privacy preserving billing and load monitoring applications by using a Trusted Platform Module (TPM) and a semi-trusted storage system. Household meters encrypt the energy readings and store them in the storage system. Electric service providers and load monitoring centers receive aggregated energy consumption results from the storage system upon request. This scheme has some limitations. To receive an aggregated monthly bill, the utility company has to request random numbers from the meter in order to decrypt the aggregated data. This may cause security problems in some cases, such as in the event of a man-in-the-middle attack.

Ruj et al. [15] proposed an integrated architecture for smart grids. Their solution uses homomorphic encryption for data aggregation in home area networks, building and neighborhood area networks that report to a substation. The energy consumption data are stored in substations and access to the data is managed by an access control scheme. Depending on

Table 4
Comparison of privacy-preserving approaches.

Research work	1	2	3	4	5	6
Jawurek et al. [13]	X		X			
Lin et al. [14]	X					X
Ruj et al. [15]	X				X	
Joye and Libert [16]	X					
Rial and Danezis [17]	X		X		X	
Our work	X	X	X	X	X	X

the user role, the user can get restricted access to the stored data. Users can be maintenance units, utility centers, pricing estimator units or analysis and prediction groups. However, in their approach, energy fraud issues and access to fine-grained energy consumption data are not addressed.

A disadvantage of homomorphic encryption is that having access to the private key, an aggregator can begin aggregating data starting from the second measurement. Upon receiving the next energy consumption value, it can add it to the previously aggregated result S, and then subtracting S from the newly computed summation will result in access to the fine-grained data with the new encrypted measurement. However, according to [15], if an aggregator is only required to perform the multiplication operation and send the result to the utility company, then privacy is protected.

Joye and Libert [16] proposed a solution for a private data analysis using an untrusted third party aggregator. Their scheme allows evaluating the sum of consumer energy consumption without revealing any private information. The proposed solution claims to be well-suited for low-power devices like SM. However, this work [16] does not solve the problem of storing fine-grained energy consumption data for further analysis, while at the same time preserving consumer privacy.

Rial and Danezis [17] developed a privacy-preserving scheme, where smart meters produce certified readings and send them to consumers, who calculate their bills and provide them to utility companies within web technologies or via a trusted party. Their protocol supports time-of-use tariffs, preserving consumer privacy. However, such an approach does not provide consumers with fine-grained advanced analysis (for instance, consumer advisement about reducing their bill and comparison of different energy providers based on consumer's historical data). Also, utility companies cannot fulfil privacy-preserving fine-grained data analysis like fraud detection. The data are accessible only for consumers themselves and there are no means to accomplish the above-mentioned tasks by providers. Their proposed scheme makes it difficult to distribute the symmetric key among consumers and meters. In case if consumers do not have access to the Internet, there is no proposed solution to calculate the bill without their consent. The scheme requires consumers to fulfil the billing tasks and it might not be practical to do that on a monthly basis.

Fournet et al. [34] proposed a powerful ZQL query language for making privacy-preserving queries over data stored in tables. ZQL can contribute to smart metering protocols with generating complex queries to calculate consumer bills using public tariffs and data lookup tables.

To the best of our knowledge, existing privacy-preserving research applying homomorphic encryption and aggregation techniques does not address energy consumption analysis such as energy fraud detection.

Some previously proposed AMI infrastructures [14,16] link the smart meters directly with TTP that gathers all data. We believe such an approach makes the network less scalable and secure. In addition, every UC would need to have a separate TTP. An alternative solution can be a centralized TTP, which can be managed by a corporate cloud organization providing computational and storage resources. However, for security reasons, the less number of entities that are directly connected to smart meters, the better. In this scenario, when TTP gathers data from smart meters, both UC and TTP have access to the meters. Consequently, there are more ways for malicious actors to access meter data in such architectures.

In our proposed approach most of the above-mentioned limitations are addressed. Although TTP is considered to be a centralized hub, it gathers all the data through UC via the Internet, thus reducing the network load on TTP and providing scalability for UC. UC is responsible for its own AMI and operates as a "black box" for TTP. TTP does not have direct access to SMs and, for this reason, it reduces the number of attack vectors in comparison to an architecture where TTP resides between SMs and UC.

7.2. Comparison with existing approaches

In Table 4 we compare different privacy-preserving approaches for AMI with our proposed work. The following list represents the header for the Table 4.

1. Consumer privacy preservation for billing.
2. Secure access to privacy-preserved fine-grained data for further analysis: different applications, such as energy fraud detection and consumer energy consumption advisement, can utilize consumer fine-grained data without revealing any sensitive information.
3. Supporting time-of-use billing depending on different daytime price ranges.
4. Supporting the *retail choice* [18]: it is important to allow consumers to switch between energy providers without losing their preceding energy consumption measurements and analysis.

Table 5
Smart meter's overhead comparison of privacy-preserving schemes on 100 energy consumption readings.

Research work	Computational cost (ms)	Communication cost (Bytes)
Jawurek et al. [13]	132 ms	15200 B
Lin et al. [14]	707 ms	N/A
Rial and Danezis [17]	500 ms	26000 B
Our work	270 ms	5800 B

5. Utility-only access to SMs: it drastically decreases the number of attack vectors because SMs are controlled only by utility companies.
6. Minimal processing overhead for SMs: all expensive calculations, including billing and fraud detection, are made by TTP outside of the Grid.

We also compared the communication and computational overheads of the proposed protocol's data transmission phase with the corresponding phases of other privacy-preserving schemes [13,14,17]. As demonstrated in Table 5, the proposed protocol shows efficiency with its performance characteristics that are comparable to others. In addition, it is distinguishable from other schemes by the combination of its unique features including the fine-grained data analysis (such as fraud detection) and *retail choice*.

8. Conclusion and future work

The proposed privacy-preserving scheme includes smart meters, a utility company, and a trusted third party. Smart meters are connected with the utility company and the trusted third party handles all requests and computations for data analysis and time-of-use billing. The protocol is shown to be secure against different types of attacks, protecting confidentiality, integrity, and consumer privacy.

Consumers can utilize a *retail choice* as well as retrieve a fine-grained data analysis from the trusted third party in a privacy-preserving manner. The utility company can only request billing information and the final results of the advanced analysis, such as an energy fraud detection report.

A more detailed key management scheme will follow in our future work. We are also working on relaxing the assumption that the trusted third party does not collude with the utility company.

Acknowledgments

This work is supported by the Center for Energy Systems Research of Tennessee Tech University. We are thankful to Michael Pyle (Chief Security Officer and VP of Cyber Security at Schneider Electric, Nashville, TN), Glen Chason (Senior Technical Lead Cyber Security and Privacy/Cyber Security Research Lab Manager at Electric Power Research Institute, EPRI, TN), and Ed Beroset (Principal Technical Leader at EPRI, CA) for their invaluable comments and feedback. We are also thankful to Daniel Tyler for his contribution in the performance analysis.

References

[1] B. Jiang, Y. Fei, Smart home in smart microgrid: a cost-effective energy ecosystem with intelligent hierarchical agents, IEEE Trans. Smart Grid 6 (1) (2015) 3–13, http://dx.doi.org/10.1109/TSG.2014.2347043.
[2] P. Cottone, S. Gaglio, G.L. Re, M. Ortolani, User activity recognition for energy saving in smart homes, Pervasive Mob. Comput. 16 (Part A) (2015) 156–170, http://dx.doi.org/10.1016/j.pmcj.2014.08.006, http://www.sciencedirect.com/science/article/pii/S1574119214001382.
[3] I. Monedero, F. Biscarri, C. Leon, J. Biscarri, R. Millan, Midas: detection of non-technical losses in electrical consumption using neural networks and statistical techniques, in: Computational Science and Its Applications, ICCSA 2006, in: Lecture Notes in Computer Science, vol. 3984, Springer, Berlin, Heidelberg, 2006, pp. 725–734.
[4] P. Keung, J. Karel, C. Bright, Neural networks for insurance fraud detection, https://cs.uwaterloo.ca/~cbright/reports/STAT840-project.pdf, 2009.
[5] V. Ford, A. Siraj, W. Eberle, Smart grid energy fraud detection using artificial neural networks, in: IEEE Symposium on Computational Intelligence Applications in Smart Grid, CIASG, 2014, pp. 1–6.
[6] V. Ford, A. Siraj, Clustering of smart meter data for disaggregation, in: IEEE Global Conference on Signal and Information Processing, GlobalSIP, IEEE, 2013, pp. 507–510.
[7] J. Liu, Y. Xiao, S. Li, W. Liang, C.L.P. Chen, Cyber security and privacy issues in smart grids, IEEE Commun. Surv. Tutor. (2012) 981–997, http://dx.doi.org/10.1109/SURV.2011.122111.00145.
[8] S.S.S.R. Depuru, L. Wang, V. Devabhaktuni, N. Gudi, Smart meters for power grid — challenges, issues, advantages and status, in: IEEE/PES Power Systems Conference and Exposition, 2011.
[9] M. Bae, K. Kim, H. Kim, Preserving privacy and efficiency in data communication and aggregation for AMI network, J. Netw. Comput. Appl. 59 (2016), http://dx.doi.org/10.1016/j.jnca.2015.07.005.
[10] C. Jie, J. Shi, Y. Zhang, EPPDC: an efficient privacy-preserving scheme for data collection in smart grid, IEEE Commun. Surv. Tutor. 2015 (2015), http://dx.doi.org/10.1155/2015/656219.
[11] F. Li, B. Luo, P. Liu, Secure information aggregation for smart grids using homomorphic encryption, in: 2010 First IEEE International Conference on Smart Grid Communications, SmartGridComm, 2010, pp. 327–332.
[12] X. Huang, Y. Xiang, E. Bertino, J. Zhou, L. Xu, Robust multi-factor authentication for fragile communications, IEEE Trans. Dependable Secure Comput. 11 (6) (2014) 568–581.

[13] M. Jawurek, M. Johns, F. Kerschbaum, Plug-in privacy for smart metering billing, in: Privacy Enhancing Technologies, in: Lecture Notes in Computer Science, vol. 6794, Springer, Berlin, Heidelberg, 2011, pp. 192–210.
[14] H.-Y. Lin, W.-G. Tzeng, S.-T. Shen, B.-S. Lin, A practical smart metering system supporting privacy preserving billing and load monitoring, in: Applied Cryptography and Network Security, in: Lecture Notes in Computer Science, vol. 7341, Springer, Berlin, Heidelberg, 2012, pp. 544–560.
[15] S. Ruj, A. Nayak, A decentralized security framework for data aggregation and access control in smart grids, IEEE Trans. Smart Grid 4 (1) (2013) 196–205, http://dx.doi.org/10.1109/TSG.2012.2224389.
[16] M. Joye, B. Libert, A scalable scheme for privacy-preserving aggregation of time-series data, in: Financial Cryptography and Data Security, in: Lecture Notes in Computer Science, vol. 7859, Springer, Berlin, Heidelberg, 2013, pp. 111–125.
[17] A. Rial, G. Danezis, Privacy-preserving smart metering, in: ISSE Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe Conference, 2012, pp. 105–115.
[18] U.S. Energy Information Administration, Can customers choose their electricity supplier?, http://www.eia.gov/tools/faqs/faq.cfm?id=627&t=3.
[19] NIST Smart Grid, Introduction to NISTIR 7628 guidelines for smart grid cyber security, http://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf, 2014.
[20] X. Fan, G. Gong, Security challenges in smart grid and control systems, http://timreview.ca/article/702, 2013.
[21] Y. Liu, P. Ning, M.K. Reiter, False data injection attacks against state estimation in electric power grids, ACM Trans. Inf. Syst. Secur. 14 (1) (2011) 13, http://dx.doi.org/10.1145/1653662.1653666.
[22] S. McLaughlin, D. Podkuiko, S. Miadzvezhanka, A. Delozier, P. McDaniel, Multi-vendor penetration testing in the advanced metering infrastructure, in: Proceedings of the 26th Annual Computer Security Applications Conference, 2010, pp. 107–116.
[23] Trusted Computing Group, Trusted platform module (tpm) summary, http://www.trustedcomputinggroup.org/files/resource_files/4b55c6b9-1d09-3519-ad916f3031bcb586/trusted platform module summary_04292008.pdf, 2008.
[24] Y. Sun, F. Zhang, J. Baek, Strongly secure certificateless public key encryption without pairing, in: Cryptology and Network Security, in: Lecture Notes in Computer Science, vol. 4856, Springer, Berlin, Heidelberg, 2007, pp. 194–208.
[25] S.S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, in: Advances in Cryptology, ASIACRYPT 2003, in: Lecture Notes in Computer Science, vol. 2894, Springer, Berlin, Heidelberg, 2003, pp. 452–473.
[26] Information Technology Laboratory, NIST, The keyed-hash message authentication code (HMAC), FIPS PUB 198-1, http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf, 2008.
[27] R. Housley, Rfc 5652: cryptographic message syntax, https://tools.ietf.org/html/rfc5652, 2009.
[28] J.H. Saltzer, M.D. Schroeder, The protection of information in computer systems, 1975.
[29] M. Jawurek, F. Kerschbaum, G. Danezis Sok, Privacy technologies for smart grids – a survey of options, 2012.
[30] U.S. Energy Information Administration, How much electricity does an American home use?, http://www.eia.gov/tools/faqs/faq.cfm?id=97&t=3, 2015.
[31] Recommendation for pair-wise key establishment schemes using discrete logarithm cryptography, NIST Special Publication 800-56A Revision 2, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf, 2013.
[32] Trusted Computing Group, A closer look at Atmel's trusted platform module (tpm), http://blog.atmel.com/2013/07/29/a-closer-look-at-atmels-trusted-platform-module-tpm, 2013.
[33] W. Dai, A free C++ class library of cryptographic schemes, http://cryptopp.com/, 2015.
[34] C. Fournet, M. Kohlweiss, G. Danezis, Z. Luo, Zql: a compiler for privacy-preserving data processing, in: 22nd USENIX Conference on Security, 2013, pp. 163–178.