The ISC Int'l Journal of Information Security
ISeCure

July 2022, Volume 14, Number 2 (pp. 157–165)
**http://www.isecure-journal.org**

# Revisiting the Security and Efficiency of SP2DAS, 3PDA, and EPPA Smart Grid Security Protocols [**]

Hamid Amiryousefi [1], and Zahra Ahmadian [1,*]

[1] *Department of Electrical Engineering, Shahid Beheshti University, Tehran, Iran.*

**ARTICLE INFO.**

*Article history:*
**Received:** February 25, 2021
**Revised:** July 16, 2021
**Accepted:** January 8, 2022
**Published Online:** January 10, 2022

*Keywords:*
Smart Grid, Privacy, Data Aggregation, Digital Signature, Forgery Attack

**Type:** Research Article

**doi:** 10.22042/ISECURE.2022.275064.645

**dor:** 20.1001.1.20082045.2022.14.2.4.6

**ABSTRACT**

This paper analyses the security and efficiency of some notable privacy preserving data aggregation schemes, SP$^2$DAS, 3PDA, and EPPA. For SP$^2$DAS and 3PDA schemes, We show that despite the designers' claims, there are efficient forgery attacks on the signature scheme used. We present a selective forgery attack on the signature scheme of SP$^2$DAS in the key-only attack model and a selective forgery attack on the 3PDA's signature scheme in the known-message attack model, requiring only two pairs of message-signature. These attacks enable the attacker to inject any arbitrary faulty data into the data aggregated by the network, without being detected, which is a serious threat to the performance of the whole network. We also present an improved version of the broadcast encryption scheme used in EPPA scheme, in which the decryption key is half, the decryption complexity is half, and the ciphertext size is 3/4 of the original one. The semantic security of the proposed scheme is proved under the same assumption as the original scheme.

## 1 Introduction

Advancements in technology nowadays has exposed the traditional power grids to a revolution point. Thanks to the recent progresses in communications, networking, and electrical technologies current power grids can evolve to the next generation called smart grid. These new grids enjoy many new advantages such as self healing, monitoring, load balancing, two-way communications etc. Despite these benefits, there are many challenges in the way of developing smart grid that must be addressed with. One of the key challenges in developing smart grids is security. Generally speaking, security includes confidentiality and authenticity of the user's data along with preserving its privacy. This important practical subject has attracted much research through recent years and many security protocols have been proposed by researchers aiming to achieve all these security goals, yet being efficient enough [1–5].

In the model issued by National Institute of Standards and Technology (NIST), the smart meters should record and send the instantaneous values of energy consumption data, in certain time periods usually 15 minutes, to the network for monitoring and analyzing data and a more accurate management of the power grid. The entity in the smart grid responsible for aggregating the power consumption data is called aggregator. However, such a high resolution data may leak lots of sensitive information such as the user's home activities, interests, habits and even his/her lifestyle. So, a privacy-preserving data aggregation

[*] Corresponding author.

[**]This article is an extended/revised version of an ISCISC'17 paper.

Email addresses: `h.a.yousefi70@gmail.com`, `z_ahmadian@sbu.ac.ir`

ISSN: 2008-2045 © 2020 ISC. All rights reserved.

scheme for smart grid should enable the aggregator to obtain the total amount of energy consumption data of a region, while keeping the individual user's data confidential from not only the external adversary but also the internal adversary (i.e. one or some malicious entities in the network which collude with eachother).

The other aspect of security is protecting the user's data integrity to ensure that it is received correctly by the network. In general, the integrity and authenticity of data are provided by the signature schemes. If this security goal is violated, i.e. a user's data can be forged by an attacker, the data aggregation purpose will be ruined totally, since attacker is able to falsify the victim user(s) data and consequently the total data of the region, without being detected. So, all the data aggregation schemes are equipped with a kind of signature scheme.

However, signing the plain or encrypted value of energy consumption data by the user reveals who this data belongs to. But, the authenticated data should be aggregated while the user's privacy is protected. The approach adopted for this purpose in most of the schemes is one of these two major ones: using blind factors in data encryption [6, 7] and using anonymous signatures [8, 9], each of which has its own benefits and challenges.

### 1.1 Related Work

There is a variety of related work in the literature. Some of the most recent ones are reviewd in this section. BAP protocol, proposed in [10], is based on a homomorphic encryption algorithm and a proposed identity-based signature algorithm to provide authentication and integrity of the protocol. SecGrid uses Intel Software Guard Extensions (SGX) as a building block [11]. In this protocol, the smart meters need to perform only AES encryption. In [12], a privacy-preserving multidimensional data aggregation scheme based on the ElGamal homomorphic cryptosystem is proposed. This protocol enjoys a distributed decryption scheme, resisting the coalition attack from the GW and the CC. In [13], a lightweight $t$-times homomorphic encryption scheme resistant to quantum attacks is proposed, based on which two efficient data aggregation schemes for small-size and medium-size smart grids are proposed. The identity-based metering data aggregation scheme proposed in [14] makes use of batch verification in the aggregator, for preserving the privacy and integrity of metering data. The security of this scheme is proved in the random oracle model. To reduce the data aggregators, [15] benefites from the aggregation trees using an identity-based signcryption scheme without oracles. In this protocol, the compressed signatures can be verified in batches. EBDA is another recent privacy-preserving data aggregation for smart grid which integrates edge computing and blockchain technology to design a three-layer architecture data aggregation scheme [16].

### 1.2 Our Contributions

In this paper we first analyze the security of two privacy preserving data aggregation schemes, SP$^2$DAS [9] and 3PDA [17]. In SP$^2$DAS the authenticity of data is based on a newly designed anonymous signature, which is claimed to be existentially unforgeable under the adaptive chosen message attack model. However, we show a polynomial time selective forgery attack under the key-only attack model for this scheme. Moreover, we present a selective forgery attack on the signature scheme used in 3PDA protocol, as well. This attack works in the known-message attack model and requires only two known pairs of message-signature. 3PDA uses a signature scheme which, in an incorrect way, is a modified version of the CL* signature [18].

We also present an improved version of another privacy preserving data aggregation scheme, called EPPA [3], by improving the broadcast encryption scheme used in. The improved version, has a decryption process twice as fast as the original one and a decryption key whose size is half of the original version. Moreover the ciphertext size is 3/4 of the original one. The semantic security of this scheme is also proven under the same assumption of the original one.

A primary version of this paper was presented in [19], which includes the proposed attacks on SP$^2$DAS and 3PDA. However, the rest of the contribution, i.e. improving EPPA, brought in Section 5 of this paper, is totally new.

The structure of this paper is as follows. In Section 2, we introduce the preliminaries required for the paper, including the bilinear pairing and CL* signature. In Section 3 we present the signature scheme used in SP$^2$DAS protocol and our forgery attack on this scheme. In Section 4, 3PDA signature scheme along with our attack on that is presented. In Section 5, the broadcast encryption scheme used in EPPA protocol and our improved version, including its security proof, is presented. Finally, Section 6 concludes our work.

## 2 Preliminaries

In this section, we review some preliminaries that is necessary for the rest of the paper, including the bilinear pairing and the original version of CL* signature scheme.

### 2.1 Bilinear Pairing

Bilinear pairing is an operation over finite groups that is widely used in verification process of many signature schemes. Let $G_1, G_2$ and $G_T$ be three multiplicative groups with the same prime order $q$, and let $g_1$ and $g_2$ be generators of groups $G_1$ and $G_2$, respectively. The map $e : G_1 \times G_2 \rightarrow G_T$ is called a bilinear map if it has the three following properties:

- **Bilinearity**: $\forall g \in G_1$ and $h \in G_2$ and $\forall a, b \in Z_p$, it holds that $e(g^a, h^b) = e(g, h)^{ab}$.
- **Non-degeneracy**: $e(g_1, g_2) \neq 1$, where $g_1$ and $g_2$ are generators of groups $G_1$ and $G_2$, respectively.
- **Efficiency**: The map $e(g, h)$ must be computed in an efficient way $\forall g \in G_1$ and $\forall h \in G_2$.

### 2.2 CL* Signature

In this section, we overview the CL* signature scheme, which is proposed in [18] and is secure under the LRSW assumption [20] which is defined in the following.

**Definition 1 (LRSW assumption [20]).** Consider a bilinedar pairing as defined in Section 2.1, where $G_1 = G_2 = G$ and $q \in \Theta(2^l)$ where $1^l$ is the security parameter. Let $X, Y \in G$, $X = g^x$ and $Y = g^y$. Let $\mathcal{O}_{X,Y}(\cdot)$ be an oracle that on input a value $m \in Z_q^*$, randomely $a \in G$ and outputs $A = (a, a^y, a^{x+mxy})$. The LRSW assumption says that for all PPT adversaries $(A)^{(\cdot)}$, $v(l)$ defined as follows is a negligible function:

$$\begin{aligned} &Pr[x \leftarrow Z_q; y \leftarrow Z_q; X = g^x; Y = g^y;\\ &(m, a, b, c) \leftarrow (A)^{\mathcal{O}}_{X,Y}(q, g, G, G_T, e, X, Y) : m \notin Q \wedge\\ &m \in Z_q^* \wedge a \in G \wedge b = a^y \wedge c = a^{x+mxy}] = v(l) \end{aligned}$$

where $Q$ is the set of queries that $\mathcal{A}$ moade to $\mathcal{O}_{X,Y}(\cdot)$.

The CL* Signature has two variants, one of which allows batch verification with only three pairing operations. This scheme consists of three phases which are explained bellow. Let $G$ and $G_T$ be two multiplicative cyclic groups of prime order $q$ over which a bilinear map $e : G \times G \rightarrow G_T$ is defined. Let $g$ be the generator of group $G$ and $H_1 : \{0,1\}^* \rightarrow G$, $H_2 : \{0,1\}^* \rightarrow G$ and $H_3 : \{0,1\}^* \rightarrow Z_q$ be three hash functions.

- **Key generation**: Signer choose a random number $x \in Z_q$ as the private key and sets $Y = g^x$ as the public key.
- **Signing**: Let $m \in M$ be the message that is supposed to be signed. Signer takes value $TS$ that is the time stamp, and computes $a = H_1(TS)$, $b = H_2(TS)$ and $w = H_3(TS||m)$, and signs the message as $\sigma = a^x b^{xw}$.
- **Verification**: To verify signature $\sigma$ over message $m$, verifier first computes $a = H_1(TS)$, $b = H_2(TS)$, and $w = H_3(m||TS)$ and accepts signature if equation $e(\sigma, g) = e(ab^w, Y)$ holds, otherwise it would be rejected. Also, verifier can verify all the signatures $\sigma_i$ ,$i = 1, 2, \ldots, n$ at the same time through the following batch verification equation.

$$e(\prod_{i=1}^{i=n} \sigma_i, g) = e(a, \prod_{i=1}^{i=n} Y_i) e(b, \prod_{i=1}^{i=n} Y_i^{w_i}) \quad (1)$$

## 3 Cryptanalysis of SP$^2$DAS

In 2013, a self-certified data aggregation scheme was proposed for smart grid with the aim of increasing computation efficiency and achieving privacy protection of end users [9]. This scheme was called SP$^2$DAS, as an abbreviation of Self-certified PKC-based Privacy preserving Data Aggregation Scheme.

In order to preserve the users' data privacy, this scheme uses a novel self-certified anonymous signature scheme and claims that this signature scheme ”is secure against existential universal forgery under adaptive chosen message attack”. In the following, without going into the details of other parts of the protocol, we explain this signature scheme only. Then our proposed attack will be explained which, contrary to their claim, is a *selective forgery attack under the key only attack model.*

### 3.1 Signature Scheme in SP$^2$DAS

This scheme composed of five phases of system setup, key generation, witness registration, signing and verification, which work as follows.

- **System setup**: In this phase, two cyclic groups $G_1$ and $G_2$ with the same prime order $p$ are chosen, over which a bilinear map $e : G_1 \times G_2 \rightarrow G_T$ is defined. Let $g_2$ be the generator of $G_2$ and $\varphi : G_2 \rightarrow G1$ be an isomorphic map. The TTP first chooses two hash functions $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : \{0,1\}^* \rightarrow Z_p$. Then, it chooses his own pair of private and public keys as $(msk, mpk) = (\alpha, g^\alpha)$ where $\alpha \in Z_p$ is a randomly chosen element. Finally, the TTP publishes the public parameters in the form of below:

$$(G_1, G_2, G_T, g_2, \varphi, p, e, H_1, H_2, mpk)$$

- **Key generation**: In this section, the user with identity $ID$ first chooses a random element $x \in Z_p$, then he calculates his public key as $pk = e(g, g)^x$.

- **WitReg**: Having generated the private key, users should be registered to the TTP. For this purpose, each user computes a proof of zero knowledge $\pi$ in the form of $pk\{x|v = g_2^{\alpha x}\}$ and sends $(ID, pk, v, \pi)$ to the TTP.

  Once TTP received this vector, it first calculates $e(g_1, v)$ and checks the validity of this vector by checking the following equation.

  $$e(g_1, v) = pk^{\alpha} \tag{2}$$

  If (2) holds, then TTP generates a witness as follows and sends it through a secure channel to the users.

  $$w = (v^{\frac{1}{\alpha}} H_1(ID))^{\frac{1}{\alpha}} \tag{3}$$

- **Anonymous signing**: After receiving witness from TTP, the user can generate a signature through the following steps.
  (1) He first chooses a random element $l \in Z_p$ and computes $\hat{w} = \varphi(w)^l$ and $\hat{R} = \varphi(H_1(ID))^l$.
  (2) Then, he chooses a random element $r \in Z_p$ and computes the following elements.

  $$\begin{aligned} \hat{u} &= g_1^r \\ \hat{t} &= \frac{1 - rH_2(m\|\hat{u}\|\hat{R}\|\hat{w})}{lx} \end{aligned} \tag{4}$$

  (3) Having calculated the above parameters, the user's signature over message $m$ would be $\sigma = (\hat{w}, \hat{R}, \hat{u}, \hat{t})$.

- **Verification:** For verification of the four tuple $(\hat{w}, \hat{R}, \hat{u}, \hat{t})$ as a signature over message $m$, the verifier firstly computes $h_2 = H_2(m\|\hat{u}\|\hat{R}\|\hat{w})$ and verifies the signature through the following equation:

  $$e(\hat{w}^{\hat{t}}, mpk)e(\hat{u}^{h_2}\hat{R}^{-\hat{t}}, g_2) = e(g_1, g_2) \tag{5}$$

In Lemma 1 of [9], it is claimed that ”If there exists an adversary $A$ who can forge the previous anonymous signature on a message $m$, then the $(k+1)$-Exponent Problem (EP) can be solved in the polynomial time”. The $(k+1)$-EP, which is assumed to be hard there, is as follows according to [9]. Let $G_1$ be a multiplicative cyclic group of order $q$ with generator $g_1$. Given $k+1$ values $(g_1, g_1^a, g_1^{a^2}, \ldots, g_1^{a^k})$, where $k$ is an integer and $a \in Z_q$, compute $g_1^{a^{k+1}}$.

A proof for the mentioned lemma under the hardness of $(k+1)$-EP is given in [9]. However, there are some ambiguities regarding this proof. For example it is not clear that how the two sides of relation (13) of [9] has been considered equivalent. However, without exploiting this drawback, in the next part our key-only selective forgery attack on this signature is presented.

### 3.2 Key-Only Selective Forgery Attack on SP$^2$DAS Signature

The SP$^2$DAS signature involves four random elements while in its verification, according to (5), the right side is always constant. We show that the attacker is free enough to choose these random elements in such a way that (5) holds. Using the bilinear pairing properties, (5) can be rewritten as follows.

$$e(\hat{w}^{\hat{t}\alpha}\hat{u}^{h_2}\hat{R}^{-\hat{t}}, g_2) = e(g_1, g_2) \tag{6}$$

The Attacker first chooses a random number $\beta_1 \in Z_p$ and sets $\hat{w} = g_1^{\beta_1}$. Then, he chooses another random number $k \in Z_p$ and sets $\hat{u} = g_1^k$. Moreover, according to the isomorphism property of $\varphi$, it holds that

$$\varphi(g_2^{\alpha}) = g_1^{\alpha} \tag{7}$$

Then, attacker chooses $\hat{R}$ in the following form:

$$\hat{R} = \varphi(g_2^{\alpha})^{\beta_1} g_1^{\beta_2} = g_1^{\beta_1\alpha} g_1^{\beta_2} = g_1^{\beta_1\alpha+\beta_2} \tag{8}$$

Let $m'$ be the message for which the signature is supposed to be forged. The attacker computes $h_2 = H(m'\|\hat{w}\|\hat{u}\|\hat{R})$ and $\hat{t} = -\beta_2^{-1} + kh_2\beta_2^{-1}$ and finally generates the signature as the four tuple $\sigma = (\hat{w}, \hat{u}, \hat{R}, \hat{t})$.

Now we show that such a forged signature is accepted by the verifier since it is consistent with (5). For this signature, the left side of (5) is simplified to

$$\begin{aligned} e(\hat{w}^{\hat{t}}, mpk)e(\hat{u}^{h_2}\hat{R}^{-\hat{t}}, g_2) &= e(g_1^{\beta_1\hat{t}} g_1^{kh_2} g_1^{(\beta_1\alpha+\beta_2)-\hat{t}}, g_2) \\ &= e(g_1^{\beta_1\alpha\hat{t}+kh_2-\beta_1\alpha\hat{t}-\beta_2\hat{t}}, g_2) \\ &= e(g_1, g_2) \end{aligned} \tag{9}$$

which is equal to the right side of (5). The last line equality of (9) is concluded according to the followings.

$$\begin{aligned} \beta_1\alpha\hat{t} + kh_2 - \beta_1\alpha\hat{t} - \beta_2\hat{t} &= kh_2 - \beta_2\hat{t} \\ &= kh_2 - \beta_2(-\beta_2^{-1} + kh_2\beta_2^{-1}) \\ &= kh_2 + 1 - kh_2 = 1 \end{aligned} \tag{10}$$

So, the forged signature can pass the verification process successfully. This attack does not need any valid message-signature pair, so it work under the key-only model. Furthermore, it can forge valid signature for any arbitrary message $m'$ in polynomial time, hence it is a selective forgery attack.

## 4 Cryptanalysis of 3PDA

3PDA [17], the Practical Privacy Preserving Data Aggregation scheme, was proposed in 2019, as a TTP-free scheme aiming to reduce the computation cost and communication overhead, along with providing the security. In 3PDA, enjoying the bath verification capability of the CL* signature, a modified version of that is used for users' data authenticity. However, for its security, it has been referred to the security of the CL* signature: "In 3PDA, all data, sent from SMs and DCU, are signed with CL* signature method, which is based on LRSW assumption, and provably secure under the random oracle model."

In this section, we first review the signature used in 3PDA scheme and highlight its modifications. Then, we proposed our attack on this signature which is a *selective forgery attack under the known message attack model*. Again, we avoid bringing other details of the protocol here. The interested reader can refer to [17].

### 4.1 Signature Scheme in 3PDA

This signature is a modified version of CL* signature. Similar to CL*, it contains three phases, as follows.

- **Key generation**: This signature scheme runs over the elliptic curve $E(F_P)$ with order $q$ and three generators $G_1, G_2$ and $G_3$. It is assumed that there are $n$ users $U_i, i = 1, 2 \ldots, n$ with secret key $x_i \in Z_{q^*}$ and public key $Y_i = x_i G_1$ and there is a hash function $H_1 : \{0,1\}^* \to Z_q$. The public parameters are as follow:
$$\{E(F_p), q, G_1, G_2, G_3, H_1(\cdot)\}$$
- **Signing**: Each user $U_i, i = 1, 2, \ldots, n$ by using his private key $x_i$ calculates signature $\sigma_i$ over message $m_i$ as follows:
$$\sigma_i = x_i G_2 + x_i H_1(m_i) G_3 \qquad (11)$$
Comparing to the CL* signature, it can be seem that the time stamp dependent hash values $a$ and $b$ used in the original version of CL* scheme has been replaced with constant generators $G_2$ and $G_3$.
- **Verification**: After receiving all the signatures and message pairs $(m_i, \sigma_i)$ , $i = 1, 2, \ldots, n$ the verifier checks the validity of all signatures in batch, as follows:
$$e(\sum_{i=1}^{n} \sigma_i, G_1) = e(G_2, \sum_{i=1}^{n} Y_i) e(G_3, \sum_{i=1}^{n} (H_1(m_i) Y_i)). \qquad (12)$$
If (12) holds, verifier accepts all the signatures otherwise the batch verification process fails and the verifier should verify each signature separately to find the invalid signature(s).

### 4.2 Known-Message Selective Forgery Attack on 3PDA Signature

In this section, we show that how the attacker can forge valid signature for any arbitrary message, given only two known message-signature pairs. This attack works for any user $U_i, i = 1, \ldots, n$, so without loss of generality, we drop the subscript $i$ in the following.

Suppose that the attacker intercepts the communication link between the user and aggregator in data aggregation phase of two arbitrary sessions and he obtains $(m_1, \sigma_1)$ and $(m_2, \sigma_2)$. So, the attacker can construct the following system of equations:

$$\begin{aligned} \sigma_1 &= xG_2 + xH_1(m_1)G_3 \\ \sigma_2 &= xG_2 + xH_1(m_2)G_3 \end{aligned} \qquad (13)$$

These equations are linear in unknowns $xG_2$ and $xG_3$. So, it can be solved by the attacker to derive these unknowns. The attacker first computes the difference of two signature which is equal to

$$\sigma_1 - \sigma_2 = x(H_1(m_1) - H_1(m_2))G_3 \qquad (14)$$

Then, having the value of messages $m_1$ and $m_2$ and public hash function $H_1(\cdot)$, he can easily calculate the value of $(H_1(m_1) - H_1(m_2))^{-1} \in \text{mod} q$ and derive $xG_3$ by

$$(\sigma_1 - \sigma_2)(H_1(m_1) - H_1(m_2))^{-1} = xG_3 \qquad (15)$$

Now having the value of $xG_3$, the attacker can easily calculate $xG_2$ as follows.

$$\sigma_1 - xH_1(m_1)G_3 = xG_2 \qquad (16)$$

Having derived $xG_2$ and $xG_3$, the attacker can forge signature for any arbitrary message $m'$ as follows:

$$\sigma' = xG_2 + H_1(m')xG_3 \qquad (17)$$

So this is a selective forgery attack, which requires only two known message-signature pairs.

## 5 Improvement of EPPA

An Efficient and Privacy-Preserving Aggregation scheme, named EPPA, was proposed in 2012 [3] for smart grid communications. This scheme mainly consists of the following four parts: system initialization, user report generation, privacy-preserving report aggregation, and secure report reading and response, among which we focus on the response process of the last part. In this process the trusted Operation Authority (OA) broadcasts an encrypted message to all the users of the residental area (RA) network to inform them from the total electricity usage of the area.

In this section, we first review the broadcast encryption process used in EPPA, then we propose a modified version of that which has a smaller decryption key, less decryption computations, and shorter ciphertext. We also prove the semantic security of this modified version of EPPA broadcast encryption in the chosen-plaintext attack model, under the assumption of hardness of the Decisional Bilinear Diffie-Hellman (DBDH) problem.

**Application.** The boradcast encryption is a one-to-many encryption scheme, from the GW to all the users of the RA, providing a considerable reductoion in the computational complexity of the encryption process in the GW side, comparing to the conventional one-to-one encryption schemes.

The mentioned improvements in the decryption key size, decryption computations and ciphertext size, in the proposed scheme, all affects on the residental area scale, which highlights their importance more.

### 5.1 EPPA Broadcast Cryptosystem

EPPA broadcast cryptosystem consists of three phases: initialization, broadcast encryption and decryption that explained in below.

- **Initialization**: OA chooses two cyclic additive elliptic curve groups $G$ and $G_T$ both with order $q$, equipped with a bilinear map $e : G \times G \rightarrow G_T$. It also chooses two random elements $Q_1, Q_2 \in G$ and two random numbers $\alpha, x \in Z_q^*$ as its master key. Then, it computes $e(P,P)^\alpha$ and $Y = xP$, where $P$ is generator of group $G$. Then, OA chooses hash function $H_1 : \{0,1\}^* \in Z_q^*$ and publishes the following parameters.

$$\{Q_1, Q_2, e(P,P)^\alpha, Y, H_1(\cdot), q, P, G, G_T, e\} \tag{18}$$

  When user $U_i, i = 1, 2, \ldots, n$, located in the residential area $RA$, wants to register in the system, it first chooses a random number $x_i \in Z_q^*$ as its private key and computes $Y_i = x_i P$ as its public key. Also, OA computes the following $RA$-dependent parameters using its master key.

$$\begin{aligned} t_{i_1} &= H_1(U_i \parallel RA \parallel \alpha) \\ t_{i_2} &= H_1(U_i \parallel RA \parallel x) \end{aligned} \tag{19}$$

  and finally it computes the user's authorized key $ak_i$ and assigns it to user $U_i$ as follows.

$$ak_i = (\alpha P + t_{i_1} Y, t_{i_1} P, t_{i_2} P, t_{i_1} Q_1 + t_{i_2} Q_2)$$

- **Broadcast encryption**: Having analyzed the aggregated data, OA generates a response $m$ to inform the users from the electricity usage and control their cost. In order to encrypt this message for all users simultaneously, OA chooses a random number $s \in Z_q^*$ and computes the ciphertext $\bar{C} = (\bar{C}_1, \bar{C}_2, \bar{C}_3, \bar{C}_4)$ as follows

$$\begin{aligned} \bar{C}_1 &= me(P,P)^{\alpha s}, \\ \bar{C}_2 &= sP, \\ \bar{C}_3 &= sY - sQ_1, \\ \bar{C}_4 &= -sQ_2. \end{aligned} \tag{20}$$

  OA broadcasts this ciphertext among all the users $U_i, i = 1, 2, \ldots, n$ of the home area network.

- **Decryption:** Upon receiving $\bar{C}$, each user $U_i$ decrypts $\bar{C}$ by using its authorized key $ak_i$, according to the two following steps. It first obtains $e(P,P)^{\alpha s}$ as follows.

$$\begin{aligned} &\frac{e(\bar{C}_2, \alpha P + t_{i_1} Y)}{e(t_{i_1} P, \bar{C}_3) e(t_{i_2} P, \bar{C}_4) e(t_{i_1} Q_1 + t_{i_2} Q_2, \bar{C}_2)} \\ &= \frac{e(sP, \alpha P + t_{i_1} Y)}{e(t_{i_1} P, sY - sQ_1) e(t_{i_2} P, -sQ_2)} \times \\ &\quad \frac{1}{e(t_{i_1} Q_1 + t_{i_2} Q_2, sP)} \end{aligned} \tag{21}$$

  The denumerator of (21), which we call $D$, can be simplified as follows.

$$\begin{aligned} D &= e(t_{i_1} P, sY) e(t_{i_1} P, -sQ_1) e(t_{i_2} P, -sQ_2) \times \\ &\quad e(t_{i_1} Q_1, sP) e(t_{i_2} Q_2, sP) \\ &= e(t_{i_1} P, sY) \end{aligned} \tag{22}$$

  Using (22), equation (21) is equal to:

$$\begin{aligned} \frac{e(sP, \alpha P + t_{i_1} Y)}{e(t_{i_1} P, sY)} &= \frac{e(sP, \alpha P) e(sP, t_{i_1} Y)}{e(t_{i_1} P, sY)} \\ &= e(P,P)^{\alpha s}. \end{aligned} \tag{23}$$

  Then, message $m$ is computed using the following relation.

$$\frac{\bar{C}_1}{e(P,P)^{\alpha s}} = \frac{me(P,P)^{\alpha s}}{e(P,P)^{\alpha s}} = m \tag{24}$$

### 5.2 Improved EPPA Broadcast Encryption Scheme

The modified version of the EPPA broadcast encryption scheme is presented here. This new version, while being provable secure under the same assumption, shows a great improvement in the computational complexity, size of the authorized key and length of the ciphertext, comparing to the original scheme.

In the modified scheme, the definitions of all parameters are the same as the original one, except for the following ones. The parameters $Q1$, $Q2$ and $t_{i_2}$ does not need to be defined any more, and the parameter $t_{i_1}$ is defined as previous $t_{i_1} = H_1(U_i \parallel RA \parallel \alpha)$. The authorized key for user $U_i$ is defined as

$$ak_i = (\alpha P + t_{i_1} Y, t_{i_1} P) \tag{25}$$

whose size can be seen that is half of the original one. Moreover, the ciphertext of message $m$, denoted by $C = (C_1, C_2, C_3)$, is generated as follows.

$$\begin{aligned} C_1 &= me(P,P)^{\alpha s} \\ C_2 &= sY, \\ C_3 &= sP \end{aligned} \tag{26}$$

By comparing (26) with (20), we can see that the size of the ciphertext is equl to three elements of group, which is shorter than the original one which was equal to four elements.

Finally, in the modified version, the decryption process is performed by user $U_i$ in the two following steps. Firstly,

$$\frac{e(\alpha P + t_{i_1}Y, sP)}{e(t_{i_1}P, sY)} = \frac{e(\alpha P, sP)e(t_{i_1}Y, sP)}{e(t_{i_1}Y, sP)} = e(P,P)^{\alpha s}. \tag{27}$$

Secondly,

$$\frac{C_1}{e(P,P)^{\alpha s}} = \frac{me(P,P)^{\alpha s}}{e(P,P)^{\alpha s}} = m. \tag{28}$$

Comparing (27) with (21), it is obvious that the number of pairing operations in decryption process of the modified scheme has been reduced to two out of four in the original one. Noting that the cost of pairing operation is much higher than the other operations in $G$ and $G_T$, our improved scheme halves the computational complexity of the decryption process.

Therefore, this improved version of EPPA broadcast encryption, has a half-length authorized key, a 3/4-length ciphertext, and a half-computational complexity decryption comparing to the original EPPA broadcast scheme.

### 5.3 Semantic Security of the Improved Scheme

In this scheme we prove the semantic security of the proposed scheme in the chosen plaintext attack model with the same assumption under which the security of the original EPPA scheme is proved [3]. we first review this assumption, then provide our proof.

**Definition 2 (Dicisional bilinear Deffie-Helman problem).** Dicisional bilinear Deffie-Helman (DBDH) problem in $G$ and $G_T$ cyclic groups with a bilinear map $e : G \times G \to G_T$ is stated as follows. Given $(P, aP, bP, cP, W)$ where $a, b, c \in Z_{q^*}$ are unknown numbers, $P$ is a generator of $G$ and $W \in G_T$, decide if $W = e(P,P)^{abc}$ or a random element drawn from $G_T$.

**Theorem 1.** *The Ciphertext $C = (C_1, C_2.C_3)$, defined in (26), is semantic secure against the chosen plaintext attack under the DBDH hardness assumption.*

*Proof.* Let $a, b, c \in Z_q^*$ and $\tilde{b} \in \{0,1\}$ if $\tilde{b} = 0$, $W = e(P,P)^{abc}$ else $W$ is a random element of $G_T$. Therefore, given the instance $(P, aP, bP, cP, W)$, the DBDH problem is equal to guess $\tilde{b}$. Now, suppose there is an adversary $\mathcal{B}$ which is able to break the semantic security of ciphertext $C$ in polynomial time with non-negligible advantage $\epsilon$. We show that if there exists such an adversary, we can construct another adversary $\mathcal{A}$ that utilizes adversary $\mathcal{B}$ and is able to solve DBDH problem in polynomial time with non-negligible advantage.

Suppose attacker $\mathcal{A}$ receives the DBDH problem instance $(P, aP, bP, cP, W)$ as input. $\mathcal{A}$ chooses two random numbers $\alpha', \beta \in Z_q^*$ and computes public parameters of the scheme, $Y$ and $e(P,P)^\alpha$, as follows.

$$\begin{aligned} Y &= \beta P \\ e(P,P)^\alpha &= e(aP, bP)e(P,P)^{\alpha'} \end{aligned} \tag{29}$$

In fact, comparing with the original scheme, $\mathcal{A}$ has implicitely set $x = \beta$ and $\alpha = ab + \alpha'$. Since $bP, \alpha', \beta$ and $aP$ have been chosen randomly, the distribution of the simulated parameters $(Y, e(P,P)^{\alpha'})$ does not change. In other words, adversary $\mathcal{B}$ can not distinguish between these simulated elements with real elements of the system.

Then, adversary $\mathcal{A}$ sends parameters $(Y, e(P,P)^\alpha)$ to $\mathcal{B}$. Upon receiving that, $\mathcal{B}$ chooses two messages $m_0, m_1 \in G$ and sends it to $\mathcal{A}$. $\mathcal{A}$ chooses a random bit $b^* \in \{0,1\}$ and computes the ciphertext of $m_{b^*}$, $C = (C_1, C_2, C_3)$, as follows,

$$\begin{aligned} C_1 &= m_{b^*}We(cP, \alpha' P) \\ C_2 &= \beta \cdot cP \\ C_3 &= cP \end{aligned} \tag{30}$$

and sends it to $\mathcal{B}$. Upon receiving $C$, $\mathcal{B}$ returns $b'$ as a guess for $b^*$ and sends it to $\mathcal{A}$. If $b' = b^*$, $\mathcal{A}$ guesses $\tilde{b} = 0$, which means $W = e(P,P)^{abc}$. Considering $\alpha' = \alpha - ab$, we have:

$$\begin{aligned} C_1 &= m_{b^*}We(cP, \alpha' P) \\ &= m_{b^*}e(P,P)^{abc}e(cP, \alpha' P) \\ &= m_{b^*}e(P,P)^{abc+\alpha' c} \\ &= m_{b^*}e(P,P)^{\alpha c} \end{aligned} \tag{31}$$

Equation (31) shows that $C_1$ is a valid component of ciphertext $C$. If $\mathcal{B}$ correctly guesses $b^*$ with probability $\frac{1}{2} + \epsilon$, then $Pr[\mathcal{A} \text{ success}|\tilde{b} = 0] = \frac{1}{2} + \epsilon$. In case of a false guess for $b^*$, i.e. $\tilde{b} = 1$, $C_1$ will be independent of $b^*$ and $Pr[\mathcal{A} \text{ success}|\tilde{b} = 1] = \frac{1}{2}$. Thus, the success probability of $A$ can be computed as follows.

$$Pr[\mathcal{A} \text{ success} = \frac{1}{2}(\frac{1}{2} + \epsilon) + \frac{1}{2}.\frac{1}{2} = \frac{1}{2} + \frac{\epsilon}{2}, \tag{32}$$

which means that $\mathcal{A}$ can break the DBDH problem with non-negligible advantage $\frac{\epsilon}{2}$, a contradiction with hardness assumption of DBDH. Thus ciphertext $C = (C_1, C_2, C_3)$ is semantic secure under the hardness of DBDH problem assumption. □

## 6 Conclusion

In the design of security protocols, efficiency and security are two important design goals that are often in the tradeoff. The same is true for the smart grid data aggregation schemes. This paper examined some of the proposed protocols in this domain from these two perspectives.

The first one, SP$^2$DAS, uses an anonymous signature scheme, which is claimed to be existentially unforgeable under the adaptive chosen message attack. However, we proposed a selective forgery attack in the key-only model for this protocol. The next protocol, 3PDA, uses a modified version of the CL* signature, which is claimed to be as secure as CL*. But, according to our proposed attack, this scheme is broken by a selective forgery attack in the known message attack model. Finally, the third protocol, EPPA, does not suffer from any security drawback, however, its broadcast encryption scheme was so inefficient. We proposed an improved broadcast encryption scheme for this protocol, in which the decryption key is half, the complexity of decryption is half, and the ciphertext size is 3/4 of the original one. Moreover, we proved the semantic security of our proposed protocol.

## References

[1] Rongxing Lu. Differentially private data aggregation with fault tolerance. In *Privacy-Enhancing Aggregation Techniques for Smart Grid Communications*, pages 129–151. Springer, 2016.

[2] Gergely Ács and Claude Castelluccia. I have a dream!(differentially private smart metering). In *International Workshop on Information Hiding*, pages 118–132. Springer, 2011.

[3] Rongxing Lu, Xiaohui Liang, Xu Li, Xiaodong Lin, and Xuemin Shen. Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Transactions on Parallel and Distributed Systems*, 23(9):1621–1631, 2012.

[4] Fabian Knirsch, Günther Eibl, and Dominik Engel. Error-resilient masking approaches for privacy preserving data aggregation. *IEEE Transactions on Smart Grid*, 9(4):3351–3361, 2016.

[5] Asmaa Abdallah and Xuemin Sherman Shen. A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. *IEEE Transactions on Smart Grid*, 9(1):396–405, 2016.

[6] Erfaneh Vahedi, Majid Bayat, Mohammad Reza Pakravan, and Mohammad Reza Aref. A secure ecc-based privacy preserving data aggregation scheme for smart grids. *Computer Networks*, 129:28–36, 2017.

[7] Chun-I Fan, Shi-Yuan Huang, and Yih-Loong Lai. Privacy-enhanced data aggregation scheme against internal attackers in smart grid. *IEEE Transactions on Industrial informatics*, 10(1):666–675, 2014.

[8] Xuefeng Liu, Yuqing Zhang, Boyang Wang, and Huaqun Wang. An anonymous data aggregation scheme for smart grid systems. *Security and communication networks*, 7(3):602–610, 2014.

[9] Jianhong Zhang, Liying Liu, Yuanbo Cui, and Zhipeng Chen. Sp2das: self-certified pkc-based privacy-preserving data aggregation scheme in smart grid. *International Journal of Distributed Sensor Networks*, 9(1):457325, 2013.

[10] Zhiyuan Sui and Hermann de Meer. Bap: a batch and auditable privacy preservation scheme for demand response in smart grids. *IEEE Transactions on Industrial Informatics*, 16(2):842–853, 2019.

[11] Shaohua Li, Kaiping Xue, David SL Wei, Hao Yue, Nenghai Yu, and Peilin Hong. Secgrid: A secure and efficient sgx-enabled smart grid system with rich functionalities. *IEEE Transactions on Information Forensics and Security*, 15:1318–1330, 2019.

[12] Xiangjian Zuo, Lixiang Li, Haipeng Peng, Shoushan Luo, and Yixian Yang. Privacy-preserving multidimensional data aggregation scheme without trusted authority in smart grid. *IEEE Systems Journal*, 15(1):395–406, 2020.

[13] Jiawei Qian, Zhenfu Cao, Xiaolei Dong, Jiachen Shen, Zhusen Liu, and Yunxiu Ye. Two secure and efficient lightweight data aggregation schemes for smart grid. *IEEE Transactions on Smart Grid*, 12(3):2625–2637, 2020.

[14] Yong Ding, Bingyao Wang, Yujue Wang, Kun Zhang, and Huiyong Wang. Secure metering data aggregation with batch verification in industrial smart grid. *IEEE Transactions on Industrial Informatics*, 16(10):6607–6616, 2020.

[15] Zhiyuan Sui and Hermann de Meer. An efficient signcryption protocol for hop-by-hop data aggregations in smart grids. *IEEE Journal on Selected Areas in Communications*, 38(1):132–140, 2019.

[16] Weifeng Lu, Zhihao Ren, Jia Xu, and Siguang Chen. Edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid. *IEEE Transactions on Network and Service Management*, 18(2):1246–1259, 2021.

[17] Yining Liu, Wei Guo, Chun-I Fan, Liang Chang, and Chi Cheng. A practical privacy-preserving data aggregation (3pda) scheme for smart grid. *IEEE Transactions on Industrial Informatics*, 15(3):1767–1774, 2019.

[18] Jan Camenisch, Susan Hohenberger, and Michael Østergaard Pedersen. Batch verification of short signatures. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 246–263. Springer, 2007.

[19] Hamid Amiryousefi and Zahra Ahmadian. Crypt-

analysis of sp 2 das and 3pda, two data aggregation schemes for smart grid. In *2019 16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, pages 45–48. IEEE, 2019.

[20] Anna Lysyanskaya, Ronald L Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In *International Workshop on Selected Areas in Cryptography*, pages 184–199. Springer, 1999.

**Hamid Amiryousefi** received his B.S. degree in electrical engineering from Semnan University, Semnan, Iran, in 2016, and his M.Sc. degree in electrical engineering (communication systems) from Shahid Beheshti University, Tehran, Iran, in 2019. His research interests lie in the area of IoT security, blockchain, public key cryptography and wireless communication systems.

**Zahra Ahmadian** received the B.Sc. degree in electrical engineering (communications and electronics fields) from Amirkabir University of Technology, Tehran, Iran, in 2006, and the M.Sc. degree in electrical engineering (secure communications) and Ph.D. degree in electrical engineering (communication systems) both from Sharif University of Technology, Tehran, in 2008 and 2014 respectively. Since 2014, she has been with the Electrical Engineering Department of Shahid Beheshti University, Tehran, as an assistant professor. Her special fields of interests include wireless security and cryptology with an emphasis on cryptanalysis.