Privacy-Friendly Tasking and Trading of Energy in Smart Grids

Tassos Dimitriou
Athens Information Technology, Greece,
Kuwait University, Kuwait
tassos.dimitriou@ieee.org

Ghassan Karame
NEC Laboratories Europe
69115 Heidelberg, Germany
ghassan.karame@neclab.eu

ABSTRACT

The smart-grid is gaining increasing attention nowadays, owing to its premise to offer increased reliability and performance. However, the current design of smart-grids raises serious concerns with respect to the privacy and anonymity of users.

In this paper, we address the problem of enhancing the privacy of users in the smart grid throughout the reporting and the billing phases. To that end, we propose a taxonomy of solutions that enable *(i)* the privacy-preserving aggregation of smart meters' energy consumption reports without relying on a single point of trust, *(ii)* the anonymous tasking, e.g., the outsourcing of (maintenance) tasks to smart meters, and *(iii)* the privacy-preserving billing and barter of energy between the utility provider and the smart meters. To the best of our knowledge, this is the first contribution that comprehensively addresses privacy issues in the tasking and energy-trading processes in smart grids. Our proposed solutions complement previous work in the area and can be easily integrated within existing smart grids.

1. INTRODUCTION

The electrical grid is currently undergoing a major transformation with the introduction of infrastructural support for a "smarter" grid. The new grid, the *smart grid*, leverages this support to achieve fine-grained power consumption monitoring, and integrate appliances and new sources of renewable energy in an attempt to offer higher efficiency, reliability and security [21].

The smart grid uses various information and communication technologies to provide better "contextual awareness" regarding the state of the grid. Using such intelligent communications and appropriate consumption data, load shedding can be implemented and both users and utility providers can benefit from a balanced utilization of energy to meet the various customers demands. Core to such technology is the use of *smart meters*, i.e. devices that record and communicate consumption of electric energy to the central system for monitoring and billing purposes. Smart meters thus support both dynamic pricing and a two-way flow of electricity between homes and the grid.

However, the widespread deployment of smart meters introduces serious privacy risks since the frequent collection of power data may reveal considerable information about residential appliance usage. In fact, previous studies [1, 2] have shown that energy signatures of home appliances can be used to remotely eavesdrop at activities within homes, thus exposing a wealth of private information to anyone with access to such usage data. Furthermore, even when not all appliances can be identified within a person's electricity profile, the surrounding context and the use of statistical tools along with information that is willingly shared in the Internet can be used to intrude at the life of individuals [2]. For instance, typical questions that can be answered by data analytics include "How many hours did Alice sleep last night?", "When was Bob out of home?" [3, 4], etc.

Currently, only a handful of solutions exist to protect smart grid privacy; most of these solutions rely on the use of anonymization/escrow [5] or aggregation techniques [6] so that clients' information can be aggregated and encrypted. Other solutions require users to prove in Zero Knowledge the correctness of computations based on readings on their own devices [7] or rely on statistical tools to minimize the risks of information leakage while retaining the benefits of the transmitted information (see [8] and the references therein). As such, most of these contributions rely on the existence of a third party that can be fully trusted for the aggregation of the reports and do not address privacy implications that can arise from other operations that are envisioned by the smart grid such as billing, payments, maintenance, etc.

Indeed, the premise behind smart grids goes beyond the simple collection of measurements as smart grids enable the utility provider to update the pricing function or to send new measurement tasks to smart meters; these tasks could correspond to immediate monitoring tasks due to an outage or could support maintenance activities, etc. Furthermore, the smart grid was designed to support the "smart" integration of the (surplus) energy originating from home owners within the smart grid; home owners can produce energy (e.g., from solar power) and sell their surplus back to the utility provider. This is specifically important for small rural areas that are remote from the utility provider. By gathering the surplus of energy from end-users of the grid back to other users of the smart grid, the provider minimizes energy distribution costs and increases the utility of the grid. Clearly, these use-cases might incur privacy threats that can-

SAC'13 March 18-22, 2013, Coimbra, Portugal.

not be addressed solely by ensuring the private aggregation of reports.

In this work, we present a taxonomy of solutions to enhance the privacy of smart grids subject to the aforementioned use-cases. For that purpose, we explore the solution space to ensure a privacy-preserving aggregation of measurement reports without relying on a single trusted entity. We then go one step further and we discuss the privacy implications of executing additional tasks (e.g., for maintenance reasons) that might be requested from the utility provider. We also discuss possible privacy-preserving incentive mechanisms to encourage home owners to provide tasking results. Finally, we propose a solution that relies on the generation of anonymous "energy tokens" in order to enable a secure barter of energy between users and utility providers. As a by-product, our proposed solutions, combined, suggest a new framework for smart grids that does not only respect the privacy of users when collecting measurement reports from the smart meters, but goes one step further beyond existing techniques to support other possible operations that the smart meter can be involved in. As far as we are aware, this is the first contribution that comprehensively addresses privacy issues in the aggregation, tasking, and billing processes within smart grids.

The remainder of this paper is organized as follows. In Section 2, we outline our model and we explore the solution space for enhancing the privacy of users given the (tasking) measurements collected by smart meters. In Section 3, we propose the reliance on anonymous energy tokens to enable privacy-preserving trade of energy. In Section 4, we overview related work in the area and we conclude the paper in Section 5.

2. PRIVACY-PRESERVING AGGREGATION OF REPORTS IN SMART GRIDS

In this section, we introduce our model and threat assumptions and we explore a number of solutions that achieve the privacy-preserving aggregation of (tasking) measurements generated by various SMs.

2.1 Model

Our model consists of Smart Meters (SM) and a Utility Provider (UP). Smart meters are enhanced metering devices that can be used to measure the consumption of electricity and communicate with other parties such as the utility provider. The SM is installed within a home and can interact with a home area network. We assume that the SM has access to the Internet, at least intermittently, through some open-access Wi-Fi infrastructure. We also assume that SMs feature secure storage and autonomous cryptographic functionality. This can be achieved, for example, by using tamper-evident meters or TPM chips (see [7] for a similar assumption). Typically, the UP interacts with individual SMs by exchanging price information, meter data and control commands. As we show later (cf. Section 2.2), this can be performed through a Report Server (RS), which is trusted by the SMs to correctly aggregate and de-anonymize the reports.

In this work, we envision a setting in which the UP does not only regularly receive measurements from the associated metering devices, but can *task* certain SMs to report relevant contextual information, or can also *purchase* energy from a given SM, etc. Clearly, this comes at the expense of privacy leaks with respect to sensitive user data [1, 2]; users (or their corresponding SMs)—and rightly so—should maintain control of the release of their sensitive measurements throughout their participation in the smart grid. This includes the protection of information that can be inferred from the readings themselves as well as from the interaction of the users with the various smart grid system components. In this respect, the UP or an external eavesdropper should not be able to determine information about the clients (e.g., profiling the client's consumption of electricity), even when their SMs are executing specific tasks requested from the UP or when trading energy with the UP.

Algorithm 1: Report Submission using the RS

```
   // SM on reporting consumption values
1  Ts = timestamp()
2  σ = Sig_SM(SM, cons, Ts)
3  SM → RS : ⟨SM, cons, Ts, σ⟩
   // RS on handling measurements
4  if Verify(SM, cons, Ts, σ) = True then
5     data = Anonymize(SM, cons, Ts)
   // Aggregate data
6  report = report ∪ data
7  RS → UP : ⟨report⟩
```

2.2 Private Aggregation of SM Reports

In what follows, we explore possible ways that can be used to enable a private aggregation of the reports sent by SMs. The goal here is to compute the aggregated energy consumption of n consumers (different granularity levels may range from the typical neighborhood to the city level) and forward it to the UP without revealing any information about individual readings.

2.2.1 Reporting

During the billing period, the smart meter obtains consumption values *cons* and outputs measurements $m = (cons_j, timestamp_j)$, where j is a counter initialized at 0 that is incremented each time the SM outputs a new tuple. These measurements can be used to calculate a *fee* to be paid to the UP using a predefined billing policy. This policy is assumed to be known to the SM in advance but can be updated on demand through a request from the UP (cf. Section 2.3). As shown in [5,14], SMs typically send reports to the UP via the Report Server (RS). The RS can be seen as a trusted intermediary whose sole role is to strip away any information[1] that may serve as a quasi-identifier for the corresponding SM.

As such, at the end of the billing period, the SM transmits the consumption tuples to the RS. These tuples should be typically signed in order for the RS to verify the integrity of the data. A digital signature can be used if the SM trusts that the RS will not compromise its privacy. Otherwise, a *group signature* can be used instead. Group signatures [11] can be used to authenticate an SM or sign a report without revealing the identity of the signing SM. Here, we assume that at least k SMs hold the same group signature key; this ensures the k-anonymity of the signing SMs [10].

[1]For instance, special distinguishers could be introduced by the UP to track the activates of SMs.

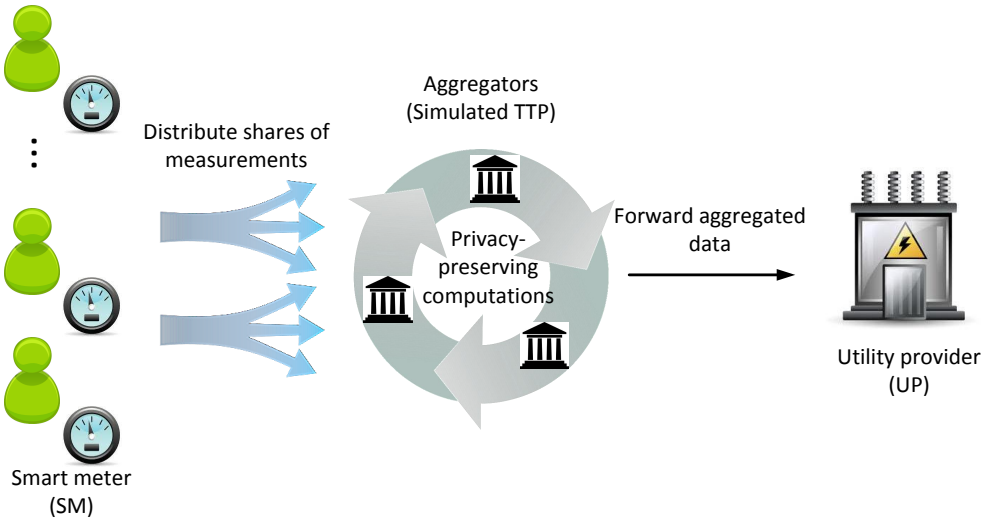

Figure 1: Distributing trust among a number of RSs using secret sharing. Secret sharing ensures that no single RS can learn the values of the data that is being aggregated. Note also that the role of the RSs can also be assumed by SMs themselves.

The RS simply aggregates these values and forwards a collective summary to the utility provider. For example, after collecting the monthly reports of n smart meters, the RS produces a report summarizing the total consumption incurred by all the SMs during the same period. This report masks the individual consumption values, provided that the number participating SMs is large enough to smooth out individual energy patterns. A high level description of this process is shown in Algorithm 1.

In what follows, we extend this solution and we propose a technique that enables to distribute trust among the aggregators.

2.2.2 *Distributing Trust among Aggregators*

One way to reduce the amount of trust placed on a single RS is to distribute the consumption values among two or more RS servers. Each time an SM must report its readings, it breaks them in disjoint subsets and forwards each one to a replica RS server (Figure 1). For example, the daily readings can be broken in intervals of e.g., one hour so that every replica gets a fraction of the original report. Although this approach may still leak information to a malicious RS, the disclosed readings may not allow a precise representation of the user's activities and profile.

Note that instead of sending plain but partial measurements to different RSs, each SM can use *secret sharing* ([12, 13]) to break its readings in multiple shares so that at least t shares are necessary to recover the original measurement. Each share is sent to a different RS, thus offering increased protection against collusion attempts; this solution would require the collusion of at least t RSs to obtain individual measurements. In addition, no individual readings or fractions of them are visible to the RSs. Upon reception of the various readings, the RSs can concurrently sum their shares obtained from either different SMs or from the same SM at different times. The summed shares are then forwarded to the UP who can recover the collective measurements. By virtue of the homomorphic properties of the secret sharing scheme, the recovered measurement corresponds to the aggregated sum of the individual readings. Thus, the UP obtains the total consumption values, but gets no information about individual measurements.

Remark (A Decentralized Variant)
We point out that the role of the RSs can be assumed by the SMs themselves. That is, a subset *of the SMs can organize themselves in a group (the* aggregators*) to emulate the role of the RSs. While SMs do not have to trust each other, we assume that the SMs are rational entities that aim at maximizing their advantage in the system. As such, we believe that it is realistic that SMs will agree on a set of most-trusted participants (as an extension, a mix of SMs and external entities can be used) for hosting the aggregators. These aggregators can be selected by means of a reputation management system [22, 23]. Incentives can also be used (for example credit can be given for providing this service) for SMs to agree to play the role of aggregators (cf. Section 2.3.2). We further assume that the aggregators, themselves, are rational and are interested in the correct outcome of the computation; it is highly likely therefore that they will comply with the protocol, at least in a* semi-honest *way (to execute the aforementioned secret-sharing scheme).*

Protocol for Sharing Measurements: In what follows, we propose a protocol to distribute trust amongst several RSs. Our protocol is based on the verifiable secret sharing protocol of [13]. Here, each SM generates a random polynomial $p()$ of degree $t-1$ over the field $\mathcal{Z}_q$, where q is a public parameter. The value $p(0)$ corresponds to a secret measurement which is recoverable only when t tuples $(i, p(i))$ are used by the RSs/UP to reconstruct the secret.

Let g, h be two generators of a group $\mathcal{G}_q$, of prime order q, such that computing discrete logarithms in this group is computationally hard. Furthermore, let x_i denote the private key of RS_i and $y_i = h^{x_i}$ its registered public key. We assume that y_i is known to all SMs given an initial

Algorithm 2: Distributing Trust among RSs

// SM on creating and distributing shares
Obtain public key y_i of RS_i
Set $\alpha_0 = \langle cons_j, timestamp_j \rangle$
Create random polynomial $p(x) = \alpha_0 + \sum_{j=1}^{t-1} \alpha_j x^j$
$SM \rightarrow RS_i : C_j = g^{\alpha_j}$ and $Y_i = y_i^{p(i)}$
// RS_i on recovering the share
$S_i = Y_i^{1/x_i}$ by decrypting with private key x_i
// t RSs working together to reconstruct the measurement
$\prod_{i=1}^{t} S_i^{\lambda_i} = h^{\alpha_0}$

Algorithm 3: Private Tasking in Smart Grids

// Utility Provider generates a new task
Ts = timestamp()
Task $t = \langle T_{id}, Ts, frequency, duration, ... \rangle$
$\sigma = Sig_{UP}(UP, t, Ts)$
// Register task with TS
$UP \rightarrow TS : \langle SM, cons, Ts, \sigma \rangle$
// TS on validating tasks
if Verify$(U, t, Ts, \sigma) = True$ **and** AcceptableTask$(t) = True$ **then**
  $T = T \cup t$
// Task download
$SM \rightarrow TS :$ Request for new tasks
$TS \rightarrow SM :$ Tasks $t_1, t_2, \ldots$

setup/boostrapping phase. Each SM picks a polynomial p of degree at most $t-1$ with coefficients α_j chosen at *random* from $\mathcal{Z}_q$, $p(x) = \sum_{j=0}^{t-1} \alpha_j x^j$, and sets α_0 equal to the secret measurement. Note that the SM keeps this polynomial secret but publishes the related commitments $C_j = g^{\alpha_j}$ and the encrypted shares $Y_i = y_i^{p(i)}$, using the public keys of the RSs. To reconstruct the secret value, each RS uses its private key x_i to obtain the share $S_i = h^{p(i)}$ by computing Y_i^{1/x_i}. It can also provide a non-interactive proof that S_i is a correct decryption of Y_i thus enhancing the public verifiability of the scheme (details omitted due to lack of space). Finally, the measurement can be constructed by t RSs working together using Lagrange interpolation:

$$\prod_{i=1}^{t} S_i^{\lambda_i} \;=\; \prod_{i=1}^{t} \left(h^{p(i)}\right)^{\lambda_i} \;=\; h^{\sum_{i=1}^{t} p(i)\lambda_i} = h^{p(0)},$$

where $\lambda_i = \prod_{j \neq i} j/(j-i)$ is a Lagrange coefficient. Note also that the homomorphic property of the scheme can be used to aggregate measurements not only for the same SM but also from different SMs. The RSs must collect all these encrypted shares (corresponding to the various SMs) and perform one collective reconstruction operation. This process is outlined in Algorithm 2.

2.3 Private "Tasking" in Smart Grids

As mentioned earlier, it is envisioned that in smart grids the UP requests smart meters to execute additional tasks, e.g., to perform additional/specific measurements for maintenance/monitoring purposes. In this section, we analyze the privacy implications of this typical use-case of smart grids and we outline a number of solutions that can ensure anonymous "tasking".

Example of tasks include requests to update the pricing function, to change the reporting frequency, the extent/type of measurements, etc. for all SMs that are located within a specific neighborhood/region. Note that this process might endanger the privacy of the participants in several ways. On one hand, if SMs were to contact the UP directly to fetch these tasks, then this may provide information to the UP about the location of the querying SMs. On the other hand, the nature of the tasks may also leak information about the SMs accepting the task. For example, when the task consists of monitoring energy consumption in a very small area containing just a couple of houses, this may be used to infer the location of the SM that is executing the task. We call this last threat *selective tasking* as its goal is to differentiate and identify anonymous participants by linking the SMs (that are fetching the tasks) with the received reports. As the number of tasked SMs is restricted, an adversary can easily link each device to the submitted report, even if anonymous reporting can be enforced using other techniques (e.g., TOR [9]).

2.3.1 Relying on a Task Server

To counter the aforementioned threats, we envision the existence of a Task Server, TS (Figure 2) whose role is to *(i) distribute tasks* that are outsourced by the UP, and *(ii)* to protect the anonymity of the users from the UP.

In fact, as shown in Algorithm 3, whenever the UP wishes to issue a new task t, it registers t with the TS. The TS then checks the validity of the signature but most importantly whether the task execution ensures the anonymity of the SMs (and the home owners) willing to execute the task. For example, the task server must ensure that the task is not an instance of selective tasking and that it can be executed by at least k SMs in the requested region (k-anonymity). If this last test fails, the task is dropped; alternatively, it can be flagged as privacy leaking and it can be left to the discretion of the SM whether it wants to execute it or not. Otherwise, task t is advertised by the TS for download. Once SMs have access to this list of tasks advertised by the TS, they can select the tasks $t_1, t_2, \ldots$ that they are willing to execute. The measurement results corresponding to the tasks (if any) are then sent to the RSs as shown in Figure 2.

2.3.2 Incentives for Tasking

To provide incentives for detailed tasking measurements, Algorithm 3 can be extended to support a privacy-preserving credit reward mechanism for users submitting enhanced reports. We show how this can be achieved using a *(i)* a central bank, and *(ii)* a decentralized digital payment system.

Using a Central Bank: In what follows, we assume the existence of a payment/banking service B that can reward the home owners for their contributions. Note that these payments should not be linkable to SMs contributing measurements or other payments rewarded to the same home owner.

One possible solution unfolds as follows. Let F be a one-way function and h a hash function. Given a collection of measurements m associated with a task t, the SM

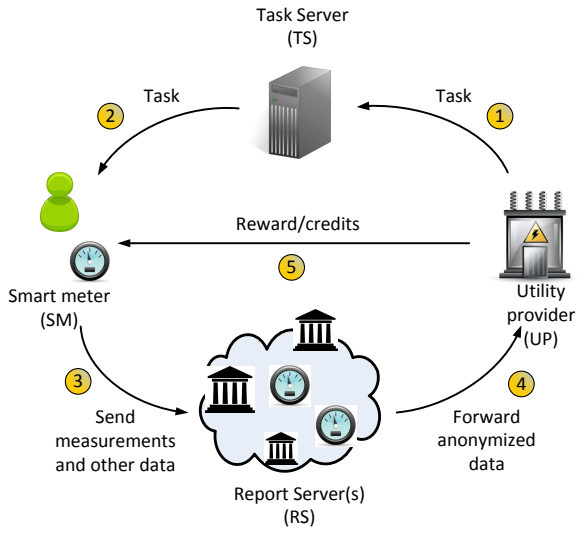

Figure 2: Smart grid model including a Task Server (TS). The (ordered) sequence of steps starting from the task submission to receiving credit is depicted.

(or the home owner) computes $s = F(h(m), t, N)$, where N is a fresh nonce. The SM appends s to the report sent to the trusted RS. Note that the signature s needs to be blinded from the UP; this can be achieved using the RS that anonymizes the report (strips the signature away) and forwards everything to the UP. Alternatively, group signatures can be used as we already mentioned in Section 2.2.1. In this case, an anonymization network such as TOR can be used to provide the desired unlinkability between the report sent to the RS and the IP address of the SM meter producing the measurements m.

Once the UP receives the report associated with task t (along with s), it produces a payment $\langle h(m), s, p_U \rangle$, where p_U is the payment for the measurements m, and forwards it to the payment service B. To redeem its contribution, the user sends to the bank an *anonymous* claim message of the form $\langle h(m), t, s, N \rangle$. The payment p_U is forwarded to the user once the bank verifies the validity of s. The above scheme protects the users' privacy as long as B and the utility provider do not collude (hence the use of anonymous channels) and the payment p_U is not connected to m or t that may help de-anonymize the user. If this is the case, coins can be constructed by the user corresponding to the amount of data reported and then blindly signed by the UP (cf. Section 3.2).

Using a Decentralized System – Bitcoin: Another possible way to issue (monetary) rewards in a decentralized manner (i.e., without relying on central banks) would be to rely on digital currencies, such as Bitcoin [19,24]. Bitcoin is a decentralized P2P payment system that was introduced in 2008.

In Bitcoin, peers transfer coins to each other by issuing a transaction. A transaction is formed by digitally signing a hash of the previous transaction where this coin was last spent along with the public key of the future owner and incorporating this signature in the coin [24].

As shown in [25], Bitcoin provides support for making deposits to third parties. In Bitcoin, transactions can be associated with a "lock time"; this allows transactions to remain pending until the lock time is exceeded. During this time, the transaction can be replaceable if all parties can reach such an agreement. We leverage this mechanism of Bitcoin to reward users of the smart grid in exchange of tasking as follows. Here, the UP can use Bitcoin to "commit" to a given user U that U will eventually receive a (monetary) reward if U correctly executes the required tasks. This can be achieved by issuing a transaction with a large lock time (e.g., one year) that requires the signatures of both the UP and the user U; in Section 3, we show how such signatures can be blinded. As such, provided that U correctly executes the tasks outsourced by the UP, U can be sure that it will receive a reward from the UP. That is, if the UP does not offer a reward in the form of BTCs to U in exchange for the executed tasks, then the deposit made by the UP cannot be later redeemed by the UP (i.e., U will block the deposit transaction). We point out that Bitcoin ensures that *(i)* all transactions/deposits can be publicly verifiable by all entities and *(ii)* transactions cannot be double-spent/withdrawn once they are included in the Bitcoin block chain.

In Section 3.2, we describe another payment mechanism that relies on tokens of energy.

3. ANONYMOUS ENERGY TOKENS

In the previous section, we showed how to distribute trust among a number of entities to ensure a privacy-preserving aggregation or smart meter reports. We point out that this is not a sufficient solution, alone, to ensure the privacy of users, as their anonymity can be compromised when users are paying the utility provider in exchange for their energy consumption. In this case, both the cumulative energy measurements of users and the anonymity of users is revealed.

In this section, we show how to construct anonymous and secure tokens to trade energy between users and the UP. These tokens do not require functionality from external entities (such as Bitcoin) and can be generated by the UP.

3.1 Tokens of Energy

To enable an anonymous trade of energy between users and the UP, we propose the reliance on anonymous *tokens of energy*. A token essentially corresponds to a prepaid amount that can be used by a SM (or the home owner) to pay for the corresponding electricity consumption. The token reveals no information about the underlying SM or home owner. Tokens are submitted with the anonymized data (Section 2.2.1) and the recipient (RS or UP) has first to verify their validity and then verify whether the tokens have been spent before.

For the purpose of our analysis, we assume that tokens are generated by both the UP and the users. In order to ensure that tokens are untraceable and to protect the privacy of users, tokens are not be associated with a particular user, but will solely contain a unique identifier, the expiration date, and the amount of energy that the token is worth. As we describe later, we require that each token is blindly signed by UP [16]. This ensures that the UP cannot insert *distinguishers* (e.g., unique identifiers) to track users while signing a given message.

We assume that tokens are purchased by a user U using an anonymous channel using e.g., a gift card or through the RS. As such, no information is leaked about the identity of U throughout the purchase of the token. Finally, we assume

that the UP keeps track of the unique identifiers of spent tokens (e.g., by means of a hashtable) in order to prevent double-spending attempts.

3.2 Anonymous Trade of Energy

In what follows, we describe our protocol that enables the generation and exchange of secure and anonymous tokens of energy.

Let p and q be primes such that $q|p-1$ and let g be a generator of order q in the group $\mathcal{Z}_p^*$. Typically, p and q are expected to be 1024-bit and 160-bit long, respectively. Each user U then selects two secret values $s_i, r_i \in \mathcal{Z}_p$ and computes $v_i = g^{-s_i} \mod p$ and $x_i = g^{r_i} \mod q$. The pair (v_i, x_i) constitutes a unique identifier for token i. The user U will then *blind* the pair (v_i, x_i). This can be achieved by choosing two random values $R_1, R_2 \in \mathcal{Z}_p$ and computing the pair $(R_1 v_i, R_2 x_i)$. $(R_1 v_i, R_2 x_i)$ is then submitted to the UP so that it can blindly sign it. Note that this process ensures that the UP does not learn any information about (v_i, x_i) at this stage, since R_1, R_2 are random group elements.

The UP submits a blind signature on the pair $(R_1^i v_i, R_2^i x_i)$ as follows. We assume that the UP is equipped with a public/private pair (e_1, d_1) (computed using a semi-prime N). For each token, the UP computes a new public/private key (e_2, d_2) as follows. It constructs message $m \leftarrow \langle \text{Value} \,\|\, \text{date} \,\|\, \text{time} \rangle$, where Value denotes the energy value to be consumed (e.g., in kwh), and $\|$ denotes message concatenation. The UP then generates the public key $e_2 = F(m, Sig_{d_1}\{m\})$, where $F : \{0,1\}^* \rightarrow \{0,1\}^*$ is a collision-resistant pseudorandom function and the corresponding private key d_2 (modulo a fresh semi-prime). The UP then sends to U the pair $\langle e_2, \sigma \rangle$, where $\sigma \leftarrow Sig_{d_2}\{R_1^i v_i\} \| Sig_{d_2}\{R_2^i x_i\}$. Here, it is easy to see that σ is a blind signature on $(R_1^i v_i, R_2^i x_i)$ that can only be created by the UP (since it is a function of the private key of the UP); however, the UP can ensure that its blind signature cannot be misused by any user. This is case since the private key used to sign $(R_1^i v_i, R_2^i x_i)$, is computed as a function of the value of the token and its validity. This also means that no user can (ab-)use σ to claim a token with different value.

Upon the reception of $\langle e_2, \sigma \rangle$, the user U verifies that e_2 is correctly computed (i.e., that $e_2 = F(m, Sig_{d_1}\{m\})$. U then extracts the signatures on the (v_i, x_i) as follows: $\sigma_1^i = Sig_{d_2}\{v_i\} \leftarrow {R_1^i}^{e_2} \cdot Sig_{d_2}\{R_1^i v_i\}$ and $\sigma_2^i = Sig_{d_2}\{x_i\} \leftarrow {R_1^i}^{e_2} \cdot Sig_{d_2}\{R_1^i x_i\}$. Given this, U is now equipped with a valid token $\tau \leftarrow \langle e_2, \text{Value}, v_i, \sigma_1^i, x_i, \sigma_2^i \rangle$ for Value kwh signed by the UP.

When U wishes to pay to the UP in exchange for energy consumption, it sends the UP the token τ. The UP first verifies that the signatures σ_1^i and σ_2^i are correct, and that they correspond to the authentic public key e_2 corresponding to Value. If this verification passes, it then performs a non-interactive identification protocol based on the identification scheme of Schnorr [17] to validate that U indeed knows the secrets s_i and r_i. More specifically, U sends to the UP $\langle y, date/time \rangle$, where $y = r_i + e_i s_i \mod q$ and $e_i = F(\text{token}, date/time)$.

Given the token τ, the UP verifies that $x_i = g^y v_i^{e_i} \mod p$. If the verifications succeed, the UP considers the token *valid*. However, the UP still has to determine whether the token is *fresh* or an attempt of double-spending. For that reason, we assume that the UP maintains a database of received tokens. Thus, the UP searches its records for a token containing the same v_i, x_i values. If no match is found, the token is considered fresh and the UP records the values $\langle \text{token}, y, date/time \rangle$.

Note that if a match is found, then the UP can provide strong evidence that the token has been used before. In fact, our protocol guarantees that if the token is double-spent then the secret values s_i, r_i chosen by U can be acquired by the UP; the UP can then submit the computed s_i, r_i to prove the existence of a double-spending attempt by a given user U. To see why this is sufficient evidence, we note that there will be two transcripts $\langle \text{token}, y, date/time \rangle$ and $\langle \text{token}, y', date'/time' \rangle$ such that $x_i = g^y v_i^e \mod p$ and $x_i = g^{y'} v_i^{e_i'} \mod p$. This enables the UP to compute $s_i = \frac{(y-y')}{(e_i - e_i')} \mod q$ by solving the equations:

$$y = r_i + e_i s_i \mod q,$$
$$y' = r_i + e_i' s_i \mod q.$$

The UP can also obtain r_i in a similar way. Note that (s_i, r_i) are not connected with the ID of U, hence they are not used in identifying U. They are only used to provide evidence that a token has been double-spent. We point out, however, that the communication between U and the UP needs to be performed over anonymous channels so that the UP does not leak information about U's location e.g., the IP of U. This can be achieved through the use of TOR or by relying on the RS to mediate the exchange of messages between U and the UP. Note that while the RS is trusted not to leak information about U to the UP (i.e., to anonymize the messages of U by inserting its own ID as the originator of messages), our proposed solution ensures that the RS cannot assume property of the token possessed by U; this is the case since the secret values (s_i, r_i) corresponding to a token are not revealed to any party, including the RS.

Trading Energy: As shown in Figure 3, our aforementioned protocols enable *(i)* a privacy-preserving payment by the SMs to the UP in exchange of energy and *(ii)* a privacy-preserving payment by the UP to the SMs in exchange for a purchase of energy from the users. In the former case, users of the smart grid simply have to purchase tokens from the UP and submit them along with their measurement reports as a proof of payment. In the latter case, the protocols depicted in Figure 3 have to be preceded by a "negotiation" phase where the user U and the UP negotiate a "price" in exchange of the energy that the SM of U will redirect to the UP. Note that this negotiation phase can be mediated by the RS, to ensure that the identity of U is not revealed in the process. Once the negotiation phase is completed and both the UP and U agree on a price Value, the UP issues a token to U according to the agreed upon Value; U can subsequently use that token to purchase energy from the UP at a later point in time (e.g., during periods where U's solar energy depletes). Note that, given our protocols, both the purchasing and the selling of energy by U can be achieved without leaking any information about U.

4. RELATED WORK

In this section, we overview related work in the area. Most contributions in the literature focus on the privacy threats that result from the analysis of the SMs' reports; as far as we are aware, this is the first contribution that addresses privacy issues with respect to tasking and energy barter in

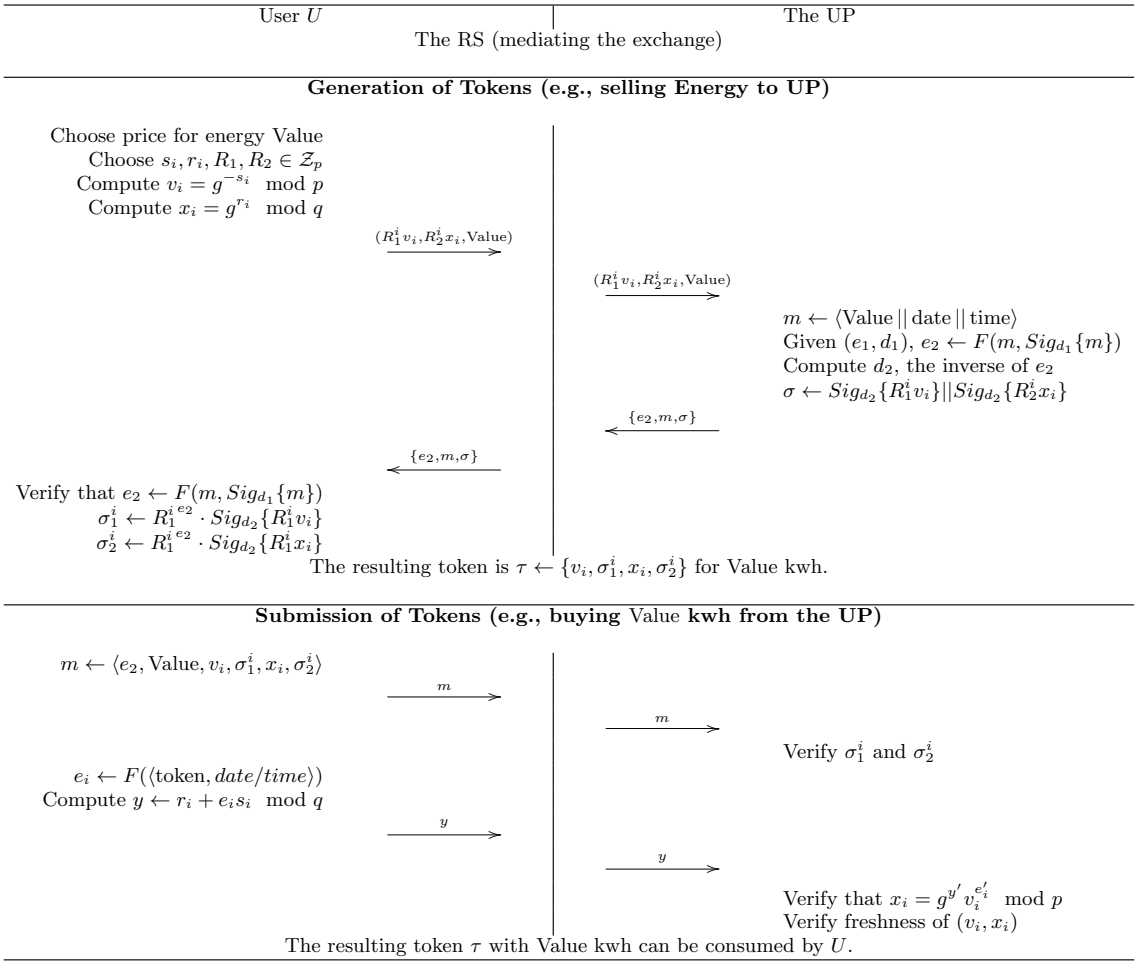

Figure 3: Trading energy between U and the UP. Here, the RS mediates the communication between the user U and the UP. We assume the existence of secure and authenticated channels between *(i)* U and the RS, and *(ii)* the RS and the UP, respectively.

smart grids.

In [2], Molina-Markham *et al.* demonstrate that power traces can reveal fine-grained usage patterns and as such reveal a range of private information about customers. They also propose a scheme based on "neighborhood gateways" to enhance the privacy of users. In their scheme, neighborhood gateways act as relays between the provider and the users so as to hide the correlation between the power consumption trace from the total consumption.

A number of contributions suggest the reliance on rechargeable batteries to obfuscate the detailed energy consumption of individual households. Here, the main intuition is to install a rechargeable battery within each household and to use it as the main supplier of energy. In this way, the UP can only see constant energy consumption (i.e., that of the battery). In [18], Kalogridis *et al.* analyze a power mixing model that relies on the use of rechargeable batteries and the evaluate it with respect to a number of privacy-related metrics. The use of rechargeable batteries is further analyzed in [20] where it was shown that the batteries still leak some information when compared to a device that would always hold the output load constant whenever possible. One major limitation behind the use of batteries is that the resulting consumption trace does not offer the UP any information about the generic consumption patterns of neighborhoods and as such masks one of the most important advantages of smart grids [21].

The concept of privacy preserving aggregation has also been proposed in [14]. Here, the authors describe a protocol where a collection of smart meters interact with a local substation which is responsible for producing the aggregated results. We point out that the protocol is not efficient with respect to the communication rounds that it incurs since the SMs use the substation as an intermediary to send their shares to the other SMs and receive back aggregated results. Then, in another step, the substation collects final contributions from all users and adds them to obtain the aggregated consumption values. In [5], Efthymiou *et al.* propose an escrow scheme that relies on a trusted party to protect SMs' reports.

In [15], Kursawe *et al.* discuss a scheme for aggregating energy traces by blinding the power traces of users with some randomness, that would cancel out once enough power traces are aggregated together. In [7], Rial *et al.* propose

a method to calculate energy fees while protecting meter data using ZK proofs and commitment schemes. Here, the provider is ensured that the correct fee is calculated but no detailed readings are learnt during the process. This solution can only ensure a correct calculation of fees in exchange of the energy consumed by the clients, but do not enable clients to "sell" their energy back to the grid in a privacy-preserving manner. In this work, we propose the reliance on anonymous energy tokens to enable a privacy-preserving trade of energy between the SM (or the home owner) and the UP.

5. CONCLUSION

Within existing smart grids, smart meters undergo a set of essential operations: collection of measurements, reception and execution of (maintenance) tasks, and billing and trading of energy with the utility provider (in case of a surplus of energy). In this paper, we presented a set of solutions and protocols that protect the privacy of smart meters and home owners when subject to these aforementioned use-cases.

For that purpose, we proposed a solution that relies on secret sharing among dedicated report servers to enable a privacy-preserving aggregation of the smart-meter energy consumption reports. We then discussed the privacy implications of executing additional tasks (e.g., for maintenance reasons) that might be requested from the utility provider. We showed that such tasks could be abused by the utility provider to de-anonymize users and we discussed possible ways to alleviate this threat. Finally, we proposed a set of solutions that *(i)* enable a privacy-preserving reward mechanism (e.g., Bitcoin) and *(ii)* that rely on the generation of anonymous "energy tokens" in order to enable a secure barter of energy between users and utility providers. Our proposals complement previous work in the area and can be easily integrated within existing smart grids.

6. REFERENCES

[1] H. Y. Lam, G. S. K. Fung, and W. K. Lee, "A Novel Method to Construct Taxonomy Electrical Appliances Based on Load Signature", In IEEE Transactions on Consumer Electronics, vol. 53, no. 2, pp. 653-660, 2007.

[2] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private Memoirs of a Smart Meter", In the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys 2010), Zurich, Switzerland, November 2010.

[3] E. L. Quinn, "Privacy and the New Energy Infrastructure", Center for Energy and Environmental Security (CEES), Working Paper No. 09-001.

[4] NIST, "Guidelines for Smart Grid Cyber Security, Vol. 2, Privacy and the Smart Grid", NISTIR 7628, August 2010.

[5] C. Efthymiou and G. Kalogridis, "Smart Grid Privacy via Anonymization of Smart Metering Data", In Proceedings of IEEE SmartGridComm, 2010.

[6] F. Li, B. Luo, and P. Liu, "Secure information Aggregation for Smart Grids using Homomorphic Encryption", In the 1st IEEE International Conference on Smart Grid Communications, 2010.

[7] A. Rial, G. Danezis, "Privacy-Preserving Smart Metering", In Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society (WPES), 2011.

[8] S. Raj Rajagopalan, L. Sankar, S. Mohajer, H. Vincent Poor, "Smart Meter Privacy: A Utility-Privacy Framework", In Proceedings of IEEE SmartGridComm, 2011.

[9] R. Dingledine, N. Mathewson and P. Syverson,"TOR: The Second-Generation Onion Router", In Proceedings of the 13th Conference on USENIX Security Symposium (USENIX Security), pp. 21-38, 2004.

[10] L. Sweeney, "K-anonymity: a Model for Protecting Privacy", In International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems 10, 557-570, 2002.

[11] D. Chaum and E. van Heyst, "Group Signatures", Advances in Cryptology–EUROCRYPT 1991, Volume 547 of Lecture Notes in Computer Science. pp. 257-265, 1991.

[12] A. Shamir, "How to Share a Secret", In Communications of the ACM 22 (11): 612-613, 1979.

[13] B. Schoenmakers, "A Simple Publicly Verifiable Secret Sharing Scheme and its Application to Electronic Voting", In Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology–CRYPTO 1999, pp. 148-164.

[14] F.D. Garcia, B. Jacobs, "Privacy-Friendly Energy-Metering via Homomorphic Encryption", In Proceedings of the 6th Workshop on Security and Trust Management (STM), 2010.

[15] K. Kursawe, G. Danezis, and M Kohlweiss, "Privacy-Friendly Aggregation for the Smart-Grid", In Proceedings of PETS, 2011.

[16] D. Chaum, "Blind Signatures for Untraceable Payments", In CRYPTO, 1982.

[17] C.P. Schnorr, "Efficient Signature Generation by Smart Cards", In Journal of Cryptology 4(3), 1991.

[18] G. Kalogridis, C. Efthymiou, S.Z. Denic, T.A. Lewis, and R. Cepeda, "Privacy for Smart Meters: Towards Undetectable Appliance Load Signatures", In Proceedings of IEEE SmartGridComm 2010.

[19] G. Karame, E. Androulaki, and S. Capkun, "Double-Spending Fast Payments in Bitcoin", In Proceedings of ACM CCS 2012.

[20] D. P. Varodayan, A. Khisti, "Smart Meter Privacy Using a Rechargeable Battery: Minimizing the Rate of Information Leakage", In ICASSP 2011.

[21] P. McDaniel and S. McLaughlin, "Security and Privacy Challenges in the Smart Grid", In IEEE Security and Privacy, Vol. 7, No. 3., 2009.

[22] F. Cornelli, E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing Reputable Servents in a P2P Network", In 11th International World Wide Web Conference, 2002.

[23] T. Dimitriou, G. Karame, and I. Christou, "SuperTrust – A Secure and Efficient Framework for Handling Trust in Super Peer Networks", In Proceedings of PODC, 2007.

[24] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, 2009.

[25] Contracts – Bitcoin, Available from `https://en.bitcoin.it/wiki/Contracts`