Enabling Anonymous Authorization and Rewarding in the Smart Grid

Tassos Dimitriou, *Senior Member, IEEE* and Ghassan O. Karame, *Member, IEEE*

Abstract—The smart grid leverages infrastructural support to achieve fine-grained power consumption monitoring in an attempt to offer higher efficiency, reliability, and security. Such functionality, however, requires the collection of fine-grained usage data which may raise serious concerns with respect to consumer privacy. Thus far, existing work has solely focused on the problem of privately aggregating energy measurements. However, these solutions do not allow the provider to acquire detailed energy measurements which are essential for maintaining the network, debugging configuration problems, etc. In this work, we address this problem and we propose an authentication scheme that allows a smart meter to anonymously interact with the utility provider when submitting detailed consumption data. We then move one step further, enabling the incorporation of anonymous rewarding mechanisms in the smart grid in exchange for detailed measurements that users report. We argue that such rewarding mechanisms provide solid incentives for users to accept the release of their detailed energy consumption; we show that our proposal does notleak any information about the identity of users—even when redeeming the rewards. Finally, we implement a prototype based on our proposal and we evaluate its performance in realistic deployment settings.

Index Terms—Smart grid, privacy, anonymous data reporting, anonymous rewarding

1 Introduction

THE electrical grid has undergone a major transformation with the introduction of communication-infrastructure support for the creation of a "smarter" grid. This smart grid combines traditional grid technologies, which allow the flow of electricity from plants to customers, with information and control technologies that enable a two-way transmission of electricity in an attempt to offer increased efficiency and reliability.

At the core of the smart grid lie household metering devices that can record and communicate consumption of energy data to the central system for monitoring and billing purposes. These intelligent devices help provide an unprecedented degree of contextual awareness about the state of the electric power grid, thus reducing outages, improving reliability and benefiting both users and electricity providers from a balanced utilization of energy and lower costs.

A big challenge, however, for the deployment of smart grid technologies on a larger scale lies in sustaining an acceptable level of user privacy and anonymity. Indeed, while smart grid deployments have been supported (and even driven) by governments around the world, there are serious concerns with respect to the privacy-invasive character of these technologies. For example, the frequent collection of energy data reveals private information about behaviors, activities or preferences of inhabitants [1]. Household readings can also be used to eavesdrop at activities within homes, determine if a person is at home, what type of appliances the person possess, how many hours of sleep did the person get last night, or even what are the religious beliefs of that person [2], [3].

So far, research has focused on the development of various privacy-preserving technologies that help ensure meter privacy by aggregating meter readings and thus preventing the provider from acquiring fine-grained consumption measurements. However, such solutions do not entirely solve the problem. Indeed, in some settings, the utility provider (UP) needs to collect detailed energy reports from the smart meters, e.g., to maintain the grid or debug a configuration problem. Given that most users are reluctant on submitting their detailed measurements to the *UP*, *rewarding* mechanisms constitute one of the few workable mechanisms to solicit user collaboration in this case. For instance, cost reductions, additional services, and/or monetary remuneration can be used by the *UP* to convince users to temporarily give away parts of their privacy and release detailed measurements over a short period of time.

Here, a problem that remains largely unexplored is how to minimize information leakage that is associated with the (frequent) authentication of reported data, and the issuance/redemption of rewards. Such a leakage is especially detrimental since it allows the *UP* to gain additional knowledge about users (that it would not have gained otherwise) and to couple it with their energy consumption patterns as a means to better profile them. What makes this problem even more challenging is that security and anonymity are two *contradictory* objectives; on one hand, the utility provider should be able to authenticate the source of the measurements, while on the other hand, the identity of the meter/user (and other contextual information) should not be revealed throughout the entire process.

In this work, we propose an efficient scheme to anonymously authenticate frequent energy consumption reports sent by meters. Our scheme leverages blind signatures and hash chains to provide implicit authentication between a meter and the *UP*. We then extend our scheme to incorporate anonymous tokens which can be used by the utility provider as a means to reward those users who share details about their energy consumption; we show that this process does not leak any information about the identity of users and cannot be used in any way by the *UP* to track users or profile them. We also evaluate and implement our solution in a realistic smart grid setting; our findings show that our proposal does not deteriorate the performance of the system.

This paper extends and improves our prior work in [19] with significant new material. More specifically, our contributions can be summarized as follows:

- We present and analyze an efficient authorization scheme that allows a smart meter to anonymously authenticate its consumption data exchanged with the utility provider.
- We extend our solution and propose the generation of tokens which are anonymously associated with the meters' consumed energy. We thoroughly analyze the security of our extended proposal and we show that such anonymous tokens can be used by the utility provider to provide rewards for users in exchange of their fine-grained consumption details.
- We implement a prototype based on our proposals and we evaluate its performance in realistic settings. Our results suggest that our scheme scales well with the number of meters in the system and does not incur significant overhead on the meters/utility provider.

The remainder of the paper is structured as follows. In Section 2, we outline our model and assumptions. In Section 3, we show how anonymous authorization and reporting can be achieved through the combination of hash chains and blind signatures. In Section 4, we extend our scheme with the notion of anonymous reward tokens to enable privacy-preserving rewarding in smart grids, and we analyze the security and performance of our extended proposal in Section 5. In Section 6, we overview related work in the area, and we conclude the paper in Section 7.

- *T. Dimitriou is with the Computer Engineering Department, Kuwait University, Kuwait. E-mail: tassos.dimitriou@ieee.org.*
- *G.O. Karame is with the NEC Laboratories Europe, Heidelberg 69115, Germany. E-mail: ghassan.karame@neclab.eu.*

Manuscript received 31 Mar. 2015; revised 31 Aug. 2015; accepted 24 Oct. 2015. Date of publication 29 Oct. 2015; date of current version 1 Sept. 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TDSC.2015.2496211

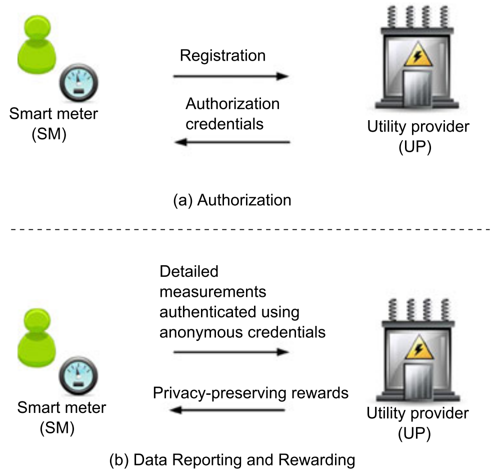

Smart meter (SM)
Registration
Authorization credentials
Utility provider (UP)
(a) Authorization
Detailed measurements authenticated using anonymous credentials
Privacy-preserving rewards
Smart meter (SM)
Utility provider (UP)
(b) Data Reporting and Rewarding

Fig. 1. Sketch of our model.

2 Model and Notation

We consider a smart grid network comprising of N Smart Meters (SMs) (associated with different households) which are connected to a central Utility Provider. We assume that each SM has access to the Internet, at least intermittently, through some open-access WiFi infrastructure. We also assume that SMs feature secure storage and autonomous cryptographic functionality (e.g., using TPM chips [21]). Typically, the UP interacts with individual SMs by exchanging price information, meter data, and control commands. We assume that the communication channel between the UP and the meters does not provide information about the identity of the meters or the location of the household. This can be achieved, e.g., by sending data through an anonymization network [22].

In this work, we consider the setting in which certain SMs can be tasked to report relevant contextual information, send detailed measurements to the UP at a given interval (e.g., to debug a configuration problem).

Threat model. In the sequel, we assume that each smart meter S is preloaded with a set of public-private keys (K_S, K_S^{-1}).

Although the smart grid can truly benefit from the collection of fine-grained data points to better configure the network, privacy risks render users rather reluctant on submitting detailed measurements to the UP. In this work, we consider *rewarding* mechanisms as one of the few workable mechanisms to solicit user collaboration. For instance, monetary remuneration can be used by the UP to convince users to temporarily give away parts of their privacy and release detailed measurements over a short period of time. By doing so, the UP does not only increase the amount of collected data but can also better maintain and configure the grid, eventually increasing reliability and offering an improved service back to the users. As shown in Fig. 1, a careful design of data reporting/rewarding mechanisms is required in order to ensure the privacy and anonymity of users who are redeeming the rewards, namely:

- During the detailed data reporting phase, smart meter anonymity should be preserved. Neither the utility provider nor other users of the system should be able to learn anything about the identity of the smart meter. Context information (access patterns, requests, etc.) should also be protected against external and internal eavesdroppers who interact with the user/smart meter.
- Only authenticated meters should be able to submit detailed energy consumption data. The utility provider should be able to verify that the data originates from a legitimate smart meter/owner.
- The interactions between a smart meter and the utility provider should be protected from anyone who is not authorized to have access to this data. Reported data should also be protected from modifications.
- Rewards should be redeemed only once by users. Any double-spending attempt of the same token should be easily detectable and users should not be able to forge their own rewards.
- Finally, the rewarding mechanism should not leak any information about the underlying user, and rewards should be unlinkable to each other and to the data reported by the meters. Otherwise, users risk leaking additional details about spending patterns, types of services they request, etc. This is captured by the following definition (see also the unlinkability game defined in Section 5.2).

Definition 1. *We say that two rewards are unlinkable if an adversary $\mathcal{A}$ cannot tell with more than $1/2 + \epsilon$ probability of correctness if these rewards are associated with the same meter. We say that the anonymous rewarding system is ϵ-unlinkable if the advantage of* any *distinguishing adversary is bounded by ϵ.*

Design goals. In addition to the security goals, our solution should avoid reliance on third parties. This is especially important when issuing/spending reward tokens, or when reporting detailed data measurements. Moreover, our solution must be computationally- and communication-efficient. For instance, since our protocols will be executed on resource-constrained devices (e.g., SMs), reliance on expensive cryptographic operations should be avoided.

Related approaches. In the context of reporting detailed energy measurements, one possible way to realize anonymous authentication of the smart meters would be to rely on group signature schemes. Such schemes allow any member of a group to sign a message on behalf of the entire group—thus hiding its identity. However, existing group signature schemes are either *(i)* decentralized and thus do not scale well with the number of members, e.g., to agree on group keys, or *(ii)* are centralized but enable the central entity to revoke anonymity of members in case of need [4]; clearly, this can be (ab-)used by a malicious UP to identify meters, and track their actions in the grid.

On the other hand, one possible way to issue anonymous (monetary) rewards would be for the UP to issue digital payments to their users, such as payments over Bitcoin [5], [27]. In Bitcoin, peers are anonymous, and referenced in each transaction by means of virtual (anonymous) pseudonyms—referred to as *Bitcoin addresses*. Although Bitcoin is gaining considerable popularity and adoption, most utility providers do not issue nor accept Bitcoins (e.g., due to the volatility of the currency). Moreover, rewarding users by means of Bitcoin payments requires these users to have active Bitcoin installations in order to collect their rewards.

In what follows, we present and analyze our proposals that support the privacy-preserving reporting of fine-grained consumption data in smart grids. We will do so incrementally, starting with an initial scheme which enables anonymous authorization based on blinded hash-chains, and later on extending it to construct rewarding mechanisms based on the use of efficient and anonymous tokens.

3 Anonymous Authorization in Smart Grids

We first begin by introducing our protocol which enables the efficient construction of anonymous credentials. We will then leverage this protocol to anonymously authenticate the detailed energy reports sent by meters.

3.1 Anonymous Credentials Using Hash Chains

Our authorization protocol is based on the use of blind RSA signatures and one-way hash chains. The outcome of this phase will be a set of authorization credentials that will allow the meters to subsequently submit detailed consumption reports in a privacyrespecting manner, without the need for additional, explicit authentication.

Our protocol unfolds in Algorithm 1. First, the meter picks a random seed R_S and sets it as the root of the chain $A_0 \leftarrow R_S$. The meter subsequently computes a hash chain of n credentials, where $A_i \leftarrow H(A_{i-1})$, given a cryptographic hash function $H(.)$. Once the hash chain is created (Steps 1 and 2), the meter submits A_n for authorization; this value will be the starting point for a sequence of authorization credentials $A_n, A_{n-1}, \ldots, A_0$. However, since A_n (and the rest of chain values) will be used in reporting user data later on, A_n has to be *blinded* before it is given to the utility provider.

Algorithm 1. Setup: Credential Generation and Authorization for a Smart Meter S

```
1: Pick a random number $R_S$ and set $A_0 \leftarrow R_S$.
2: Compute the chain of hash credentials $A_i \leftarrow H(A_{i-1})$, for $1 \leq i \leq n$.
3: Blind $A_n$ by computing $A^* \leftarrow b^e A_n \bmod N$, where $b$ is another random number.
4: Send $A^*$ and $Sig_S(A^*)$ to the utility provider.
5: After verifying the signature of $S$, the $UP$ returns the signature $\sigma^* \leftarrow (A^*)^d \bmod N$ to $s$.
6: The smart meter computes $\sigma \leftarrow b^{-1}\sigma^* \bmod N$ and recovers the $UP$'s signature on $A_n$.
```

Credentials are blinded in RSA as follows [6]. Let $N = pq$ be an RSA modulus, given primes p and q. We denote by (e, N) the public key of the utility provider UP and by d its corresponding private key. To obtain a blind signature from UP, the meter chooses a random b, computes b^e and multiples it with the credential value to blind it. The meter then asks the utility provider to sign the blinded version $A^* \leftarrow b^e A_n \bmod N$ (Steps 3-5). Note that the blinded credential A^* should be signed by the meter in order to allow the utility provider to verify that the blinded credential originates from a *legitimate* smart meter (Step 5). We denote the signature of A^* by the SM as $Sig_S(A^*)$. If this verification passes, then the UP signs the blinded version of A^* and gives it back to the meter; the latter checks the signature and removes the blinding factor b, thus obtaining the UP's signature on the unblinded credential A_n (Step 6). Although the provider is not aware of the value A_n, he can check its authenticity at reporting time, and verify the correctness of his own signature. Additionally, even though the UP only signed the value A_n, all the hashed credentials A_i, $i < n$, are implicitly authenticated as they hash to the last value A_n. Therefore, they can also be used for subsequent data reporting. Notice that this protocol needs to be executed when the hash chain is exhausted.

Alternatively, BLS signatures can also be used to blind A_n. BLS signatures are based on elliptic curves, and are therefore faster to computer than RSA [20]. However, signature verification in BLS incurs considerable overhead on the meters—which explains why we rely on RSA in our scheme.

3.2 Anonymous Authentication of Data Reports

We now show how to leverage the aforementioned credentials to anonymously authenticate the reporting of fine-grained energy measurements.

Assume that the smart meter obtains consumption values *cons* and outputs measurements $m = (cons,\ timestamp)$. Clearly, the meter needs to authenticate every such measurement (e.g., with its own signature) to prevent data forgery or reports sent by unregistered meters, etc. Existing authentication techniques, such as digital signatures or MACs, leak information about the identity of the meter. This is exactly where our authorization credentials come into play.

When a meter wants to send a consumption measurement, it can append it to an unused authorization credential—which will act as a proof that the measurement issuer is a legitimate user in the network. More specifically, the protocol begins by the meter sending the pair $\{m, A_j\}$, encrypted with the public key of the UP, e. By doing so, we ensure that *(i)* no other entity can learn the meter's consumption data m other than the utility provider itself, and that *(ii)* nobody can learn the authorization credential A_j used to authenticate the data. Otherwise, if A_j is sent in the clear, it can be intercepted by an adversary who can modify it, inject her own data, etc.

Upon receiving the credential A_j, the provider first checks whether this is the first submission by the meter (i.e., $j = n$), in which case the provider verifies its signature on A_n. If this test succeeds, UP accepts the measurement and stores A_n in its database of submitted authorization credentials. For any other value, the provider hashes A_j and compares it against the stored A_{j+1} in the chain indexed by A_n. The provider then updates the currently stored credential A_{j+1} with A_j, thus preventing a credential to be reused for data submission.

The aforementioned protocol enables an implicit authentication of the meters based on the authorization credential. Although the UP can tie two submissions to the *same* hash chain, the use of blind signatures ensures that no information can be tied to a particular user.[1] In Section 5.2, we show how this side channel can be eliminated.

4 Anonymous Reward Tokens

As mentioned earlier, one of the few workable ways to solicit user collaboration in acquiring fine-grained measurements would be to reward the users in exchange for their potential loss of privacy. In what follows, we extend our proposal in Section 3 to incorporate anonymous reward tokens.

4.1 Main Intuition

To enable anonymous rewarding, we show how to construct anonymous *tokens* which can be used by a meter (or the home owner) to pay for the corresponding electricity consumption or to acquire new services in exchange for detailed consumption reports sent to the UP. To ensure that tokens are untraceable and to protect the privacy of users, they must not be associated with a particular user and can only contain public information (e.g., an expiration day, the value of the token). As we show later, these reward tokens will be blindly signed by the UP in order to ensure that the UP cannot insert *identifiers* to track users while signing a given token.

The advantages of our proposal are manyfold:

- There is no need for trusted-third-parties to issue the tokens or vouch for the eligibility of transactions. Meters interact with the utility provider only when they wish to benefit from the rewards provided.
- The system does not require users nor the provider to setup specific accounts in order to collect the rewards (e.g., like in Bitcoin).
- Double-spending of the same token would result in immediate rejection of the token, without necessarily leaking information about the corresponding user.
- Two tokens with the same public information (e.g., same expiration date) are indistinguishable from each other.

In our solution, each reward token consists of two parts; a *distinguisher* D that is introduced by the meter and helps UP identify the token (not the meter) in case of double-spending, and a public part $P = \langle Val, Exp \rangle$ containing the value of the token and an expiration date. The value of the token should not be used to identify/trace the token, thus it must be chosen from a *coarse* set of predefined values known in advance to all meters in the grid.

[1] Recall that we assume here that the communication channel does not leak information about the identify of the meters.

A token must also be used within its expiration date (which must also be coarsely defined to prevent tracking). The expiration date helps the utility provider in maintaining a short database of used tokens. As tokens need to be stored in order to check against double spending, tokens can be deleted from the database once they expire.

4.2 Token Construction

For each token, the distinguisher D is an identifying piece of information that helps deter double-spending. Clearly, D could then be potentially abused by the utility provider to track the token and profile the meter/owner. This is exactly why the distinguisher must be blinded before being integrated into the token. We construct D by leveraging the identification scheme of Schnorr [7] and borrowing concepts from Brands' protocol [8] in order to detect double-spending of the same token.

In our solution, the meter uses two large primes $\bar{p}$ and $\bar{q}$, such that $\bar{q}$ divides $\bar{p}-1$ and g is a generator of order $\bar{q}$ in the group $Z^*_{\bar{p}}$. The meter S then selects two secret values $s, r \in Z_{\bar{p}}$, computes $\alpha = g^{-s} \bmod \bar{p}$ and $\beta = g^r \bmod \bar{p}$. The distinguisher D will consist of the two values α and β which will be used *(i)* by the meter to prove proper construction of the token during redemption using appropriate zero knowledge proofs, and *(ii)* protect the UP from double-spending attempts.

As the token C must also contain a public part $P = \langle Val, Exp \rangle$, this part must be tightly coupled with D since otherwise an attacker can use two tokens to construct a third one by interchanging their D/P values. However, as D will be blinded before signing, the UP cannot be sure that P was correctly incorporated into D.

One solution to this problem is for the UP to sign with a different key for every P pair. For example, if UP wishes to sign a token that will be valid until the end of the month, it can use a public-private key that will not be valid beyond the end of the month. This suggests that not only the UP must have a different key for every $\langle Val, Exp \rangle$ combination, but also that everybody must be aware of these keys and their validity. Clearly, this seriously affects availability and performance.

A more elegant solution is to use the concept of a *partially* blind signature [9]. A partially blind signature allows the signer (the UP in this case) to include public information in the blinded token under some "agreement" with the meter. This agreement simply consists of the value of the token which reflects the value of the data submitted by the meter—this can be known in advance under some predefined policy—and the expiration date of the token.

The basic idea is to make the public information P *part* of the public key of the UP. This can be done by having a public function G that transforms an arbitrary string P to a public key K^P_{UP} whose private key is only known to UP and can be used to sign the token. Since the resulting signature is bound to key K^P_{UP}, the public part of the token is also bound to the signature. By doing so, the two parts D and P of the token are now tied to each other.

More specifically, let $N = pq$, be the product of two safe primes, (e, N) the public key of the utility provider and (d, N) its corresponding RSA private key. Let also $G : \{0,1\}^* \rightarrow Z^*_p$, be a function transforming an arbitrary string to an element of Z^*_p. To embed the public part $P = \langle Val, Exp \rangle$ into a new token signing key, we proceed as follows:

1) Let $e_P = G(e, \langle Val, Exp \rangle)$. This generates a new public key from the public key of the provider and the P info of the token.
2) Set $d_P = 1/e_p \bmod \phi(N)$. This is going to be the new signing key for the token with public info P.

4.3 Token Redemption

Recall that, in our scheme, the user can redeem his tokens, e.g., to acquire a monetary reward, or to pay its electricity bill.

The construction of α and β allows the meter to prove in zero knowledge the validity of the token when it tries to redeem it—without leaking any information about the detailed energy consumption submitted previously to the UP. In particular, let C be the complete token and $D = \langle \alpha, \beta \rangle$ its distinguisher which is contained in C. When the meter/owner wants to redeem the token, it sends the tuple:

$$\langle C, y, date/time \rangle, \text{ where } \begin{array}{l} y = r + h \cdot s \mod \bar{q} \text{ and} \\ h = H(C, date/time). \end{array} \tag{1}$$

4.4 Token Verification

Upon reception of a submitted token, UP first checks its expiry date. If the token has not expired yet, UP then verifies its signature on the token and checks that:

$$\beta = g^y \alpha^h \mod \bar{p}. \tag{2}$$

If these verifications succeed, UP considers the token valid, but it further needs to be assured that it has not been redeemed before. For that purpose, it searches its database for a token with the same distinguisher D. If no match is found, the token is considered fresh and UP records the values $\langle C, y, date/time \rangle$. If a match is found, UP can recover the secret values r, s that were used to construct D—thus providing evidence of double-spending. More specifically, if the token has been used before, there will be two instances $\langle C, y, date/time \rangle$ and $\langle C, y', date'/time' \rangle$ such that $\beta = g^y \alpha^h \bmod \bar{p}$, $\beta = g^{y'} \alpha^{h'} \bmod \bar{p}$, $y = r + hs$ and $y' = r + h's$. From these two last values, UP can obtain $s = (y - y')/(h - h')$ and subsequently r. Thus, two submissions of the same token will result in hard evidence that the token has already been redeemed. As these values are not connected with the ID of the meter S, they cannot be used in identifying S. Hence, the anonymity of the meter is maintained.

5 Privacy-Preserving Authorization and Rewarding Protocol

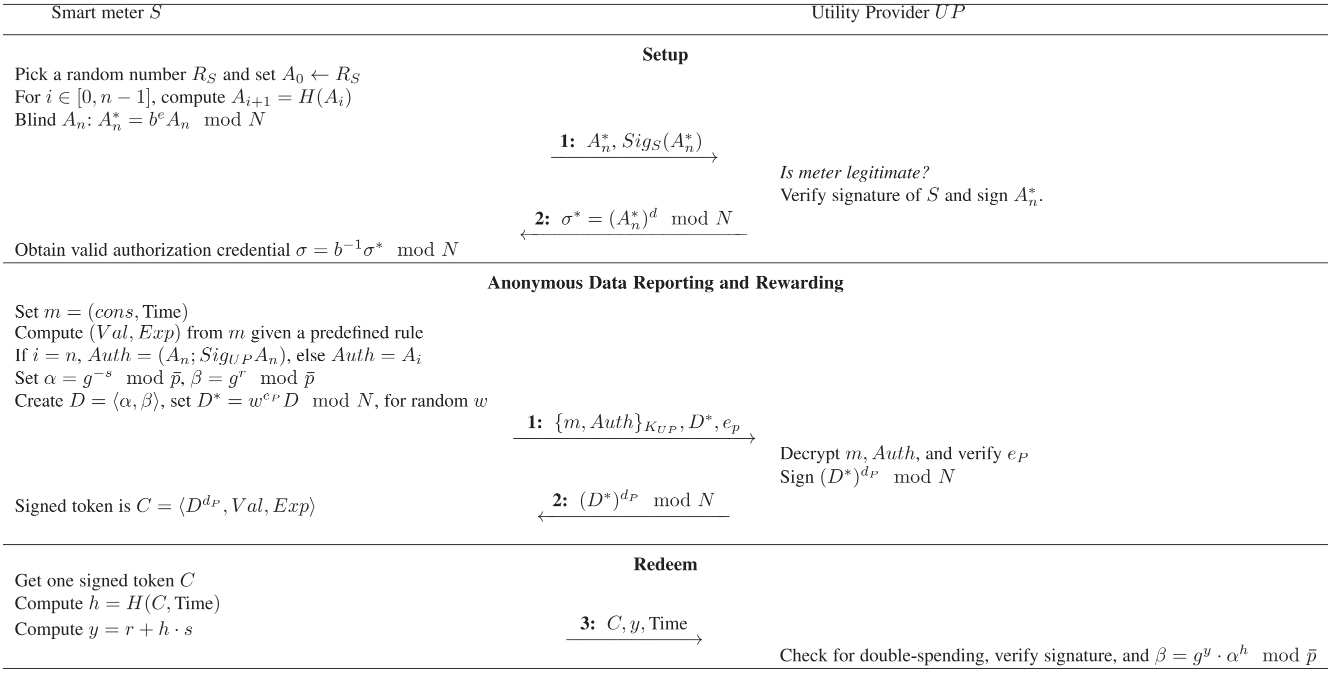
We now show how to combine the use of anonymous credentials and reward tokens in a unified protocol which captures our model outlined in Fig. 1. We then analyze the security and performance of our proposal.

5.1 Protocol Specification

Our protocol unfolds in Fig. 2. Here, we require the user to initially setup authorization credentials which will be used the first time the user reports its measurements. This setup phase has been detailed in Section 3.1.

These credentials will be used when the meter reports its detailed data consumption for the first time. Subsequent credentials (denoted by $Auth$ in the figure) are tied to the first one by making use of the hash-chain mechanism. Should the user wish to be rewarded for the data provided, it constructs the token by first computing its distinguisher and relevant data (value, expiration) based on some predefined policy.[2] It then proceeds to blind the distinguishing part of the token by sending $D^* = w^{e_P} D \bmod N$ to the utility provider (Step 1), where w is some newly selected random number. UP checks *(i)* if the meter is legitimate, and *(ii)* if the per token public key e_P has been computed correctly, and then returns

[2] This is possible since Val can be computed by the meter according to some policy agreed in advance while the expiration date can be set to a coarsely defined value (e.g., the end of the current year).



Smart meter S | Utility Provider UP

Setup

Pick a random number R_S and set $A_0 \leftarrow R_S$
For $i \in [0, n-1]$, compute $A_{i+1} = H(A_i)$
Blind A_n: $A_n^* = b^e A_n \mod N$

$\xrightarrow{\textbf{1: } A_n^*, Sig_S(A_n^*)}$

Is meter legitimate?
Verify signature of S and sign A_n^*.

$\xleftarrow{\textbf{2: } \sigma^* = (A_n^*)^d \mod N}$

Obtain valid authorization credential $\sigma = b^{-1}\sigma^* \mod N$

Anonymous Data Reporting and Rewarding

Set $m = (cons, \text{Time})$
Compute (Val, Exp) from m given a predefined rule
If $i = n$, $Auth = (A_n; Sig_{UP}A_n)$, else $Auth = A_i$
Set $\alpha = g^{-s} \mod \bar{p}$, $\beta = g^r \mod \bar{p}$
Create $D = \langle \alpha, \beta \rangle$, set $D^* = w^{e_P} D \mod N$, for random w

$\xrightarrow{\textbf{1: } \{m, Auth\}_{K_{UP}}, D^*, e_p}$

Decrypt $m, Auth$, and verify e_P
Sign $(D^*)^{d_P} \mod N$

$\xleftarrow{\textbf{2: } (D^*)^{d_P} \mod N}$

Signed token is $C = \langle D^{d_P}, Val, Exp \rangle$

Redeem

Get one signed token C
Compute $h = H(C, \text{Time})$
Compute $y = r + h \cdot s$

$\xrightarrow{\textbf{3: } C, y, \text{Time}}$

Check for double-spending, verify signature, and $\beta = g^y \cdot \alpha^h \mod \bar{p}$

Fig. 2. Sketch of the combined anonymous reporting and rewarding protocol. Here, $\{X\}_{K_{UP}}$ refers to the encryption using the public key of the UP.

the signature $(D^*)^{d_P} \bmod N$ (Step 2). The meter recovers the signature on D by dividing with w—in which case the signed token is ready and has the form:

$$C = \langle D^{d_P}, Val, Exp \rangle.$$

To redeem the token (Step 3), the meter supplies the required non-interactive zero-knowledge proof (Equation (1)) for the structure of the distinguisher. The UP then goes on to check the validity of the token and reward the user.

5.2 Security Analysis

In what follows, we analyze the security provisions of the protocol depicted in Fig. 2.

Authentication/Authorization. By relying on our anonymous authorization mechanism, the source of the data reports is appropriately authenticated due to the reliance on the blinded hash chain at the registration phase. On the other hand, the meter knows it is interacting with the provider, since all computations and messages are tied to the provider's credentials. This is true for both the authorization protocol, the data submission protocol, and the rewarding protocol where the signing of the distinguisher D is done with a key derived from the original public key of UP. Moreover, our protocols ensure the integrity and confidentiality of all exchanged messages at all times.

Unforgeability/Unreusability. The ZK proofs used while redeeming a reward ensure that only a meter who knows the genuine representation of α and β in the distinguisher D can supply these proofs [7]; consequently only the meter that created the token can redeem it. Moreover, if the meter/owner supplies two such proofs for the same reward, the secret values r, s used in the construction of both α and β can be extracted thus offering an undeniable proof of double-spending.

Untraceability/Unlinkability. When a user tries to redeem a reward and gives the UP the ZK tuple (Eq. (1)), the UP cannot tell which meter created the token as the only visible part during the token construction is the public part P of the token; as we discuss later, there are also other *side channels* that can be used to infer this information.

To show that two tokens with the same public part are unlinkable, we use the indistinguishability experiment shown in Fig. 3. In this setting, meters belonging to a smart grid system S interact with an adversarial utility provider $\mathcal{A}$.

In particular, two meters $\mathcal{M}_0$, $\mathcal{M}_1$ are involved in the following game with $\mathcal{A}$. First the meters create two tokens $\mathcal{C}_0$ and $\mathcal{C}_1$, respectively, following the protocol outlined in Fig. 2 and bearing the *same* public part $P = \langle Val, Exp \rangle$. After unblinding and retrieving the corresponding distinguishers $\mathcal{D}_0$ and $\mathcal{D}_1$, the meters give the tokens to a challenger who then proceeds as follows. It gives $\mathcal{A}$ the token $\mathcal{C}_a$, where a is a bit chosen uniformly at random. $\mathcal{A}$ outputs a guess bit a' and wins the game if $a = a'$.

If the adversary cannot distinguish the two cases with probability significantly more than random guessing, we say that the scheme provides unlinkability (cf. Definition 1). In our case, privacy is a consequence of the blindness property; $\mathcal{UP}$ signs the blinded distinguishers $\mathcal{D}_0^*$ and $\mathcal{D}_1^*$ but is given access to $\mathcal{D}_0$ and $\mathcal{D}_1$. As the blindness of these values ensures that the two tokens are indistinguishable from each other, $\mathcal{UP}$ will not be able to guess any of the two tokens with more than negligible advantage [6].

Experiment $\mathbf{Exp}_{\mathcal{A},\mathcal{S}}^{ind}$:

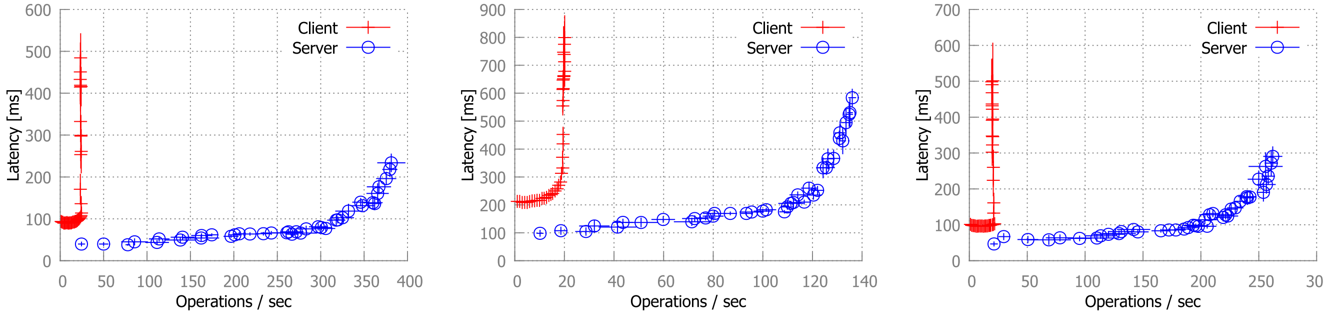
Setup phase: A set S of smart meters is initialized in order to participate in the data submission and token creation processes with an adversarial provider $\mathcal{A}$.

Challenge phase:

- $\mathcal{A}$ selects two meters $\mathcal{M}_0$ and $\mathcal{M}_1$ and sets the value and expiration dates $P = \langle Val, Exp \rangle$. It then engages in the token creation protocol with each meter until the *blinded* distinguishers $\mathcal{D}_i^*$ are received. Each meter then creates a token $\mathcal{C}_i$, and $\mathcal{A}$ is given access to a smart meter oracle.
- $a \overset{R}{\leftarrow} \{0, 1\}$. The oracle gives $\mathcal{A}$ the token $\mathcal{C}_a = \langle \mathcal{D}_a^{d_P}, Val, Exp \rangle$.
- $\mathcal{A}$ outputs a guess bit a'.

Exp is successful if $a = a'$.

Fig. 3. Token unlinkability experiment.



(a) Setup phase.

(b) Data reporting and rewarding.

(c) Token redemption.

Fig. 4. Performance evaluation of our proposals. Each data point in our plots is averaged over 10 independent runs; where appropriate, we also show the corresponding 95 percent confidence intervals.

Eliminating side channels. However, while the tokens themselves leak no information, other side channels exist that can be used to break meter privacy. Consider, for example, the case where the IP address of a meter is visible when the meter submits consumption data/retrieves a reward to/from the *UP* and then tries to spend this reward. Obviously, in such a case additional mechanisms are required to ensure that a network connection remain anonymous. One such mechanism, as we mentioned in Section 2, is the anonymity network TOR [22].

Another side channel stems from the structure of the token's public part $P = \langle Val, Exp \rangle$. If the value of the token or its expiration date is unusual, both can be used to associate the data with the meter upon redeeming the token. Hence, these values must be drawn from a distribution that does not allow for such discrimination. For example, expiration dates can be set to the end of the current year and token values can be coarsely defined.

Finally, while the *UP* can tie two submissions to the *same* hash chain, the use of blind signatures ensures that no information can be tied to a particular user. While there is no direct breach of privacy here, we can eliminate hash chains entirely as follows. The user obtains the first authorization credential σ_0 in the usual manner during setup. Subsequent credentials can be obtained *on the fly* during data submission. Therefore, once the *UP* validates its signature on σ_0, it proceeds to blindly sign the next credential σ_1, and so on. As these credentials do not bear any identification data, the *UP* cannot relate the two tokens—thus effectively providing unlinkability between different reporting sessions. However, this approach is more computationally expensive than our hash chain solution (since each token has to be blindly signed), and does not allow the *UP* to force re-registration of SMs, when needed (e.g., to revoke malicious SMs).

5.3 Performance Evaluation

We implemented a prototype of our proposals in Scala. We relied on SHA256, the Java built-in secure random number generator, and we set $|N| = 2{,}048$, and $|q| = 224$ bits.

We deployed our implementations on a network consisting of two 24-core Intel Xeon E5-2640 with 32 GB of RAM; the communication between various machines was bridged using a 100 Mbps switch. To emulate a realistic Wide Area Network (WAN) in our network, we relied on NetEm [10]. For that purpose, we add a Pareto distribution with a mean of 20 ms and a variance of 4 ms, to shape all traffic exchanged on the networking interfaces and emulate the packet delay variance specific to WANs [11]. The *UP* was running on one of the 24-core Xeon E5-2640 machine, whereas the smart meters were co-located on the second 24-core Xeon E5-2640 machine.

When implementing our scheme, we spawned multiple threads on the *UP* machine, each thread corresponding to a unique worker performed to serve a given smart meter request.

We evaluate the response time of the *UP* with respect to the achieved throughput in the different stages of the protocol in Fig. 3. We measure throughout as follows: we require that each smart meter performs back to back requests for acquiring authorization tokens,[3] and/or for submitting detailed measurements; we then increase the number of meters in the system until the aggregated throughput attained by all meters is saturated. The throughput is then computed as the aggregated number of requests that can be concurrently handled by the *UP* per second. We also measure the end-to-end latency witnessed by the SMs with respect to the number of concurrent operations that they issue.

Communication efficiency. Our evaluation results are depicted in Fig. 4. Our findings show that our proposals scale well with the number of SMs in the grid. Namely, the *UP* can handle up to 380 concurrent SM requests/sec for authorization, with a response time of 220 ms for each request. Moreover, the *UP* can handle 135 reporting and rewarding requests/sec with a response time of 590 ms for each request, and almost 265 concurrent token redemption requests (with a response time of 290 ms per request).

Our results show that the processes of token generation, data reporting, and reward redemption can be easily tolerated by the SMs in realistic deployment settings. That is, the end-to-end latency witnessed by SMs when acquiring authorization tokens and redeeming rewards is at most 100 ms—which is largely dominated by the communication latency. Recall that in these cases, the response time of the *UP* is at most 50 ms under modest computational load. Our results suggest that the data reporting and rewarding phase consumes almost double the latency of the remaining two phases; in this case, the end-to-end latency witnessed by SMs is approximately 200 ms per operation. Recall that, in the data reporting and rewarding phase, SMs need to compute a number of exponentiations to construct and blind D, and should also verify the signature issued by the *UP*.

Given our multi-threaded implementation, SMs can also batch their authorization and reward redemption requests without deteriorating the witnessed end-to-end latency. Here, as the number of concurrent requests increases, the threads in our thread pool are exhausted and the system saturates—which explains the sharp increase in the latency measured by SMs who issue more than, e.g., 25 concurrent requests per second.

Computation efficiency. During setup, the meter has to compute a blinded authorization token. As shown in Fig. 4a, this operation requires few tenths of milliseconds. During the data reporting and rewarding phase (cf. Fig. 2), the most expensive operation is the actual transmission of the data; token creation requires two modular exponentiations for α and β, and two modular multiplications for the blinding and unblinding of D. On the other hand, during redemption, each meter needs to prove knowledge of α and β;

[3] In our implementation, we used hash chains of length 10.

however this requires only one extra addition and multiplication to compute y.

Storage efficiency. We note that, although the UP has to keep track of the spent tokens in order to detect any double-spending, the UP has only to store those tokens which have not yet expired (as indicated in their public part). Notice that storing 564 bytes per unexpired token (i.e., 264 bytes to store C, and 288 bytes to store y) is sufficient to detect double-spending. This clearly suggests that the storage overhead incurred by our proposals can be easily tolerated, e.g., when compared to the storage overhead required to store the data reports.

6 Related Work

To the best of our knowledge, this is the first contribution which proposes the use of anonymous reporting/rewarding mechanisms in the smart grid.

Our anonymous authorization problem resembles the problem of *anonymous subscriptions* [12], where a client sets up a secret to login to a service, without, however, allowing the server to distinguish which user logged in nor link this session with a user's past logins. This was later improved in [13] who presented a more practical subscription scheme. Recently, Lee et al. [14] presented a system that confines the search of whether a credential has been reused within a predefined epoch, thus ignoring all past sessions. This can be used as a complement to our scheme, to speed up the detection of token double-spending attempts.

Efthymiou and Kalogridis [15] propose a protocol that allows a meter to send anonymized energy consumption data to the utility provider. The protocol assumes the existence of a trusted third (escrow) party which must first be used to authenticate the data as coming from a legitimate meter. Molina-Markham et al. [16] show how auxiliary information such as one coming from social networking exposure can help in building analytic tools for profiling users. Although they suggest the use of Zero-Knowledge proofs for submission and billing of meter data, they did not provide any explicit construct that can be adapted to the smart grid environment.

Kursawe et al. [17] propose a number of protocols for private aggregation of energy measurements. These protocols enable the exact calculation of sum aggregates through appropriate masking of the original data, or comparison with a known value using appropriate homomorphic operators. Dimitriou [18] further extended these protocols to support privacy-friendly aggregation in the presence of honest-but-curious and malicious adversaries; the proposed schemes only require $O(1)$ work per meter under certain conditions.

Li et al. [23] propose to aggregate smart meter readings along tree paths using homomorphic encryption. Rottondi et al. [24] present an architecture in which a central controller node defines information flows between producers and consumers in the grid based on appropriate secret shares distributed by the producers of energy data. In a similar manner, Lu et al. [25] presented a security architecture that aims to securely aggregate measurements received by users.

In [26], Dimitriou and Karame proposed a solution that enables the planning of energy distribution in the grid without leaking any information about the energy requests of individual smart meters. In [28], the authors propose rewarding users of the smart grid in order to incentivize them to *reduce* their consumption when there is shortage of electricity.

7 Conclusion

Within existing smart grids, smart meters participate in the collection of measurements and the reporting of data to the utility provider. While solutions exist that ensure meter privacy with regards to these operations, these solutions do not allow the provider to acquire detailed energy measurements which are essential for accurately maintaining and configuring the smart grid.

In this work, we proposed an authorization scheme that allows a smart meter to anonymously interact with the utility provider when submitting detailed consumption data. Additionally, our solution enables the incorporation of anonymous rewarding mechanisms in the smart grid in exchange for detailed measurements that users report. We analyzed the security of our proposal and evaluated its performance by means of implementation in a realistic deployment setting. Our results suggest that our scheme incurs tolerable performance overhead on both the utility provider and the users, and scales with the number of users in the grid. We therefore hope that our findings motivate further research in this area.

Note that our scheme relies on the existence of an anonymizing network to prevent the identification of the message source. In terms of future work, it would be interesting to investigate the feasibility and realization of anonymizing networks within smart grids.

Acknowledgments

The authors would like to thank Ludovic Barman for the help in implementing and evaluating our protocols. The research is supported by Kuwait University, Research grant no. QE02/13. T. Dimitriou is the corresponding author.

References

[1] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security Privacy Mag.*, vol. 8, no. 1, pp. 11–20, Jan./Feb. 2010.

[2] E. L. Quinn, "Privacy and the new energy infrastructure," *Center for Energy and Environmental Security (CEES)*, Working Paper No. 09-001, 2009.

[3] NIST, "Guidelines for smart grid cyber security," Privacy and the Smart Grid, *NISTIR 7628*, vol. 2, Aug. 2010.

[4] T. Jeske, "Privacy-preserving smart metering without a trusted-third party," in *Proc. Int. Conf. Security Cryptography*, 2011, pp. 114–123.

[5] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Whitepaper, 2009, https://bitcoin.org/bitcoin.pdf

[6] D. Chaum, "Blind signature system," in *Proc. CRYPTO: Adv. Cryptol.*, 1984, p. 153.

[7] C. P. Schnorr, "Efficient signature generation by smart cards," *J. Cryptol.*, vo1. 4, no. 3, pp. 161–174, 1991.

[8] S. Brands, "Untraceable off-line cash in wallets with observers," in *Proc. 13th Annu. Int. Cryptol. Conf. Adv. Cryptol.*, 1993, pp. 302–318.

[9] M. Abe and E. Fujisaki, "How to date blind signatures," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security: Adv. Cryptol.*, 1996, pp. 244–251.

[10] NetEm. NetEm, the Linux Foundation. Website. (2009) [Online]. Available: at http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

[11] D. Dobre, G. Karame, W. Li, M. Majuntke, N. Suri, and M. Vukolic, "PoWerStore: Proofs of writing for efficient and robust storage," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2013, pp. 285-298.

[12] I. Damgard, K. Dupont, and M. O. Pedersen, "Unclonable group identification," in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptographic Techn. Adv. Cryptol.*, 2006, pp. 555–572.

[13] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich, "How to win the clone wars: Efficient periodic n-times anonymous authentication," in *Proc. ACM 13th Conf. Comput. Commun. Security*, 2006, pp. 201–210.

[14] M. Z. Lee, A. M. Dunn, B. Waters, E. Witchel, and J. Katz, "Anon-Pass: Practical anonymous subscriptions," in *Proc. IEEE Symp. Security Privacy*, 2013, pp. 319–333.

[15] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, 2010, pp. 238-243.

[16] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proc. 2nd ACM Workshop Embedded Sens. Syst. Energy-Efficiency Buildings*, 2010, pp. 61–66.

[17] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Proc. 11th Int. Conf. Privacy Enhancing Technol.*, 2011, pp. 175–191.

[18] T. Dimitriou, "Secure and scalable aggregation in the smart grid," in *Proc. 6th IFIP/IEEE Int. Conf. New Technol.,Mobility Security*, 2014, pp. 1–5.

[19] T. Dimitriou and G. Karame, "Privacy-friendly tasking and trading of energy in smart grids," in *Proc. 28th Symp. Appl. Comput.*, 2013, pp. 652–659.

[20] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weil pairing,” *J. Cryptol.*, vol. 17, no. 4, pp. 297-319, 2004.

[21] A. Rial and G. Danezis, “Privacy-preserving smart metering,” in *Proc. ACM 10th Annu. Workshop Privacy Electron. Soc.*, 2011, pp. 49–60.

[22] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The Second-generation onion router,” in *Proc. 13th Conf. USENIX Security Symp.*, 2004, pp. 21–38.

[23] F. Li, B. Luo, and P. Liu, “Secure information aggregation for smart grids using homomorphic encryption,” in *Proc. IEEE 1st Int. Conf. Smart Grid Commun.*, 2010, pp. 327—332.

[24] C. Rottondi, G. Verticale, and A. Capone, “A security framework for smart metering with multiple data consumers,” in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2012, pp. 103–108.

[25] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, “EPPA: An efficient and Privacy-preserving aggregation scheme for secure smart grid communications,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[26] T. Dimitriou and G. Karame, “Privacy-friendly planning of energy distribution in smart grids,” in *Proc. 2nd Smart Energy Security Workshop*, 2014, pp. 1–6.

[27] G. Karame, E. Androulaki, and S. Capkun, “Double-spending fast payments in bitcoin,” in *Proc. ACM Conf. Comput. Commun. Security*, 2012, pp. 906–917.

[28] S. Jain, B. Narayanaswamy, and Y. Narahari, “A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 721–727.