A protocol for Metering Data Pseudonymization in Smart Grids

Cristina Rottondi†∗, Giulia Mauri, and Giacomo Verticale

ABSTRACT

A tradeoff between data collection needs and user privacy is of paramount importance in the Smart Grid. This paper proposes a pseudonymization protocol for data gathered by the Smart Meters, which relies on a network infrastructure and a dedicated set of nodes, called Privacy Preserving Nodes (PPNs). The network privacy is enforced by a separation of duties: the PPNs perform data pseudonymization without having access to the measurements, which are masked by means of a secret sharing scheme, while the entities accessing the data recover and relate the plain measurements generated by the same meter along a time window of finite duration, but have no access to the meter identities.

The paper also provides an evaluation of the security and of the performance of the protocol, comparing it to the two alternative encryption techniques which mask the measurements by means of the Chaum Mixing scheme or of an Identity Based Proxy Re-Encryption scheme. Copyright © 2012 John Wiley & Sons, Ltd.

∗Correspondence

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano, Italy. Email: rotondi@elet.polimi.it

1. INTRODUCTION

Many countries are modernizing their power grid into a more stable and efficient “Smart Grid”, which is expected to be reliable, scalable, manageable, and extensible, but also secure, interoperable, and cost-effective. Therefore, the new grid is a very complex ecosystem involving both information technology and power grid operations and governance.

The Smart Grid can capture and analyze data regarding power usage and generation in near-real time, providing forecasts and recommendations to the customers regarding the optimization of their consumption. Such a system presents many privacy and cyber-security challenges.

In fact, the Smart Meters collect energy consumption data with high frequency, improving the quality of the information available to enhance electricity provision, adding value to services for customers, and improving billing. However, this can also result in a violation of people’s privacy, since users’ personal habits and customs can be inferred by analyzing energy consumption data gathered by the meters. Therefore, numerous governmental authorities and standardization bodies have emanated rules and restrictions about the treatment and diffusion of meter readings: for example, NIST [1] mandates that, unless strictly necessary, metering data should be anonymized in order to prevent utilities and third parties from linking the collected information to the identity of the customers that generated them. A recent Act of the USA Committee on Homeland Security [2] imposes that procedures for the anonymization of cyber-information must be defined in order to make such information...
available to external parties, e.g. for academic research or actuarial purposes. Depending on the specific contexts and applications, data anonymization can be achieved through different approaches [3], ranging from generalization (where information is coarsened into representative sets) to perturbation (where data are polluted by means of noise addition), pseudonymization (which replaces the individuals’ true identities with pseudonyms), and aggregation (which releases cumulative data computed on the information provided by multiple individuals, so that the contribute of a single entity is no longer identifiable in the aggregated data).

This paper proposes a metering data pseudonymization protocol (first introduced in [4]) that allows multiple External Entities (such as utilities, third parties, and service providers) to obtain disaggregated data generated by the Smart Meters. Data maintain their temporal sequentiality along a time window of finite duration, but the protocol does not allow the association between the data and the identity of the meter that generated them. To do so, we rely on a privacy-preserving infrastructure which introduces a set of Privacy Preserving Nodes (PPNs) in the Smart Grid architecture. The infrastructure shares similarities with the one we defined in [5] for anonymization through aggregation. In fact, the same set of PPNs could provide both aggregation, as described in [5], and pseudonymization, as described in this paper. Note that the PPNs allow a considerable reduction of the computational effort required to the energy Meters, which are usually resource-constrained low cost devices, with limited computational capabilities. The PPNs can be operated in a centralized fashion by independent parties or regulation authorities, as envisioned by different research and standardization bodies. For example, a recent proposal of the California Public Utilities Commission [6] speculates the realization of Energy Data Centers aimed at the collection and dissemination of aggregated and/or anonymized energy consumption data and run by governmental entities. Collecting Centers gathering data from local units deployed at household level in a GSM/GPRS-based AMI infrastructure have been demonstrated by several utilities in the Shandong Province of China [7]. While such Data Centers are assumed to be fully trusted, our proposed architecture ensures no violation of the customers’ privacy even in presence of “honest-but-curious” collectors, thus not requiring full trustability. However, the functionalities of the PPNs could also be performed by nodes already existing in the Smart Grid scenario: a hierarchical Smart Metering System architecture providing a non-trusted \( k \)-anonymity service based on Gateways located at the customers’ premises and mandated by the German Federal Office for Information Security [8] has been discussed in [9]. A fully distributed peer-to-peer network for AMI using Smart Meters realized with off-the-shelf hardware and relying on existing communication infrastructure has recently been proposed by a German technology provider [10].

Though customer pseudonymized data cannot be used for billing purposes, nevertheless their release would be highly beneficial for numerous entities with a wide variety of scopes. For example, vendors and societies conducting target marketing should not have access to user-related data but only to anonymized measurements, since energy load profiles could for example be used to track the usage of particular electrical appliances [11]. Another relevant example is the analysis of the anonymized energy usage patterns aimed at monitoring the distributed generation capabilities of the customers, the rate impact within/across climate zones and other relevant parameters to evaluate the efficiency of the Smart Grid [6].

It is worth noting that our privacy-preserving infrastructure can also support spatial and/or temporal aggregation of metering data according to the needs expressed by the External Entities, which we investigated in [5]. Moreover, as it will be detailed in the remainder of the paper, our framework can be easily integrated with data perturbation and obfuscation techniques. Therefore, we can state that our proposed infrastructure can combine three different methods to perform data anonymization, in order to increase the resiliency of the system to different categories of de-anonymization attacks to smart metering data (as described e.g. in [12]). This paper gives the following contributions:

1. Defines a set of security properties, which capture the needs of the Smart Grid scenario.
2. Proposes a pseudonymization cryptosystem for frequent re-pseudonymization and multiple External Entities.
3. Describes a network architecture and a communication protocol for pseudonymizing metering data and analyzes how they satisfy the stated security properties.
4. Compares different cryptographic approaches for preventing the network from accessing the metering data and shows that the Shamir Secret Sharing (SSS) scheme is the only one compatible with real-time operations.

We compare our protocol with two alternative very popular encryption techniques: the Chaum Mixing scheme [13–16] and the Identity Based Proxy Re-Encryption scheme [17–23], which have been proposed by the research community and applied in numerous scenarios, ranging from web browsing to data storage and sharing.

The remainder of the paper is structured as follows: Section 2 provides an overall view about data pseudonymization and anonymization in various contexts of communication networks, focusing on the Smart Grid scenario. Section 3 recalls some background notions. Section 4 introduces the pseudonymization framework and its possible deployment in the Smart Metering system, while the pseudonymization cryptosystem and its implementation are discussed in Section 5. Section 6 describes the communication protocol relying on Shamir Secret Sharing (SSS) scheme and compares it to two alternative cryptographic schemes, while Section 7 discusses the security guarantees the protocol achieves. The security assessment and performance evaluation are provided in Section 8. Concluding remarks are left for the final Section.

2. RELATED WORK

The problem of metering data pseudonymization in the context of Smart Grids has recently attracted the interest of several researchers. Efthimiou and Kalogridis [24] describe a method for the anonymization of electrical metering data sent by smart meters. They propose to separate high frequency and low frequency data and to assign an identity to each set of measurements: the High Frequency ID is anonymous, while the Low Frequency ID is attributable. The association between the two IDs is prevented by inserting long random intervals during the system setup. This solution has the drawback of requiring a long setup time and of hard-coding the IDs in the smart meter itself.

Jawurek et al. [12] develop two attacks to the privacy of pseudonymized consumption traces: the first is used to link an identity to a consumption trace by anomaly correlation, while the second links different pseudonyms of a customer by identifying common patterns in electricity consumption. The authors also analyze three countermeasures based on data aggregation, frequent re-pseudonymization and privacy preserving techniques, and provide numerical values for the correct tuning of the time aggregation ad re-pseudonymization windows. In particular, they state that raising the time aggregation windows from 3 to 24 hours causes a decrease in the accuracy achieved through behavioral patterns linking from 70% to 4%, while re-pseudonymization must be performed every one or very few days to obtain an accuracy below 50% (20 days long intervals lead to accuracy levels above 80%). Our protocol allows the choice of an arbitrarily low re-pseudonymization time window.

A privacy preserving protocol is presented by Rial and Danezis [25]. In this scheme, the meter outputs certified readings of measurements using cryptography; the user combines those readings with a certified tariff policy to produce the final bill. A zero-knowledge protocol ensures the correctness of the bill. The proposed protocol guarantees integrity and privacy and can perform the secure computation of a generic additive function, while our protocol does not perform any elaboration on the collected data but addresses the issue of replacing the identities of the subjects generating the measurements with unattributable pseudonyms.

Stegelmann and Kesdogan [9] analyze the issues related to anonymity and pseudonymity within the German BSI’s Protection Profile. The authors identify several problems and propose GridPriv, an architecture with a non-trusted k-anonymity service that allows to overcome the challenges. In particular, they consider churning attacks that a service provider can perform to determine an anonymity set, i.e. a set of Gateways which can be the data’s originator. Their architecture ensures anonymity within a set of a certain size k, whereas our protocol guarantees pseudonymity for all the Smart Meters generating the data.

Privacy protection is an important topic also in other contexts, from mobile ad hoc networks (MANETs) to RFID systems and health-care. Public-key based solutions have been proposed to guarantee communication anonymity, which means that the sender’s and receiver’s identities are hidden to external observers. Zhang et al. propose in [26] a pairing-based anonymous on-demand
A protocol for Metering Data Pseudonymization in Smart Grids

C. Rotondi, G. Mauri, and G. Verticale

routing protocol: in this approach, a Trust Authority (TA) administers the anonymous communication system by providing each node with a sufficiently large set of collision resistant pseudonyms, so that each node can dynamically change its pseudonym, and communicate the set of system parameters to each Anonymous User (AU). Though the protocol guarantees sender anonymity, receiver anonymity and relationship anonymity, the communications are not anonymous to the TA. To solve this problem, Huang proposes in [27] pairing-based encryption/decryption, key exchange, blind certificate and revocation schemes for anonymous communications. The drawback of this solution is the high computational cost to compute pairings.

A game-theoretic approach to anonymous networking in the context of wireless networks is proposed by Venkitasubramaniam et al. in [28]: anonymity is quantified by the conditional entropy of the routes, and specific network design strategies are proposed to balance throughput and route anonymity, which is achieved by combining packet relay and injection of dummy traffic. Our proposed pseudonymization infrastructure also relies on packet forwarding, but it does not adopt dummy traffic injection.

A pseudonym-based infrastructure based on one-way hash functions is adopted by Henrici et al. [29] in the context of RFID systems. The main idea is to use pseudonyms that change regularly and are linked to the owner of a tag, without affecting location privacy. The pseudonyms are computed collecting inputs from each node on the path to the receiver. The main disadvantage of the infrastructure is that it is static and thus cannot ensure long term security.

Burkhart et al. [30] propose SEPIA (SEcurity through Private Information Aggregation), a library that allows efficient aggregation of multi-domain network data and preserves privacy using MPC (MultiParty Computation). Their proposal includes efficient MPC comparison operations and MPC protocols for events correlation and distinct counts computation. Moreover, they implement the protocols in SEPIA library, evaluating the performance on realistic settings.

Ozdemir et al. [31] propose a data aggregation and authentication protocol for wireless sensor networks, which supports false data injection by a fraction of compromised nodes by verifying integrity directly on the encrypted data. However, these two papers present a solution for aggregating data but they do not consider pseudonymization, which is the aim of the security infrastructure proposed in this paper.

The previously proposed methods are not suited for an anonymized Smart Grid deployment, because they do not satisfy the condition of low computational cost at the meter. Our approach solves this problem by using a Shamir Secret Sharing scheme. Moreover, we jointly address both the problem of frequent re-pseudonymization and of IDs recovery, by leveraging on a single pseudonymization protocol. The former is addressed by a multiple tier pseudonymization network, while the latter exploits a novel key escrow procedure.

3. BACKGROUND

This section provides a short overview of the cryptographic schemes used in the pseudonymization protocol.

3.1. Shamir Secret Sharing Scheme

Shamir Secret Sharing (SSS) [32] is a threshold scheme proposed to divide a secret in \( w \) parts called shares. The shares are distributed among the participants to the protocol: in order to recover the secret, at least \( t \leq w \) participants must cooperate.

The scheme works as follows: let \( \mu \in \mathbb{Z}_q \) be the secret, where \( q \) is a prime number, greater than all the possible secrets. To split the secret in \( w \) shares, generate \( t - 1 \) integer random numbers \( \rho_1, \rho_2, \ldots, \rho_{t-1} \) uniformly distributed in \([0, q - 1]\) and compute the \( s \)-th share \((x_s, y_s)\), for \( 1 \leq s \leq w \), where \( x_s \) are distinct integer numbers and \( y_s = \mu + \rho_1 x_s + \rho_2 x_s^2 + \ldots + \rho_{t-1} x_s^{t-1} \mod q \). The secret can be reconstructed if \( t \) or more shares are available by using the Lagrange interpolation method.

Moreover, SSS is known to be a perfect secret sharing scheme [33]. Therefore, for any subset \( S \) of shares of cardinality at most \( t - 1 \), it holds that:

\[
\Pr(M = \mu | S) = \Pr(M = \mu)
\]

for every \( \mu \in \mathbb{Z}_q \), where \( M \) is the random variable indicating the secret chosen by the dealer.
3.2. Chaum Mixing

Chaum [34] presents a technique based on public key cryptography that permits one correspondent to remain anonymous to a second, while allowing the second to respond via an untraceable return address. This technique relies on a mixer that processes each message before it is delivered. The use of mixing guarantees anonymity by hiding the correspondence between the sender and the receiver, which is achieved by wrapping the messages with a public-key cryptography. The Chaum Mixing scheme is deployable, usable and has a simple design and that is why is widespread in numerous scenarios. [13–16].

The algorithm works as follows: a participant prepares a message $M$ for delivery to a receiver at address $A$ by sealing it with the addressee’s public key $K_A$, appending the address $A$, and then sealing the results with the mixer’s public key $K_1$. The mixer receives the encrypted message $K_1(R_1, K_a(RO), M), R_0$ and $R_1$ are random strings. Then, it decrypts the input with its private key, removes $R_1$, and outputs $K_a(RO, M), A$. Finally, the addressee decrypts the message with its private key, removes $R_0$ and obtains the original message $M$.

3.3. Identity Based Proxy Re-Encryption

Green and Ateniese [35] propose an Identity Based Proxy Re-Encryption (IB-PRE) protocol based on the assumed intractability of the Decisional Bilinear Diffie-Hellman problem (DBDHH) [35, Definition 3.2] in $G_1, G_T$. The IB-PRE scheme allows a proxy to convert an encryption under a user’s identity into an encryption computed under another user’s identity. Moreover, the proxy does not learn the secret keys of the users nor the plaintext. Furthermore, this scheme guarantees unidirectionality, meaning that a user A can delegate to another user B, without permitting A to decrypt B’s ciphertexts, and non-interactivity, meaning that a user A can construct a re-encryption key without the participation of B. The IB-PRE has been adopted to secure communications and provide anonymity in many different frameworks [17–23].

It comprises the following set of algorithms:

- **Setup($l$)** accepts a security parameter, $l$, and outputs both the master public parameters, params, which are distributed to users, and the master secret key, $msk$, which is kept private. Let $e: G_1 \times G_2 \rightarrow G_T$ be a bilinear map, where $G_1 = \langle g \rangle$ and $G_T$ have order $q$. Let $H_1, H_2$ be independent full-domain hash functions $H_1: \{0,1\}^* \rightarrow G_1$ and $H_2: G_T \rightarrow G_1$. To generate the scheme parameters, selects $s \leftarrow Z_q^{*}$, and outputs $params = (G_1, H_1, H_2, g, g^s), msk = s$.

- **KeyGen(params, msk, id)** on input an identity, $id$, and the master secret key, outputs a decryption key, $sk_{id}$ corresponding to that identity. To extract a decryption key for identity $id \in \{0,1\}^*$, returns $sk_{id} = H_1(id)^s$.

- **Encrypt(params, id, m)** on input a set of public parameters, an identity $id$ and a plaintext, $m \in M$, where $M$ is the messages space, outputs $c_{id}$, the encryption of $m$ under the specified identity. To encrypt $m$, select $r \leftarrow Z_q^{*}$ and output $c_{id} = (g^r, m \cdot e(g^s, H_1(id))^r) = (C_1, C_2)$.

- **RKGen(params, skid, id, id, id)** on input a secret key $sk_{id}$ and identities $id_1, id_2$, outputs a re-encryption key, $rk_{id_1 \rightarrow id_2}$. It selects $X \leftarrow G_T$ and compute $(R_1, R_2) = Encrypt(params, id_2, X)$ and returns $rk_{id_1 \rightarrow id_2} = (R_1, R_2, sk_{id_1}^{-1}, H_2(X)) = (R_1, R_2, R_3)$.

- **Reencrypt(xs, rk_{id_1 \rightarrow id_2}, c_{id_1})** on input a ciphertext $c_{id_1}$ under identity $id_1$, and a re-encryption key, $rk_{id_1 \rightarrow id_2}$, outputs a re-encrypted ciphertext $c_{id_2}$.

- **Decrypt(params, skid, c_{id})** decrypts the ciphertext, $c_{id}$ using the secret key $sk_{id}$, and outputs $m$ or an error.

3.4. RSA Cryptosystem with Optical Asymmetric Encryption Padding

The RSA Cryptosystem with Optional Asymmetric Encryption Padding (OAEP) [33, Cryptosystem 5.4] is defined as follows. Let $K_e = (b, e)$ and $k_d = (b, d)$ be the RSA public/private key pair with modulus $b$, which is $l$ bits long, and encryption and decryption exponents, respectively, $e$ and $d$. Let $o$ be a positive integer with $o < l < 2o$. The deterministic one-way functions

\[
H_1: \{0,1\}^{l-o-1} \rightarrow \{0,1\}^o \\
H_2: \{0,1\}^o \rightarrow \{0,1\}^{l-o-1}
\]

are systemwide masking generation functions (MGF), which can be implemented using the construction in PKCS#1 [36, Appendix B2].
The encryption function is defined as:

\[ E_{k_e} : \{0, 1\}^m \times \{0, 1\}^{l-o-1} \rightarrow \{0, 1\}^l \]

The ciphertext \( y = E_{k_e}(x, r) \) is calculated as follows:

\[
\begin{align*}
x_1 &= x \oplus H_1(r) \\
x_2 &= r \oplus H_2(x_1) \\
E_{k_e}(x, r) &= (x_1 \parallel x_2)^c \mod b
\end{align*}
\]

The decryption function performs the following calculations:

\[
\begin{align*}
x_1 &= L_{m+1}(y^d \mod b) \\
x_2 &= R_{l-m-1}(y^d \mod b) \\
x &= H_1(x_2 \oplus H_2(x_1)) \oplus x_1
\end{align*}
\]

where \( L_n(x) \) and \( R_n(x) \) denote the \( n \) leftmost bits of \( x \) and the \( n \) rightmost bits of \( x \) respectively.

Note that we use some different bit lengths with respect to the reference in order to guarantee that the modulus operation does not exceed the length limit. Moreover, note that we consider the RSA cryptosystem with OAEP secure against CCA, as stated in [37].

### 3.5. Security against Chosen-Ciphertext Attacks (CCA)

In a chosen-ciphertext attack, the adversary has the ability not only to encrypt messages of her choice, but also to request decryption of arbitrary ciphertexts. In fact, the adversary can access a decryption oracle \( \text{Dec}_{sk}(\cdot) \) in addition to the encryption oracle \( \text{Enc}_{pk}(\cdot) \). The only restriction to the oracle access is that the adversary is not allowed to request the decryption of the challenge ciphertext. A cryptosystem is assumed to be secure under a chosen-ciphertext attack if the adversary is not able to distinguish between the encryption of two arbitrary messages. The detailed description of the CCA indistinguishability experiment is given in [38].

Here we report the description of the experiment \( \text{PubK}_{\text{CCA}}(\cdot, \cdot)(n) \) for a public key encryption scheme \( \Pi \) and an adversary \( B \):

1. Gen \( (1^n) \) is run to obtain the keys \( (pk, sk) \).

2. The adversary \( B \) is given \( pk \) and access to a decryption oracle \( \text{Dec}_{sk}(\cdot) \). It outputs a pair of messages \( x_0, x_1 \) of the same length.

3. A random bit \( b \leftarrow \{0, 1\} \) is chosen, and then a ciphertext \( c \leftarrow \text{Enc}_{pk}(x_b) \) is computed and given to \( B \).

4. \( B \) continues to interact with the decryption oracle, but may not request a decryption of \( c \) itself. Finally, \( B \) outputs a bit \( b' \).

5. The output of the experiment is defined to be 1 if \( b' = b \), and 0 otherwise.

It holds that:

\[
\Pr[\text{PubK}_{\text{CCA}}(\Pi, \cdot)(n) = 1] \leq \frac{1}{2} + \text{negl}(n)
\]

where \( \text{negl}(n) = \frac{1}{p(n)} \), for an arbitrary polynomial \( p \), and a large integer \( n \).

### 4. AN ARCHITECTURE FOR METERING DATA PSEUDONYMIZATION

#### 4.1. The Pseudonymization Architecture

As depicted in Fig. 1, three different sets of nodes are comprised in our proposed architecture:

- The set of smart Meters, \( M \), which generate the energy consumption data.
- The set of Privacy Preserving Nodes (PPN), \( N \), which are the nodes that perform data pseudonymization.
- The set of information External Entities, \( E \), which receive pseudonymized data and represent the utilities or other third party services.
The architecture also includes a Configurator node, which checks whether the monitoring requests received from the External Entities are compliant to the grid privacy policies, periodically updates the public/private key pairs and recovers the Meter’s identities from their pseudonyms in case of emergencies or faults.

In the remainder of the paper, we assume that the communication network is reliable and timely, i.e., no message can be lost due to communication delays or node malfunctioning. For an extensive fault-tolerance analysis of our privacy-preserving infrastructure, the reader is referred to [5]. Moreover, it is worth noting that our protocol is agnostic to the type of data to be anonymized. Therefore, our pseudonymization protocol can easily be integrated with data perturbation and obfuscation techniques: for example, as proposed in [39], a battery can be installed at the customer’s premises in order to partially hide the energy consumption profile, thus reducing the accuracy of linking attacks based on behavioral patterns.

4.2. Problem Statement

We assume that time is divided in intervals of given duration $\tau$ (in the order of seconds or minutes) and that all the nodes can be loosely synchronized to a common time reference. Each Meter, PPN and External Entity is characterized by a unique identifier. At each time interval, $i$, the $m$-th Meter generates a measurement $x_m^i$, which is expressed as an integer number modulo $q$. During a setup phase, the $e$-th External Entity specifies the set of Meters $\Pi_e$ he/she wants to monitor. At every time interval, for each of the monitored Meters, the External Entity expects to learn a set $\Omega_e^i$ of cardinality $|\Pi_e|$ of pseudonymized measurements:

$$\Omega_e^i = \{(x_m^i, PD_e^m) : m \in \Pi_e\}$$

(1)

where $PD_e^m$ is the pseudonym of the $m$-th Meter towards the $e$-th External Entity.

4.3. Scheme Description

Our data pseudonymization protocol consists of the following tuple of probabilistic polynomial-time (p.p.t.) and polynomial time (p.t.) algorithms:

- $(k_d, params) \leftarrow$ Setup($1^l$): takes as input the security parameter $l$, and outputs the public parameters $params$ and the Configurator’s private key $k_d$.
- $(z_m^1(1), \ldots, z_m^n(n), \ldots, z_m^N(N), ID_m, r_m^i) \leftarrow$ mSend($param, i, m, x_m^i$): during each round $i$, each Meter $m$ calls the mSend algorithm to encode its data $x_m^i$ and then it sends the message $msg_m^i$, composed by the encrypted data $z_m^i(n)$, its identity $ID_m$, and a nonce $r_m^i$, to the $n$-th PPN.
- $(PD_e^m, z_m^i(n)) \leftarrow$ PNSend($param, i, n, ID_m$, $r_m^i, z_m^i(n)$): at each time interval $i$, each PPN $n$ encodes the Meter’s identity $ID_m$ and sends the message $msg_e^m$, composed by the encrypted data $z_m^i(n)$ and the pseudonym $PD_e^m$, to an External Entity.
- $(PD_e^m, x_m^i) \leftarrow$ eReceive($param, i, e, PD_e^m$, $z_m^i(1), \ldots, z_m^n(n), \ldots, z_m^N(N)$): finally, the External Entity $e$ decodes the encrypted data and obtains the measurement $x_m^i$ with the associated pseudonym $PD_e^m$.

We assume that the Secret Sharing Scheme used in the algorithm mSend is unconditionally secure. Thus, the adversary is allowed to interact with an encryption oracle that encrypts a plaintext message $x$ using the Shamir Secret Sharing scheme with threshold $t$ and returning a ciphertext $z(t-1) \leftarrow Enc(t)$, where $z(t-1)$ is a vector of $t-1$ shares. A cryptosystem is unconditionally secure if the adversary is not able to distinguish the encryption of two arbitrary messages with less than $t$ shares.

Now we describe the experiment $UnSec_{\Pi, B}$ for an encryption scheme $\Pi$ and an adversary $B$:

1. A threshold $t$ is chosen.
2. The adversary $B$ is given access to the encryption oracle $Enc(t)$. It outputs a pair of messages $x_0, x_1$ of the same length.
3. A random bit $b \leftarrow \{0, 1\}$ is chosen, and then a ciphertext $z_b(t-1) \leftarrow Enc(t)$ is computed and given to $B$. We call $z_b(t-1)$ the challenge ciphertext.
4. $B$ continues to interact with the encryption oracle. Finally, $B$ outputs a bit $b'$.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

It holds that:

$$Pr(UnSec_{\Pi, B} = 1) = \frac{1}{2}$$
4.4. Security Properties

In this Section we enlist the security properties that capture the privacy requirements of the Smart Grid infrastructure.

4.4.1. Full Pseudonymization

Consider the following experiment full-p for a given algorithm $\mathcal{A}$ and a parameter $l$: the experiment assumes as adversary a malicious External Entity $e^*$ and focuses on two Meters $ID_1, ID_2 \in \Pi_{e^*}$.

1. The Setup($l^1$) algorithm outputs the system parameters.
2. The first Meter executes $mSend(param, i, 1, x_1^1)$ and outputs the messages $msg_1^1, ..., msg_n^1, ..., msg_2^1$.
3. The second Meter executes $mSend(param, i, 2, x_2^1)$ and outputs the messages $msg_2^1, ..., msg_2^2, ..., msg_2^1$.
4. Each PPN $n$ receives the two messages $msg_n^1, msg_n^2$ and calls the $PPNSend(param, i, n, ID_m, \tau_i^m, z_i^m(n))$ algorithm. Then each PPN sends two messages $pmsg_n^m$ (with $m \in \{1, 2\}$) to the External Entities.
5. Finally each External Entity runs $eReceive(param, i, e, PD_e^m, z_i^m(1), ..., z_i^m(n), ..., z_i^m(N)$ (with $m \in \{1, 2\}$) and obtains the measurement with the associated pseudonym.
6. The malicious External Entity $e^*$ executes $\mathcal{A}$ and outputs $m' \in \{1, 2\}$.
7. The output of the experiment is 1 if $m' = m$, and 0 otherwise.

Definition 1

A pseudonymization protocol provides full pseudonymization relatively to full-p if for all p.p.t. algorithms $\mathcal{A}$ there exist a negligible function $negl$ such that:

$$Pr(\text{full-p} = 1) \leq \frac{1}{2} + negl(l)$$

4.4.2. Perfect Forward Anonymity

Consider the following modification to the full-p experiment for a given algorithm $\mathcal{A}$ and a parameter $l$, which we name full-p-pfa experiment. This assumes the presence of a malicious PPN $n^*$ and a malicious External Entity $e^*$ and focuses on two Meters $ID_1, ID_2 \in \Pi_{e^*}$.

The full-p experiment is repeated until step 5 for some rounds $1, 2, ..., i$, thus, each round, the algorithms executed are $Setup(1^i), mSend(param, i, m, x_i^m), PPNSend(param, i, n, ID_m, \tau_i^m, z_i^m(n))$, and $eReceive(param, i, e, z_i^m(1), ..., z_i^m(n), ..., z_i^m(N)$, all of them with $m \in \{1, 2\}$. Moreover, after the execution of step 5 and before step 6, during the round $i^* > i + \alpha \tau$, a collusion of a malicious External Entity $e^*$ and a PPN $n^*$ occurs. Such pair of malicious nodes can obtain the correspondence between the measurement $x_i^m$, the pseudonym $PD_e^m$, and the identity $ID_m$ associated to a Meter $m \in \{1, 2\}$. This happens because the malicious PPN $n^*$ knows the correspondence between $ID_m$ and $PD_e^m$, while the malicious External Entity $e^*$ knows the correspondence between $PD_e^m$ and $x_i^m$. Then, the collusion executes the algorithm $\mathcal{A}$ and outputs $m' \in \{1, 2\}$. The output of the experiment is 1 if $m' = m$, and 0 otherwise.

Definition 2

A pseudonymization protocol provides full pseudonymization with perfect forward anonymity relative to full-p-pfa if for all p.p.t. algorithms $\mathcal{A}$ there exist a negligible function $negl$ such that:

$$Pr(\text{full-p-pfa} = 1) \leq \frac{1}{2} + negl(l)$$

4.4.3. Unconditionally Indistinguishable Encryption

We define the following experiment blind for an adversary which controls a collusion of $t^* < t$ PPNs.

1. The Setup($l^1$) algorithm outputs the system parameters.
2. At round $i$, the adversary chooses two secrets $\tau_i^b$ and $\tau_i^b$ and gives them to the Meters.
3. A random bit $b \in \{0, 1\}$ is chosen and kept secret to the adversary.
4. The first Meter executes $mSend(param, i, 1, \tau_i^b)$ and outputs $t$ messages $msg_n^b$ with the encrypted data $z_i^b(n)$, each of them being the share destined to the $n$-th PPN ($1 \leq n \leq t$).
5. The second Meter executes $mSend(param, i, 2, \tau_i^{b-1})$ and outputs $t$ messages $msg_n^b$ with the encrypted data $z_i^b(n)$, each of them being the share destined to the $n$-th PPN.
6. Each PPN \( n \) receives the two messages \( msg_1^n \) and \( msg_2^n \). The adversary outputs \( b' \).

7. The output of the experiment is 1 if \( b' = b \), and 0 otherwise.

**Definition 3**

A protocol provides **unconditionally indistinguishable encryption** under \( \text{blind} \) if it holds that:

\[
Pr(\text{blind} = 1) = \frac{1}{2}
\]

In Section 7.2, we provide the description of other properties related to our pseudonymization protocol.

5. **THE PSEUDONYMIZATION FUNCTION**

Let \( E_{k_e}(\mu, r) \) be a keyed trapdoor one-way function. The function takes as input a plaintext \( \mu \) and a security nonce \( r \). The output of the function is the ciphertext \( y \).

We assume that the Configurator generates the public/private key pair, keeps the private key \( k_d \) and distributes the public key \( k_e \) to all the PPNs. The cryptosystem allows the PPN \( n \in N \) to compute the pseudonym \( PD^n_e \) which will be associated to the data generated by Meter \( m \in M \) and destined to External Entity \( e \in E \). The PPN calculates:

\[
PD^n_e = E_{k_e}(ID_m||e\|i/\alpha\|\alpha, w^n_m) \quad (2)
\]

The ciphering function \( E_{k_e} \) takes as input a concatenation of the Meter’s identity, \( ID_m \), the External Entity identification number \( e \), the round identifier \( i \), and a security nonce \( w^n_m \). As it will be detailed in Section 6, the frequent refreshment of \( w^n_m \) guarantees a prevention against linking attacks, as described in [12].

Note that such cryptosystem allows the Configurator to recover the Meter identity by decrypting \( PD^n_e \) with its private decryption key \( k_d \). In this paper we consider RSA-OAEP as a randomized trapdoor function because it is invertible, secure against CCA and it is one of the most widely used due to the easiness of implementation [40–43].

6. **COMMUNICATION PROTOCOL**

In this Section we describe the messages which constitute our proposed protocol, which exploits the homomorphic properties of SSS scheme to provide network blindness (Property 3 in Section 4.2). Then, we discuss other two possible ways to provide the same property using, respectively, Chaum Mixing and IB-PRE. In Section 8, we compare their performance, concluding that the Shamir-based one is more scalable. We stress that, while the mixing-based protocol is a straightforward implementation of [34], the IB-PRE-based one is an original elaboration over the ideas in [35]. In the initial version, however, the secret key and the re-encryption key were assumed to be held by the same entity. This is not the case with our protocol, therefore we need to prove that a node knowing the re-encryption key cannot recover the secret key. Such proof is provided in the Appendix.

All the protocols assume that a confidential, authenticated communication is established between the node pairs.

The data pseudonymization protocol consists of four phases:

1. **Setup**: the initial phase is performed only once to define the set of public parameters and to distribute them to the users. Moreover, in this phase each External Entity specifies the set of monitored Meters, the Configurator checks the admissibility of the External Entities’ requests and communicates to each Meter the set of External Entities interested in monitoring its data.

2. **Key Refresh**: this procedure is performed from time to time to update the key pairs and to communicate the new public keys to Meters, PPNs and External Entities.

3. **Data Collection**: this phase is performed at every interval to collect the pseudonymized data and involves Meters, External Entities, and PPNs.

4. **Identity Recovery**: this procedure is performed only in presence of alarms/faults to recover the identity of the faulty Meters and involves a External Entity and the Configurator.

We first describe the messages sent during the Setup and the Identity Recovery, then we discuss the Key Refresh and Data Collection phases comparing the usage of SSS
A protocol for Metering Data Pseudonymization in Smart Grids

D. C. Rottondi, G. Mauri, and G. Verticale

scheme to two alternative approaches relying on Chaum-mixing and IB-PRE.

During the initial Setup phase, the following messages are exchanged:

1.1 SPECIFYMONITOREDSET

e \rightarrow f : \Pi_e

The e-th External Entity specifies to the Configurator the set of Meters, \( \Pi_e \), that the External Entity wants to monitor. The Configurator checks the conformance of the External Entity’s request to the system policy.

1.2 SPECIFYMONITORINGSET

f \rightarrow m : \Psi_m

The Configurator computes the set \( \Psi_m \) of External Entities which are monitoring Meter \( m \) and communicates it to the Meter.

In case of faults or alarms, an External Entity is allowed to obtain the identity of a Meter (i.e., Identity Recovery) through the following steps:

4.1 RECOVERYREQUEST

e \rightarrow f : PD_e^m

The e-th External Entity communicates to the Configurator the pseudonym of the Meter whose identity he is interested in. The Configurator deciphers \( PD_e^m \) using his private key \( k_d \), removes \( e || \lceil \frac{1}{\alpha} \rceil \alpha \) and obtains \( ID_m \).

4.2 SENDIDENTITY

f \rightarrow e : PD_e^m || ID_m

The Configurator communicates the Meter’s identity and the associated pseudonym to the External Entity.

6.1. Shamir Secret Sharing Scheme

The SSS scheme works as follows: the measurements generated by every Meter are divided in \( t \) shares, where \( t \) is a system parameter, and can be recovered if and only if all the shares are available at the External Entity (i.e., we assume \( t = w \)). We suppose that the number of installed PPNs is also equal to \( t \). The Meters send each share to a different PPN, therefore individual measurements can be obtained only through a collusion of all the involved PPNs. Once the \( n \)-th PPN receives a share from Meter \( m \) destined to the External Entity \( e \), it computes the Meter’s pseudonym, whose value depends both on \( m \) and \( e \). Then, it forwards the share to the External Entity, together with the computed pseudonym (see Fig. 2). Therefore, the External Entity can recover the individual data by combining the shares associated to the same pseudonym, but obtains no information about identity of the Meters which generated them.

With reference to Fig. 3, the Key Refresh procedure includes only one message:

2.1 REFRESHEYE

f \rightarrow n : k_e

The Configurator communicates to the PPNs its public key \( k_e \) every time the key pair \((k_e, k_d)\) is refreshed. The key \( k_d \) is kept private.

During the Data Collection phase the following messages are exchanged:

3.1 SENDDATA

m \rightarrow n : s(x_i^m, n) || ID_m || r_i^m

At the \( i \)-th time interval, the Meter \( m \) produces the measurement \( x_i^m \) (the secret) and sends to the \( n \)-th PPN the corresponding share \( s(x_i^m, n) \) computed according to the SSS scheme, its identity \( ID_m \) and a random number \( r_i^m \).

3.2 SENDPSEUDONYMIZEDDATA

n \rightarrow e : s(x_i^m, n) || PD_e^m
The $n$-th PPN computes the pseudonym $PD_e^{m}$ according to (2). The pseudonym will be associated to the data generated by $m$-th Meter and destined to the $e$-th External Entity. To do so, the PPN uses the Configurator’s public key $k_e$. Note that the security nonce $w_{em}^{m}$ is updated with the current value of the hash-function $H(x_{i}^{m} || e)$, (which can be implemented using the construction in PKCS#1 [36, Appendix B2]), at all the $i$-th intervals such that $i$ is an integer multiple of $\alpha$, where $\alpha$ is a design parameter. Therefore, once $w_{em}^{m}$ is refreshed, it remains unchanged for a time window of duration $T = \alpha T$, which represents the validity time span of the pseudonym.

Once the pseudonym is computed, the PPN sends it to the External Entity, together with the share. The External Entity waits until reception of all the $t$ pseudonymized shares for each of the $|\Pi_e|$ pseudonyms and groups together the shares associated to the same pseudonym. Then, for each pseudonym it recovers the corresponding secret $x_{i}^{m}$.

### 6.2. Mixing Approach

An alternative pseudonymization scheme relies on Chaum Mixing: during the Data Collection phase, every

Meter generates the measurement $x_{i}^{m}$ and computes the pseudonym $PD_e^{m}$. Then it creates the mixing packet $MIX_e^{m} = E_{k_e}[x_{i}^{m} || PD_e^{m}]$, which includes both measurement and pseudonym, and sends it to a randomly chosen PPN through the SENDDATA message. The PPN forwards the packet (FORWARDDATA message) to the External Entity to whom the message is destined, which recovers the individual data by decrypting the packet. The Key Refresh phase is executed to update and refresh the key pairs $(k_e, k_{e'})$ for mixing and $(k_e, k_d)$ for computing the pseudonyms.

Figure 4 shows the protocol messages of the Key Refresh and Data Collection phases.
6.3. Identity-Based Proxy Re-Encryption

A second variant of the pseudonymization protocol relies on the IB-PRE scheme. In this case, the Key Refresh phase comprises also the keyGen algorithm that is executed by Configurator to generate the PPNs and external entities’ secret keys, $sk_m$ and $sk_e$. The latter is sent with SendSecretKey message. The Configurator also generates the re-encryption keys $rk_{m,e}$ thanks to RKGen algorithm and sends them in the SendRekeying message to each PPN. The keys are generated by the Configurator because it is the only node that possesses the master secret key $msk$.

The Data Collection phase comprises the Encrypt algorithm performed by the Meters to encrypt the measurements destined to the PPNs, the Reencrypt algorithm, and the computation of the pseudonyms performed by the PPNs. The messages SendEncryptedData and SendReEncryptedData are used to convey the encrypted data to the External Entities and are composed by the concatenation of the encrypted measurement $y_{me}$, the Meter identity $ID_m$, and a random number $r_m^e$, and the concatenation of the re-encrypted message $y_e$ and the pseudonym $PD_e^m$ respectively. Finally, the Decrypt algorithm is used by the External Entities to decrypt the ciphertexts. Figure 5 depicts the protocol messages in each phase.

In order to provide network blindness, the PPN cannot recover the secret key from the re-encryption key. A proof is given in the Appendix.

7. SECURITY EVALUATION

7.1. Security Proofs

This section discusses how the properties presented in Section 4.2 are satisfied by our proposed pseudonymization cryptosystem. We do not discuss further the attack scenario of a passive intruder trying to collect multiple messages from a given Meter to recover the individual measurements: the assumption of a computationally secure confidential and authenticated channel between the nodes prevents this kind of attack. Moreover, we assume that the adversary $A$ has no auxiliary information about the correspondence between the measurement $x_i$ and the identity $ID_m$ and thus cannot distinguish between two different measurements generated by different Meters.

**Theorem 1**

If the RSA with OAEP encryption scheme is CCA secure, then our pseudonymization protocol provides full pseudonymization with respect to full-p.

**Proof**

By contradiction, let $A$ be a p.p.t. algorithm that has more than a negligible advantage in the full-p experiment. Given the pseudonym $PD_m^0$ and $(ID_m\|\epsilon[i/\alpha]\|\alpha)$, algorithm $A$ yields $1$ with non-negligible probability.

We now define the algorithm $B$ that runs the CCA indistinguishability experiment where a challenge ciphertext $z = PD_e^m$ is given to $B$. Moreover, $B$ chooses two plaintexts $(ID_m\|\epsilon[i/\alpha]\|\alpha)$ with $m \in \{1, 2\}$.

At point 4 of the CCA experiment, $B$ interacts with $A$, obtaining $1$ if $PD_e^m = E_{sk_m}[ID_m\|\epsilon[i/\alpha]\|\alpha, r_m^e]$ with $m \in \{1, 2\}$, where $E_{sk_m}$ is defined in Section 3.4. The output of $A$ is used as output of $B$, solving the CCA experiment with non-negligible probability.

If $B$ outputs $1$, it means that $B$ has solved the CCA indistinguishability experiment with non-negligible probability, i.e. $Pr[PubK_{CC}\epsilon[i/\alpha]\|\alpha(n) = 1] \geq \frac{1}{2} + negl(n)$. 

**Theorem 2**

If RSA with OAEP is CCA-secure, then our protocol provides full pseudonymization with perfect forward anonymity relative to full-p-pfa.

**Proof**

By contradiction, let $A$ be a p.p.t. algorithm that has more than a negligible advantage in the full-p-pfa experiment. Given the pseudonym $PD_e^m$ and $(ID_m\|\epsilon[i/\alpha]\|\alpha)$,
algorithm $\mathcal{A}$ yields the correct answer with probability greater than $1/2$. Moreover, $\mathcal{A}$ has an oracle access to a decryption function that gives the correspondence between $x_i^m, PD^m_i, ID_m$, relative to a time interval $i^\ast$. This means that $\mathcal{A}$ can say with certainty if $PD^m_i = E_{b_k}[ID_m, e][i^\ast/\alpha, r_i^m]$ is a valid relation. The output of $\mathcal{A}$ is used as output of $\mathcal{B}$, defined in the previous proof, solving the CCA indistinguishability experiment with non-negligible probability, leading to the same proof of Theorem 1.

**Theorem 3**

If the Shamir Secret Sharing scheme is a perfect secret sharing scheme, then our protocol provides unconditionally indistinguishable encryption.

**Proof**

Since the blind experiment assumes a collusion of $t^\ast < t$ PPNs, the colluded PPNs obtain two sets $S_1, S_2$, each of cardinality at most $t - 1$, of shares of the two secrets $\pi_i^0$ and $\pi_i^{1-b}$ respectively. Therefore:

$$\Pr\{b = 0|S_1, S_2\} = \Pr\{M_1 = \pi_i^0, M_2 = \pi_i^{1-b}|S_1, S_2\} = \Pr\{M_1 = \pi_i^{1-b}|S_1, S_2\}$$

where $M_1, M_2$ are the random variables indicating the secrets encrypted by Meter 1 and by Meter 2 respectively. Since the value of $M_2$ is completely determined by knowledge of $M_1$, then $M_2$ can be deleted from the last term of (3).

Since the random polynomials used to generate $S_1$ and $S_2$ are independent, the knowledge of $S_2$ gives no information about $M_1$. Further, exploiting the perfect secrecy property of SSS, we can write:

$$\Pr\{M_1 = \pi_i^0|S_1\} = \Pr\{M_1 = \pi_i^0\} = \Pr\{b = 0\} = 1/2$$

Similar considerations hold for $b = 1$. Therefore, the knowledge of $S_1, S_2$ gives no information about the value of $b$ and no algorithm can guess $b$ with probability greater than $1/2$. 

**7.2. Other security properties**

1. There exist a polynomial time algorithm that, given the private key, can recover the identity of Meter $m$ from pseudonym $PD^m_i$.

   This property is a direct consequence of the Configurator having the private key, which makes it able to recover $ID_m$ from $PD^m_i$.

2. Before sending its data, the Meter is aware of the set of External Entities $\Psi_m = \{e : m \in \Pi_e\}$ monitoring its data thanks to the message $\text{SPECIFYMONITORINGSET}$.

3. Given a pair of distinct Meters’ identities $(m, m')$ and the same External Entity $e$, or a pair of distinct External Entities $(e, e')$ and the same Meter $m$, the output of the function $E_{b_k}$ is always different. In other words, the output of the pseudonymization function is never the same for different sets of Meters or External Entities, using the same value of $e$ or $m$, respectively.

This property is consequence of using the ciphering function $E_{b_k}$ that relies on RSA with OAEP (see Section 3.4), which guarantees that for different inputs, the outputs are never identical.

**8. PERFORMANCE ASSESSMENT**

In this Section we evaluate the computational costs of the protocol presented in Section 6 and the number of exchanged messages as a function of the system parameters $|M|$, $|N|$ and $|E|$. We also consider the case of a user, i.e. a Meter or a External Entity, joining or leaving the system.

First, it is useful to discuss a suitable choice for the system parameters: assuming 128-bit long identifiers for Meters and External Entities, 64-bit long round numbers and 128-bit long nonces, a suitable choice is $\alpha = 512$ and $l = 1024$, which results in 1024-bit pseudonyms. It is worth considering that, if the size of the pseudonym is an issue, the pseudonymization cryptosystem can be easily implemented using Elliptic Curve Cryptography, resulting in shorter pseudonyms.

**8.1. Number and Size of Exchanged Messages**

During the Setup and Identity Recovery phases the number of messages is independent from the choice of the measurement encryption scheme. In the Setup phase, the Configurator receives $|E|$ messages from the External Entities and sends $|M|$ messages to the Meters. For the Identity Recovery phase, the number of exchanged messages is at most $(2\cdot|M|\cdot|E|)$, but assuming a low
probability of faults, it tends to the lower bound, that is 2 messages (i.e., there is only one faulty Meter).

Table I summarizes the number of exchanged messages in the Setup and Identity Recovery phases.

We consider now the exchanged messages during the Key Refresh and Data Collection phases.

During the Key Refresh phase, in case the SSS scheme is used, the Configurator simply forwards $k_c$ to each of the $|N|$ PPNs. Conversely, in case of mixing scheme, each External Entity sends $k_c$ to the Configurator, which in turn forwards the External Entities’ public keys to the $|M|$ Meters according to the monitoring requests. In the IB-PRE, the Configurator sends the messages containing the public keys and the re-encryption keys to the $|N|$ PPNs and sends the secret keys to the $|E|$ External Entities.

For what concerns the Data Collection phase, in the SSS scheme, the $m$-th Meter sends a share to each of the $|N|$ PPNs, which in turn sends the shares with the associated pseudonym to the External Entities that are monitoring the $m$-th Meter. Therefore, the total number of exchanged messages is $|M| \cdot |N| + |M| \cdot |N| \cdot |E|$.

In the mixing scheme, the Meter sends the $|E|$ mixing packets to the PPNs, that simply forward them to the External Entities. This procedure requires $2 \cdot |M| \cdot |E|$ messages.

Differently, in the IB-PRE scheme the Meter encrypts the measurement and sends it to only one PPN, which computes the pseudonym and re-encrypts the packet before forwarding it to the External Entities. In this scheme, the total amount of messages is $|M| + |M| \cdot |E|$.

We now evaluate the size of the messages. Let $L[x]$ be the length in bits of $x$. In the SSS scheme, the size of the SendPseudonymizedData message is $L[s(x^n_m, n)] + L[P D^n_m] = 128 + 1024 = 1152$ bits.

Conversely, in the mixing scheme, the Meter sends the SendData message to the PPN, that is $L[e] + L[M I X^n_m] = 128 + 1024 = 1152$ bits long, while the PPN sends only the 1024 bits long mixing packet $L[M I X^n_m]$ to the External Entity.

Finally, in the IB-PRE scheme, the Meter sends the encrypted data to the PPN together with its identity and a round number, for a total length of $L[y_e] + L[I D_m] + L[r^n_m] = 1248 + 128 + 128 = 1504$ bits, while the PPN sends to the External Entity the re-encrypted message and the pseudonym, for a total message length of $L[y_e] + L[P D^n_m] = 2496 + 1024 = 3520$ bits. Therefore, the size of the single messages sent by each Meter is lower in the SSS scheme than in mixing and IB-PRE schemes, while the size of the single messages sent by each PPN in the SSS scheme is slightly higher than in the mixing scheme, but lower than in the IB-PRE scheme.

Table II compares the number of messages received and sent by each entity and reports the corresponding message sizes.

Fig. 6 depicts the trend of the total volume of input and output messages at the Meters and PPNs, assuming $|M|=200$ and $|N|=5$, for different cardinalities of $E$. Since
the Meters are expected to stay in place for several years, it is important that the architecture is capable of scaling to larger numbers of Meters and of EEs without increasing the communication and computation cost at the Meter. We note that, in the SSS scheme, the output data rate at the Meters is only related to the number of PPNs, \(|N|\), which is expected not to change over time. On the other hand, the number of Meters and EEs impacts onto the communication costs of the PPNs. However, PPNs are few with respect to the Meters and easily upgradable in case of need.

8.2. Complexity and Timing of Cryptographic Operations

In this section we evaluate the computational complexity of the cryptographic operations in terms of asymptotic values and computational time. Since the Setup and Identity Recovery phases are independent from the choice of the measurement encryption scheme, we start with the evaluation of their computational costs.

Every time a user joins or leaves the system, the Setup phase is re-executed and \(\Pi_e\) and \(\Psi_m\) are updated. In particular, if the new users are External Entities, they specify their \(\Pi_e\) to the Configurator, which checks the conformance of each request. Then the Configurator computes \(\Psi_m\) and communicates it to the Meters. The same happens in case of new Meters joining or leaving the system. Note that the costs of this phase are variable and it is out of our scope to evaluate them. The same happens for the cost relative to the definition of the system parameters, that are omitted since it is performed only once.

The Identity Recovery phase involves an External Entity and the Configurator. The latter deciphers the pseudonym with his private key, exploiting the Square and Multiply (S&M) algorithm, which has complexity \(C(RSA_{dec})\), that depends on the number of users to be de-anonymized \(|M|\) requested by the \(|E|\) External Entities.

During the Key Refresh phase, the Configurator chooses his public key \(k_e\) and computes the private key \(k_d\), with complexity \(C(RSA_{gen})\). Conversely, the mixing scheme requires each External Entity to choose his public key \(k_e\) and to compute the corresponding private key \(k_e'\) with complexity \(C(RSA_{gen})\). In the IB-PRE scheme, the Configurator performs the KeyGen and RKGen algorithms to generate the secret keys \(sk_n, sk_e\), which remain unchanged, and the re-encryption key \(rk_{n-e}\), which is frequently changed. The computational costs are dominated by the Weil Pairing operations, which have complexity \(C(Pairing)\), that depends on \(p\), that is a 1024 bits long prime number that corresponds to the field over which the elliptic curve is constructed.

The Data Collection phase is performed at every round \(i\). In the SSS scheme, assuming \(t = w = |N|\), the computation of the \(t\) shares requires the generation of \(|N| - 1\) integer random numbers, \(|N|\left(\left|N\right| - 1\right)\) modular multiplications and \(|N|\left(\left|N\right| - 1\right)\) modular sums. This operation has asymptotic complexity \(C(Share_{en}))\).

The PPNs have to compute the pseudonyms \(PD_r^{m}\) using cryptographically secure hash functions and RSA encryptions. The computational cost is dominated by the RSA encryption, which has complexity \(C(RSA_{enc})\). The External Entity receives all the shares associated to different pseudonyms and, for each pseudonym, recovers the corresponding secret with the Lagrange interpolation method, which has complexity \(C(Share_{interp})\).

Differently, in the mixing scheme the \(m\)-th Meter computes the pseudonyms \(PD_r^{m}\) and creates the mixing packet \(MIX_r^{m}\) using cryptographically secure hash functions and RSA encryptions. The computational cost is dominated by the RSA encryption, which has complexity \(2 \cdot C(RSA_{enc})\). The \(MIX_r^{m}\) message is sent to the PPNs that simply forwards the packet to the External Entity \(e\) whom the message is destined to. This operation has negligible complexity. The External Entity receives all the \(MIX\) packets and recovers the corresponding measurements performing the RSA decryption, which has complexity \(C(RSA_{dec})\).
Finally, in the IB-PRE scheme, for the computation of the encrypted measurements the Meter has to perform the HashToPoint and Weil Pairing algorithms [44]. This operation has asymptotic complexity $\mathcal{C}(Pairing)$. The PPNs compute the pseudonyms $PD_{ui}^{en}$ using cryptographically secure hash functions and RSA encryptions and have to re-encrypt the measurements using the Re-encryption algorithm. The complexity is dominated by the RSA encryptions, which have complexity $\mathcal{C}(RSA_{enc})$, and by the encryption function, that has complexity $\mathcal{C}(Pairing)$. The External Entity receives all the encrypted measurements associated to different pseudonyms and recovers the corresponding secret by using the Decrypt algorithm, with complexity $\mathcal{C}(Pairing)$.

For the sake of completeness, in Table IV we report the computational costs of the RSA, SSS and IB-PRE encryption and decryption procedures.

The computational time required by the implementation of IB-PRE scheme turns out to be much higher than in the mixing and SSS schemes. In fact, the Weil Pairing computation, that has the longer execution time, is repeated more than once per message and by every entity.

The above discussed results show that: (1) in the IB-PRE protocol the number of exchanged messages is lower than in the mixing and SSS schemes, but the encryption time is longer; (2) in the SSS scheme the total number of exchanged messages is bigger than in the other two scenarios, but the execution time of the algorithm is shorter.

Hence, we can state that the SSS scheme provides the best compromise between number of messages and encryption time. In fact, although the total number of messages is high, their encryption is computed more quickly than the pairing of the IB-PRE scheme.

9. CONCLUSIONS

This paper proposes a pseudonymization protocol for smart metering measurements, in which the data gathered by the smart meters can be collected by multiple utilities and third parties without revealing the association between users’ identities and pseudonyms. The pseudonymization procedure is performed at intermediate nodes called Privacy Preserving Nodes. We define the security properties that the protocol must satisfy and compare different implementations of the pseudonymization architecture, which leverage on the Shamir Secret Sharing Scheme, on Chaum Mixing, and on an Identity-Based Proxy Re-Encryption scheme, respectively. Results show that the Shamir-based protocol requires a processing effort which is suitable for real-time operations, even if it requires more bandwidth than the others.
A. SECURITY OF THE IB-PRE SCHEME

We prove that the PPN cannot recover the secret key in the IB-PRE scheme with security parameter $l$.

**Theorem 4**

If the DBDH problem is intractable, then there not exists a p.p.t. algorithm $A$ that, given the re-encryption key $rk_{id_1} \rightarrow id_2$, can obtain the secret key $sk_{id}$.

**Proof**

By contradiction, let $A$ be a p.p.t. algorithm that has non-negligible probability $p(l)$ to obtain the secret key, given the re-encryption key. We use $A$ to construct a second algorithm $B$, which has non-negligible advantage in solving the DBDH problem. Algorithm $B$ accepts as input a tuple $\langle G_1 = \langle g, g^a, g^b, g^c, T \rangle \rangle$ and outputs 1 if $T = e(g, g)^{abc}$.

Having the re-encryption key $rk_{id_1} \rightarrow id_2$ from algorithm $A$, we know $(R_1, R_2, R_3) = \langle g^a, x \cdot e(g^a, \mathcal{H}(id_2))^{r_2}, sk_{id_1}^{-1} \cdot \mathcal{H}(X) \rangle$. Moreover, from $A$, we obtain the correct $sk_{id} = \mathcal{H}(id)^{s}$ with non-negligible probability $p(l)$. Now we assume as input for $B$ the tuple $\langle G_1 = \langle g, g^a, g^b, g^c, g^d, T \rangle \rangle$, and as output 1 if $sk_{id}^{-1} = \mathcal{H}(R_2/R_3)$. If $sk_{id}$ obtained from $A$ is correct then $B$ gives the correct answer with probability $\Pr$. This happens with probability $p(l)$. If $sk_{id}$ obtained from $A$ is not correct, $B$ gives a random answer, which is correct with probability $1/2$. The overall probability that $B$ gives the correct answer is $1/2 + p(l)/2$, which is larger than $1/2$ by a non negligible term, violating the assumption of intractability of the DBDH problem.

Thus, we have proved that recovering the secret key from the re-encryption key is an intractable problem.

**REFERENCES**


C. Rottondi, G. Mauri, and G. Verticale A protocol for Metering Data Pseudonymization in Smart Grids


