iKUP Keeps Users' Privacy in the Smart Grid

Fábio Borges
Technische Universität Darmstadt, TK / CASED
64289 Darmstadt, Germany
fabio.borges@cased.de

Leonardo A. Martucci
Karlstad University
651 88 Karlstad, Sweden
leonardo.martucci@kau.se

***Abstract*—Privacy-enhancing technologies for the Smart Grid usually address either the consolidation of users' energy consumption or the verification of billing information. The goal of this paper is to introduce iKUP, a protocol that addresses both problems simultaneously. iKUP is an efficient privacy-enhancing protocol based on DC-Nets and Elliptic Curve Cryptography as Commitment. It covers the entire cycle of power provisioning, consumption, billing, and verification. iKUP allows: (i) utility providers to obtain a consolidated energy consumption value that relates to the consumption of a user set, (ii) utility providers to verify the correctness of this consolidated value, and (iii) the verification of the correctness of the billing information by both utility providers and users. iKUP prevents utility providers from identifying individual contributions to the consolidated value and, therefore, protects the users' privacy. The analytical performance evaluation of iKUP is validated through simulation using as input a real-world data set with over 157 million measurements collected from 6,345 smart meters. Our results show that iKUP has a worse performance than other protocols in aggregation and decryption, which are operations that happen only once per round of measurements and, thus, have a low impact in the total protocol performance. iKUP heavily outperforms other protocols in encryption, which is the most demanded cryptographic function, has the highest impact on the overall protocol performance, and it is executed in the smart meters.**

I. Introduction

Security and privacy mechanisms for the Smart Grid aim to

(*a*). secure the data communication flow between the multiple parties involved and

(*b*). prevent consumption data to be used to profile users.

The Smart Grid is an electricity power grid that provides constant feedback to utilities concerning the power load in each segment of the grid. This is realized by replacing electromechanical meters by electronic meters, or smart meters, which have network interfaces that are used to report the power consumption on a regular basis. The Smart Grid enables multiple services. It allows utilities to fluctuate the kilowatt-hour retail price according to the demand and to remote control appliances and micro-generation of electricity at the users' premises. Furthermore, the Smart Grid promotes transparency by providing means to users to closely monitor the evolution of their power consumption in real time.

Security and privacy in the Smart Grid are paramount. Power grids are part of every national infrastructure, and require proper security solutions to be deployed for both users and utilities. The constant monitoring of household electricity consumption allows utilities to build user profiles concerning people's habits and activities. The threat to privacy has delayed or stalled the deployment of smart meters in the Netherlands and Germany, which jeopardizes the EU Energy Efficiency Directive [1] goal of having 80% of households equipped with smart meters by 2020. Compliance with the relevant EU data protection and privacy legislation is included in the Directive.

Therefore, security and privacy-enhancing mechanisms need to be designed and deployed to benefit all the involved parties. They should: (*a*) not prevent utilities from obtaining real-time data that is needed to manage the power grid, (*b*) assist users to exercise control over their personal data, and (*c*) help all parties to verify the correctness of the billing information and, thus, promote transparency.

The matter of this work is to introduce iKUP, a comprehensive, efficient, and secure privacy-enhancing protocol for the Smart Grid. iKUP is based on DC-Nets and Elliptic Curve Cryptography with Commitments. It is comprehensive because it addresses two problems with a single protocol set: (*i*) the consolidation of multiple users' energy consumption into a single collective value, which a utility cannot distinguish its component parts, and (*ii*) the verification of billing information by all parties. iKUP is the first protocol that addresses both questions at once, to the best of our knowledge. It consolidates data and calculates the aggregated power consumption without leaking personal information, and it verifies the correctness of the consolidated billing information without knowing its parts.

iKUP heavily outperforms other proposed protocols in terms of encryption, which is the most demanded cryptographic function, and it is executed in the network leaf nodes, i.e., the smart meters, and has the highest impact on the overall protocol performance. It has a worse performance in decryption and aggregation than the related work, but these happen only once per round of measurements and, thus, have a low overall impact on the protocol performance. The performance of iKUP is evaluated with simulation using as input a real-world data set with more than 157×10^6 measurements collected from $6,345$ smart meters in Ireland.

In the remainder of this paper, we begin with the background information and summarize the related work. We then detail iKUP and its security and privacy analysis. Finally, we show the simulation and its results, discuss our findings, and present our conclusions.

II. Background

The techniques behind iKUP are three: the DC-Net, an anonymous communication protocol, and Elliptic Curve Cryptography (ECC) with Commitments, a cryptographic scheme. We briefly introduce them in this section.

A. DC-Net

DC-Net is a protocol that addresses the dining cryptographers problem, i.e., how to make public messages untraceable. A DC-Net can provide either an unconditionally secure untraceable-sender system or a computationally secure system, depending on how pair-wise secret keys are agreed upon or distributed among the participants [2].

A DC-Net works with a round-based, superposed sending [3]. Each pair of participants p in a DC-Net share a secret key k, where one participant stores $+k$ and the other $-k$. Each participant p chooses a message m and combines it with k, resulting in an encrypted message c. The sum $s = \sum(c)$ of all c is broadcasted to all participants from the DC-Net. It equals to the sum of all messages m, as $\sum(k) = 0$. If all but one message is not 0, then s equals that one message.

B. ECC

Elliptic Curve Cryptography (ECC) [4] is based on the assumption that is computationally intractable to find an integer k, such that $Q = k \cdot P$, where P and Q are two given points in an elliptic curve. The operation $k \cdot P$ is a scalar multiplication. Its advantage over a modular exponentiation operation is that it requires shorter cryptographic keys and it is, therefore, faster to compute.

An elliptic curve Ω over a field $\mathbb{F}$ is defined by the Weierstrass equation as

$$\Omega : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

where $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}$ and the discriminant of Ω is zero. The points of an elliptic curve have a group structure when considering that the point at infinity ∞ is the identity. Therefore, the number of elements in the group is the total of ordered pairs $(x, y) \in \Omega$ plus one. This structure is the base of the security in ECC. A simplified form of the Weierstrass equation is commonly used in cryptographic applications.

C. Commitment Schemes

Commitment schemes have two functions. The first conceals a committed value m by combining it with a random value r and outputs the commitment $c \leftarrow \mathsf{Commit}(m, r)$. The second function $\mathsf{Open}(c, m, r)$ verifies if c is a commitment of m and r, and it outputs either true ($\top$) or false ($\bot$).

The Pedersen commitment [5] offers unconditional hiding, i.e., it does not leak any information about m, and computational binding, i.e., given c, m, and r, it is hard to compute $m' \neq m$ and $r' \neq r$, such that $\top \leftarrow \mathsf{Open}(c, m', r')$. It is also homomorphic, i.e.,

$$\mathsf{Commit}(a, r) \cdot \mathsf{Commit}(b, s) = \mathsf{Commit}(a + b, r + s)$$

III. Related Work

In this section, we present privacy-enhancing protocols for the Smart Grid that are based on the DC-Net and homomorphic encryption, which aim to cryptographically sum the measurements of multiple users and deliver the result to the utility, which is then unable to determine the contribution of each user to the end result. We also present a protocol that allows the verification of the billing information through commitments.

A. Protocols Based on DC-Nets

Privacy-enhancing protocols based on DC-Nets were proposed in [6], [7]. In this section, we revisit the most efficient of all them in terms of communication overhead, the Low-Overhead Protocol (LOP) [6].

In LOP set-up phase, a set of meters $\mathcal{M}$ agree on a safe hash function H. Every meter $i \in \mathcal{M}$ also agrees on a pairwise key $k_{i \to o}$ with every meter $o \in \mathcal{M} - \{i\}$. Session keys $x_{i,t}$ are valid for a single round j and are computed as follows

$$x_{i,j} = \sum_{o \in \mathcal{M} - \{i\}} (-1)^{o<i} \, \mathsf{H}(\ k_{i \to o} \ || \ j\ )$$

where $||$ is the concatenation function.

In the operation phase, a meter i computes $c_{i,j} = x_{i,j} + m_{i,j}$, where $m_{i,j}$ is a measurement from the meter i for the round j. The meter optionally signs the encrypted message and sends it to the utility, which calculates $\sum c_{i,j} = \sum m_{i,j}$.

B. Protocols Based on Homomorphic Encryption

Paillier's scheme [8] is an additive homomorphic encryption function that is the fundamental building block of several privacy-enhancing protocols for the Smart Grid [9]–[11]. The aggregation of measurements happens either in a trusted-third party or in the smart meters, assuming that they are able to establish pairwise communication channels. On every round j, the consolidated measurement e is produced as follows

$$e = \prod_{i=1}^{|\mathcal{M}|} \mathsf{Enc}(m_{i,j}), \tag{1}$$

where Enc is the encryption function of Paillier's scheme, $|\mathcal{M}|$ is the cardinality of the set of meters $\mathcal{M}$, and $m_{i,j}$ is a measurement from meter i for a round j. The utility calculates $\mathsf{Dec}(e) = \sum m_{i,j}$, where $\mathsf{Dec} = \mathsf{Enc}^{-1}$, and obtains the consolidated measurement in plaintext.

Protocols that use the Paillier's scheme assume that adversaries neither calculate $\mathsf{Dec}(e)$, nor tamper with, nor eavesdrop the communication network. If utilities are adversaries, then e has to be computed either by a trusted third party or by the meters because the utilities can run $\mathsf{Dec}(m_{i,j})$.

It is possible to allow adversarial utilities to compute e, if the exponents in the Paillier's scheme are modified to prevent that the correct output is produced by Dec, unless all measurements are included in the calculation [7], [12].

Protocols that use the Paillier's scheme tend to be malleable, i.e., there is no detection of modifications in the ciphertext. Homomorphic signatures can be used to achieve non-malleability

[13]. The utility verifies the signature correctness to detect if the ciphertext was tampered with.

C. Verification of Billing Information with Commitments

In the Smart Grid, commitments can be used to verify the correctness of billing information. Moreover, their homomorphic properties can be used to multiply measurements and the current market price of electricity. Thus, users can track their billing information and protect their personal data, and the utility can change their tariffs according to the demand. A privacy-preserving protocol for the Smart Grid based on commitments is proposed in [14]. It works as follows.

A utility distributes an array p of pricing information that is valid for a time interval Δt, e.g., a day, to all meters in $\mathcal{M}$. On every round j, $0 \leqslant j \leqslant \Delta t$, the meters send to the utility a commitment $c_j \leftarrow \mathsf{Commit}(m_j \cdot p_j, r_j)$, where p_j is the price, m_j is a measurement, and r_j is a random number. In the end of Δt, the meters send $r = \sum r_j$ and $m = \sum m_j \cdot p_j$ to the utility. The utility computes $\prod_{j=1}^{\Delta t} c_j = \mathsf{Commit}(m, r)$, and verifies the correctness of the reported billing information by opening the commitment. No individual measurements are ever reported in this protocol and the utility only has access to the final billing information in the end of Δt. On the downside, this protocol neutralizes an important aspect of the Smart Grid: no (close to) real-time data can be obtained from the meters.

Concerning efficiency, the Pedersen commitment and the Paillier's scheme are equally efficient for prime numbers of the same length and randomly chosen parameters. While the Pedersen commitment has two modular exponentiation functions and shorter exponents, the Paillier's scheme has just one exponentiation with a longer exponent. Thus, a Pedersen commitment with 512-bit exponents will perform equally to a Paillier's scheme with a 1024-bit exponent on average.

IV. iKUP Keeps Users' Privacy

In this section, we present iKUP. We begin with a summary on how iKUP works and, then, present its network model and assumptions. We follow with the iKUP's cryptographic mechanisms and explain how it preserves users' privacy and helps the verification of the billing information.

In iKUP every meter runs two functions:

- encryption: $c \leftarrow \mathsf{Enc}(m)$ and
- commitment: $C \leftarrow \mathsf{Commit}(m)$,

where $m, c \in \mathbb{Z}$, $C \in \Omega$, and the utility runs the functions:

- decryption: $m \leftarrow \mathsf{Dec}(\,\mathsf{Enc}(m)\,)$ and
- open: $\{\top, \bot\} \leftarrow \mathsf{Open}(C, m, r)$,

where $r \in \mathbb{Z}$ is a random value.

In iKUP the utility receives an encrypted consolidated measurement $\mathsf{Enc}(\sum m)$, concerning measurements from all meters in a set $\mathcal{M}$ in a round j. The $\sum m$ is the results of an in-network aggregation operation, i.e., encrypted measurements c are added up by the meters. The utility then decrypts $\mathsf{Enc}(\sum m)$ and retrieves $\sum m$. Each meter send a commitment C of its measurement directly to the utility, which sums up the commitments from all meters and opens it using $\sum m$ as parameter, i.e., $\mathsf{Open}(\sum C, \sum m)$. Fig. 1 illustrates iKUP.

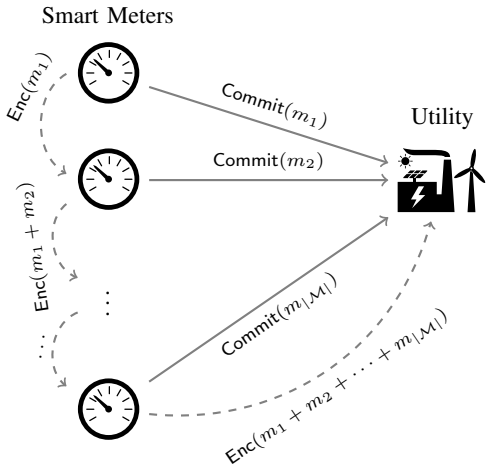

Figure 1. iKUP in a nutshell. Encrypted in-network data aggregation and commitments are sent directly to the utility.

A. Network Model

In our network model, we consider a set $\mathcal{M}$ of meters and one utility that has communication channels to all meters in $\mathcal{M}$. A meter $i \in \mathcal{M}$ has a communication channel to meters $i-1$ and $i+1$, where $1 < i < |\mathcal{M}|-1$. The meters are logically organized as a spanning tree, i.e., a connected, undirected graph that include all meters and has no cycles.

B. Assumptions

We assume that meters have two cryptographic keys, which are unique for each meter: (*a*) a symmetric key d that is shared with the utility and (*b*) a private key k, where both keys were either securely distributed or generated. We assume that there exist a secure elliptic curve Ω over a finite field $\mathbb{F}_q$ with a base point P of high order and a hash function $R \leftarrow H_\Omega(j)$, where $R \in \Omega$ is a curve element of high order, and j is the round number, which is unique and incremental, e.g., time and date. We define the $H_\Omega(j)$ as follows:

$$\begin{aligned} R_j = H_\Omega(j) &= (x, y) \\ &= \left(\min\left(\left\{r | r \geqslant \mathsf{H}(j) \text{ and } (r, y) \in \overline{\Omega}\right\}\right), y\right), \end{aligned} \tag{2}$$

where H is a secure hash function, e.g., SHA-2 and SHA-3, $\overline{\Omega}$ is a subset of Ω that contains elements of high order, and x and y are the coordinates of a point in the curve.

The cryptographic keys are random numbers belonging to the finite field $\mathbb{F}_q$. We assume that the utility knows the sum $\mathfrak{K}$ of the private keys k from all meters in $\mathcal{M}$, i.e., $\mathfrak{K} = \sum k$, and meters can establish secure communication channels. The selection of secure and efficient ECC parameters is further discussed in Sec. V.

C. Encryption and Commitments on the Smart Meters

The meters run two cryptographic functions: one for ciphering measurements and consolidating them with encrypted measurements from other meters and another for committing

their reported measurements to the utility. In this section, we present the details of both operations.

1) Encryption and Data Consolidation: in iKUP, a meter i outputs, for every round j, the encrypted message $c_{i,j}$, where

$$c_{i,j} \leftarrow \mathsf{Enc}(m_{i,j}) = m_{i,j} + \mathsf{H}(\, d_i \,||\, j\,). \tag{3}$$

The meter i sends $c_{i,j}$, over a secure channel, to the next vertex in the spanning tree, $i+1$, which sums its $c_{i,j}$ to its own encrypted measurement $c_{i+1,j}$, where $\mathsf{Enc}(m_{i+1,j} + m_{i,j})$ is

$$c_{i+1,j} + c_{i,j} = m_{i+1,j} + \mathsf{H}(\, d_{i+1} \,||\, j\,) + m_{i,j} + \mathsf{H}(\, d_i \,||\, j\,).$$

If the meter is the end node of the spanning tree, i.e., $i = |\mathcal{M}|$, it first adds its measurement to the measurements of all other meters in $\mathcal{M}$ and, then, sends the resulting $\mathfrak{c}_j$, i.e., the consolidated measurement, to the utility, where

$$\mathfrak{c}_j = \sum_{i=1}^{|\mathcal{M}|} c_{i,j} = \sum_{i=1}^{|\mathcal{M}|} \mathsf{Enc}(m_{i,j}) = \sum_{i=1}^{|\mathcal{M}|} m_{i,j} + \mathsf{H}(d_i||j). \tag{4}$$

2) Commitment of Measurements: a meter i outputs, for every round j, a commitment $C_{i,j}$, where

$$C_{i,j} \leftarrow \mathsf{Commit}(m_{i,j}) = k_i \cdot R_j + m_{i,j} \cdot P. \tag{5}$$

The meter then signs $C_{i,j}$ using its private key k_i and sends the result to the utility.

D. Decryption and Round Verification on the Utility

The utility runs two cryptographic functions: one for decrypting the consolidated measurement $\mathfrak{c}_j$ and another for opening and verifying the commitments.

1) Decryption of Consolidated Measurements: the utility decrypts $\mathfrak{c}_j$, see Eq. (4), and obtains $\mathfrak{m}_j$, which is the consolidated measurement in a round j, where

$$\mathfrak{m}_j \leftarrow \mathsf{Dec}(\mathfrak{c}_j) = \mathfrak{c}_j - \sum_{i=1}^{|\mathcal{M}|} \mathsf{H}(d_i||j) = \sum_{i=1}^{|\mathcal{M}|} m_{i,j}. \tag{6}$$

2) Opening Commitments: the utility is not able to open the commitments $C_{i,j}$ received directly from the meters as it does not know k_i and $m_{i,j}$. The utility does, nevertheless, the verification over the sum of all commitments and measurements. First, it verifies the digital signatures on $C_{i,j}$. If the signatures are correct, the utility then sums all commitments and obtains $\mathfrak{C}_j = \sum C_{i,j}$, where

$$\mathfrak{C}_j = \sum_{i=1}^{|\mathcal{M}|} (k_i \cdot R_j + m_{i,j} \cdot P) = \mathfrak{K} \cdot R_j + \mathfrak{m}_j \cdot P. \tag{7}$$

The utility opens the commitment

$$\{\top, \bot\} \leftarrow \mathsf{Open}(\mathfrak{C}_j,\, \mathfrak{A},\, \mathfrak{K} \cdot R_j) = \mathfrak{C}_j - \mathfrak{K} \cdot R_j - \mathfrak{A} \cdot P, \tag{8}$$

which returns the point at infinity for a correct value, i.e., true if and only if $\mathfrak{A} == \mathfrak{m}_j$, and returns any other point in Ω for incorrect values, i.e, false otherwise.

E. Verification of Measurements and Billing Information

In iKUP, users and the utility verify the correctness of the reported measurements and of the billing information. In this section, we explain how this verification is performed, and how iKUP handles the verification of billing information assuming a dynamic pricing policy.

1) Verification Property: a meter proves the correctness of a measurement $m_{i,j}$ in a round j by computing $V_{i,j} = k_i \cdot R_j$ and sending it with $m_{i,j}$ to the utility. The utility opens the commitment $c_{i,j}$, $\mathsf{Open}(c_{i,j},\, m_{i,j},\, V_{i,j})$, which returns true for a correct value. iKUP allows the verification of batches of measurements from a meter, which is more computationally efficient and prevent utilities from obtaining information about single measurements. For the verification of a subset of measurements $\mathcal{S}$, a meter i calculates $\mathfrak{w}_{i,\mathcal{S}}$, where

$$\mathfrak{w}_{i,\mathcal{S}} = \sum_{j \in \mathcal{S}} m_{i,j}, \tag{9}$$

and sends it to the utility with $\mathcal{V}_{i,\mathcal{S}} = k_i \cdot \mathfrak{R}_{\mathcal{S}}$, where

$$\mathfrak{R}_{\mathcal{S}} = \sum_{j \in \mathcal{S}} R_j. \tag{10}$$

The utility receives the pair $(\mathfrak{w}_{i,\mathcal{S}},\, \mathfrak{R}_{\mathcal{S}})$ from the meter i, computes the sum of commitments $\mathcal{C}_{i,\mathcal{S}}$, where

$$\mathcal{C}_{i,\mathcal{S}} = \sum_{j \in \mathcal{S}} C_{i,j},$$

and runs the function $\mathsf{Open}(\mathcal{C}_{i,\mathcal{S}}, \mathfrak{w}_{i,\mathcal{S}}, \mathcal{V}_{i,\mathcal{S}})$, which returns true for a correct value.

2) Billing Verification with Dynamic Pricing Policy: iKUP allows the billing information (with a dynamic pricing policy) to be verified by all parties. For implementing dynamic pricing, the utilities send the array of prices p to the meters. A meter multiplies its measurement $m_{i,j}$ (in Watt) and the price p_j (in \$/Watt) and use the resulting value as input parameter in Eq. (5), such as $C'_{i,j} = \mathsf{Commit}(m_{i,j}) = k_i \cdot R_j + p_j \cdot m_{i,j} \cdot P$. The utility opens $\mathfrak{C'}_j = \sum C'_{i,j}$ with

$$\mathsf{Open}(\mathfrak{C}'_j,\, \mathfrak{A},\, \mathfrak{K} \cdot R_j) = \mathfrak{C'}_j - \mathfrak{K} \cdot R_j - p_j \cdot \mathfrak{A} \cdot P.$$

The measurements $m_{i,j}$ also need to be multiplied by p_j in the functions Enc and Dec in Eq. (3) and Eq. (6), and in the verification presented in Sec. IV-E1.

V. Security and Privacy Analysis

In this section, we present the security and privacy analysis of iKUP. Furthermore, we present how to select secure elliptic curve parameters.

A. Security Analysis

iKUP is secure against a Dolev-Yao adversary, i.e., it is limited only by the constraints of the cryptographic functions.

A malicious meter that injects erroneous measurements will be detected by the utility because it has the commitments and it knows how much electricity it inputs in the power grid at all times and, therefore, the utility knows the expected aggregated power consumption of the totality of its users. If the expected

consumption is not equal to the consolidated measurements, the cause is malfunction or fraud.

A malicious utility can eavesdrop the communication channel to obtain $m_{i,j}$ from $c_{i,j}$. Similar to protocols based on homomorphic encryption, iKUP assumes that there is a secure channel between meters.

The utility needs to know the sum $\mathfrak{K} = \sum k$ of all k that are randomly chosen by the meters without knowing the individual values of k. The meters may compute and send the sum $\mathfrak{K}$ to the utility using a protocol based on homomorphic encryption that is presented in the Sec. II, e.g., [9].

Utility and meters have the same value of R_j in the round j. However, it is infeasible to relate the values of R_j between rounds of measurements j, i.e., there is no practical relation between $H_\Omega(j)$ and $H_\Omega(j+1)$, because there is no known relationship between $\mathsf{H}(j)$ and $\mathsf{H}(j+1)$. Therefore, rounds of measurements cannot be related, because iKUP computes the encryption and commitment with a secure hash function.

B. Privacy Analysis

Keeping privacy means that no one can read individual measurements $m_{i,j}$. Meters should sign and send their commitments directly to their utility. Nevertheless, utility cannot read individual measurements, but can only check the consolidated measurement $\mathfrak{C}_j$ given by Eq. (7). Since it only knows the aggregated keys $\mathfrak{K} = \sum k$ of its users, it is infeasible to reveal an individual key k_i. Indeed, $|\mathcal{M}| - 1$ meter may cooperate to disclose an individual key k_i but the collaboration of $|\mathcal{M}| - 2$ is not enough to disclose a key. Leaking of $m_{i,j}$ in Eq. (5) is related to security issues, since $m_{i,j} \cdot P = C_{i,j} - k_i \cdot R_j$. In this case, $m_{i,j}$ is a small number that could be searched, if and only if, k_i is known. If a utility cannot read $m_{i,j}$, even less an adversary.

C. Selection of Secure Elliptic Curve Parameters

The Integer Factorization Problem (IFP), i.e., the computational intractability of determining, in polynomial time, the factors of a product of two large prime numbers, is the cornerstone of many cryptographic schemes, such as the Paillier's scheme and RSA.

The Elliptic Curve Discrete Logarithm Problem (ECDLP), i.e., the computational intractability of determining, in polynomial time, the value of k assuming two points, P and Q in an elliptic curve, where $Q = k \cdot P$. In this section, we show that security assumptions of iKUP are the ECDLP and the existence of a secure hash function.

Different from protocols based on IFP that normally need random numbers and safe primes, ECC is very sensitive to the selection of the parameters. The key size reduction is based on the assumption that we cannot solve ECDLP with complexity sub-exponential as IFP is. However, Menezes et al. [15] presented an algorithm with complexity sub-exponential for supersingular elliptic curves and Smart [16] presented an algorithm with complexity polynomial for prime-field-anomalous elliptic curves. We say that a curve Ω is supersingular over a finite field $\mathbb{Z}_p$ iff the trace of Frobenius $\mathfrak{t}$ is zero $\mathfrak{t} \equiv 0 \mod p$. Its value is defined by Hasse's theorem which determines a range for the number of points $|\Omega(\mathbb{F}_q)| = q + 1 - \mathfrak{t}$, where $|\mathfrak{t}| \leqslant 2\sqrt{q}$. We say that Ω is prime-field-anomalous iff $\mathfrak{t} = 1$, thus $|\Omega(\mathbb{F}_p)| = p$. The order of a curve $|\Omega(\mathbb{F}_p)|$ can be efficiently determined in logarithmic time by Schoof's algorithm [17]. Basically, prime-field-anomalous and supersingular curves should be avoided.

In Sec. IV, the function H_Ω outputs the point R_j, which is represented by an ordered pair $R_j = (x, y)$. The security assumption of H_Ω is that a secure hash function H outputs $r = H(j)$. Thereafter, we search for the first value equal or bigger than r that is part of a valid point, and then calculate y. The calculation depends on the curve, e.g., in curves over $\mathbb{Z}_p$ such that $p \equiv 3 \mod 4$, we search for x that satisfies $x^{\frac{p+1}{2}} \equiv 1 \mod p$, and determine $y = x^{\frac{p+1}{4}} \mod p$.

In Eq. (5), we have the scalar multiplication $k_i \cdot R_j$, where k_i is determined randomly and represents the permanent key of the meter i. As k_i is large enough, an adversary cannot find the value of k_i using a brute force attack. The value of z in $k_i \cdot R_j = z \cdot k_i \cdot R_{j+t}$ for $t > 0$ cannot be determined, due to the assumptions that ECDLP is intractable and that H is a safe hash function. Therefore, an adversary cannot relate commitments to the discovery of the measurement m_i. The function H_Ω provides us a pseudo-random point on the curve, and its addition or scalar multiplication provides a pseudo-random result.

VI. Evaluation through Simulation using a Real-World Data Set

In this section, we present the performance evaluation of iKUP using as input a real-world data set. We compare the performance of iKUP with Paillier based protocols and LOP, a DC-Net based protocol. The results show that iKUP has a worse performance in aggregation and decryption operations but heavily outperforms other protocols in encryption and commitments.

Encryption and commitments have the highest impact on the overall protocol performance, as they are the most frequent cryptographic functions in all protocols included in our evaluation and run on the smart meters, which have less processing power than the utility. Aggregation and decryption occur on the utility side and only once per round of measurements and, thus, have a lower impact in the total protocol performance than encryption and commitments.

The data set used in our evaluation is from the Irish Social Science Data Archive. It has $157,992,996$ measurements collected from $6,435$ meters, in 30-minute intervals for almost one and half years, in Ireland.

A. Integrity of the Real-World Data Set

We verified the integrity of the data set for errors before using it as input in our simulation. Although iKUP would be able to detect errors in the data set in our experiments, we detected and amended them before using the data set in our performance evaluation. The presence of errors in real-world data set shows the need of security protocols for the Smart

Grid that can detect and eliminate errors from measurement reports. Errors detected included, e.g., more than one measurement per round, missing measurements, and inconsistent timestamps. We excluded two hours of measurements reported in the end of the daylight savings time, when clocks are adjusted backward in the autumn.

DC-Nets require the participation of all users. Therefore, the erroneous measurements were replaced with a null (zero) value. After the amendment, the resulting data set used in our evaluation has $165,546,810$ measurements from $6,435$ meters and $25,726$ rounds, structured in a two-dimensional array.

B. Implementation of the Core Algorithms

We implemented iKUP's cryptographic functions Enc, Dec, hash H_Ω, Commit, and Open, the encryption and decryption functions of LOP, and the encryption, aggregation, and decryption of the Paillier's scheme.

The implementations is written in C and compiled with GCC, version 4.4.6, for GNU Linux, on a Red Hat Linux server. It is linked with GNU Multiple Precision Arithmetic Library (GMP), Open Multi Processing (OpenMP) and Open Secure Sockets Layer (OpenSSL) libraries. GMP is used for handling large integer numbers. OpenMP is used to parallelize the commitment and encryption algorithms in eleven threads. OpenSSL allows us access to the SHA–256 hash function, which is part of the iKUP-Hash routine.

The testing platform is a Linux server with twelve 2.00 GHz Intel® Xeon® E5–2620 recognized cores and 70 GB of shared RAM. All measurements were collected with microsecond (μs) precision.

C. Simulation Parameters

In iKUP, each meter first selects a 190-bit long pseudo-random number. The round number is represented as a timestamp in the data set. It is the input parameter of the SHA–256 for computing $H_\Omega(j)$, in Eq. (2). We used the P-192 curve, which is recommended by NIST in FIPS 186-2. The P-192 curve is over $\mathbb{Z}_p$ with $p = 2^{192} - 2^{64} - 1$ and is defined by

$$\Omega : y^2 \equiv x^3 - 3x + b \pmod{p},$$

where the hexadecimal values of b, P, and n are included in the appendix. The point $P = (x, y)$ is the recommended base point and its order is a prime given by n. The cofactor $h = \frac{\Omega(\mathbb{Z}_p)}{n} = 1$ and the elements of the group have order 1 or n. Since n is prime, the order of $R_j = H_\Omega(j)$ can be efficiently calculated, because the order of R_j is either 1 or n.

In LOP, each cryptographic key is 32-bit long [6]. MD5 was chosen as LOP's core hash function. MD5 is faster than SHA–256, but it is vulnerable to collision and preimage attacks. The MD5 vulnerabilities have no influence in the results of our performance evaluation.

The Paillier's scheme is implemented with two 512-bit long pseudo-random primes. Their product is 1024-bit long, which provides a security strength equivalent to the NIST P-192 curve according to the NIST Recommendation for Key Management–Part 1: Revision 3 (NIST Special Publication 800-57). The authors, however, do not vouch for the security of the P-192 curve. The selection of curve parameters for obtaining a higher level of security is discussed in Sec. V-C.

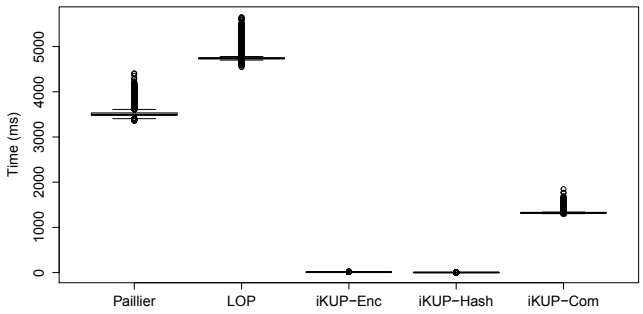

Figure 2. The boxplot shows the results of the processing time performance in ms of the encryption functions of Paillier's scheme, LOP, iKUP–Enc (Enc), iKUP-Hash (H_Ω), and iKUP-Com (Commit). All functions run on the smart meters and are all parallelized with the exception of iKUP-Hash.

D. Simulation Results

In this section, we present our simulations results. They are ordered as follows. First, we present the results for the cryptographic functions that are required to run on the smart meters in the evaluated protocols, i.e., encryption in iKUP (Enc), Paillier and LOP, and the iKUP's commitment generation (Commit) and hash function (H_Ω). Second, we show the simulation results for the data consolidation, which is a function that runs on the utility side in the Paillier's scheme and on the smart meter side in iKUP's case. Third, we present our simulations results concerning cryptographic functions that run on the utility side, i.e., decryption in iKUP (Dec), Paillier and LOP, and the iKUP's open commitment (Open) and H_Ω.

The input parameter for all simulations was the amended real-world data set. The programming language R was used to generate the boxplots in Fig. 2, 3, 4, and 5.

The simulation results for the encryption functions of Paillier's scheme, LOP and iKUP Enc, and iKUP's commitment Commit and hash function H_Ω, are presented in Fig. 2. All these operations run on the smart meters. Averages of the obtained results are: Paillier $= 3,514$ ms, LOP $= 4,755$ ms, Enc $= 12$ ms, $H_\Omega = 2$ ms, and Commit $= 1,322$ ms.

The cryptographic functions running on the smart meters are part of the process of consolidating measurements for Paillier's scheme, LOP, and iKUP. This process in Paillier and iKUP require an aggregation step, which runs on the utility in the Paillier's scheme, and on meters and on the utility for iKUP, i.e., the sum of encrypted measurements on meters, in Eq. (4), and the sum of commitments on the utility, in Eq. (7).

The simulation results for the aggregation step are depicted in Fig. 3. Averages of the obtained results are: Paillier $=$ 36.88 ms, iKUP's sum of encrypted measurements $= 0.03$ ms (on the meters), and iKUP's sum of commitments $= 45.25$ ms (on the utility).

The encryption and aggregation steps of iKUP allow it to provide the verification of billing information for both users and utilities. For the Paillier's scheme and LOP to provide this verification, they would need to run an additional protocol, e.g., [14], which adds an equivalent to the half of processing time of the Paillier's scheme to the total time. Therefore, for integrating verification of billing information in the Paillier's scheme or LOP, the total processing time in a meter would range from 5.3 to 6.5 s, while iKUP's is around 1.3 s.

The simulation results for the decryption step are depicted in Fig. 4. All these operations run on the utility. Averages of the obtained results are: Paillier $= 4,707\,\mu$s, LOP $= 35\,\mu$s, Dec $= 8,406\,\mu$s, $H_\Omega = 2,399\,\mu$s, and Open $= 2,336\,\mu$s.

Our simulation results show that iKUP is faster than Paillier's scheme and LOP for encryption, which happens once per meter per round and runs on the smart meters. iKUP is slower than Paillier's scheme and LOP in decryption, which happens only once per round and runs on the utility. Therefore, iKUP has a higher overall performance.

E. Recovering the Consolidated Measurements or Billing

The utility can recover the consolidated measurements $\mathfrak{m}_j$ from commitments in cases it does not receive the sum of the encrypted measurements $\mathfrak{c}_j$ from the meters. Assuming that the utility has received all the commitments and, thus, has computed $\mathfrak{C}_j$, the utility would need to execute a brute-force attack on $\mathfrak{C}_j$ to recover $\mathfrak{m}_j$. In this section, we evaluate computational effort for recovering $\mathfrak{m}_j$.

All the consolidated measurements obtained from the real-world data set used are smaller than 2^{24}, as presented in Fig. 5. Thus, we can encode every consolidated measurement from our data set in 24 bits.

The search for $\mathfrak{m}_j$ using a brute-force attack was simulated and its results are plotted in Fig. 6. Using our testing platform and without precomputation, $\mathfrak{m}_j$ can be recovered in ca. 11 s. It is possible to reduce substantially the processing time by parallelizing the search massively, e.g., with Graphics Processing Units (GPU) or High Performance Computers (HPC), which would allow many threads to run simultaneously. For much larger $\mathfrak{m}_j$, e.g., 2^{192}, the ECDLP is intractable.

VII. Discussion

iKUP is unique in allowing the consolidation of measurements and the verification of the billing information in a single protocol suite. We have also shown in Sec. VI that it is efficient by comparing it against the LOP and schemes based on Paillier. In this section, we discuss the cryptographic foundations of iKUP and their impact on its performance and the predicted performance impact of using longer cryptographic keys. Moreover, we analytically compare iKUP with other privacy-enhancing protocols for the Smart Grid and show how iKUP can be used to detect faulty meters in $O(\log |\mathcal{M}|)$.

A. Cryptographic Foundations and Performance

Protocols that are based on the Paillier's scheme [9], [13], [18] need a public key on the smart meters while iKUP requires every meter has two symmetric keys. iKUP and all other protocols also need a public–private key pair to sign their measurements to protect their integrity. In iKUP, the meters execute an in-network data aggregation to sum measurements, which is an approach also used in [9], [13], [18].

Most privacy-enhancing protocols for the Smart Grid are based on the IFP. iKUP is, among few others, based on the ECDLP. The feasibility of running elliptic curve protocols in smart meters was shown in [19].

The most complex operation in iKUP is the scalar multiplication, which is faster than IFP's modular exponentiation, considering similar levels of security. With the exception of iKUP, all other protocols based on the ECDLP [13], [18] also uses bilinear structures, which require longer keys, as discrete logarithms in bilinear structures are easier to solve than the ECDLP without bilinear structures [4], i.e., consider a nondegenerate bilinear map $\Gamma : G_1 \times G_2 \to G_3$ computable in polynomial time, where G_1, G_2, and G_3 are groups with G_1 and G_2 having prime order p. Thus, Γ is a bilinear structure and, therefore, the discrete logarithm in G_1 and G_2 is as secure as in G_3 that needs only $\log_2(p)$ elements. Therefore, protocols with bilinear structures do not benefit from the performance advantage offered by ECDLP, which iKUP does.

The processing time of scalar multiplications and modular exponentiations depends on the number of bits of the scalars

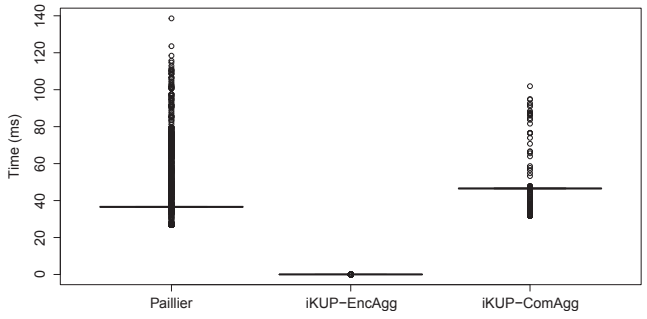

Figure 3. The boxplot shows the results of the processing time performance in ms of the aggregation for Paillier's scheme on the smart meters and iKUP (iKUP-EncAgg for encrypted measurement on the smart meters and iKUP-ComAgg for commitments on the utility).

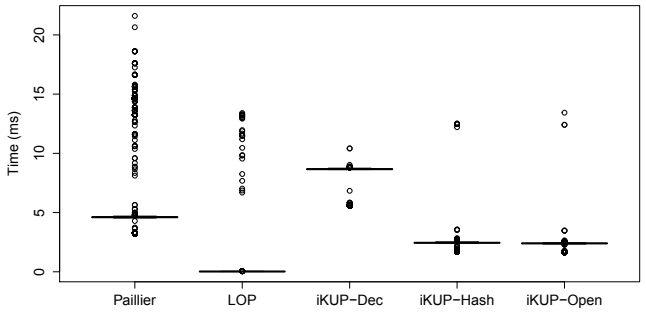

Figure 4. The boxplot shows the results of the processing time performance in ms of the decryption functions of Paillier's scheme, LOP and iKUP-Dec (Dec), iKUP-Hash (H_Ω), and iKUP-Open (Open). All functions run on the utility.

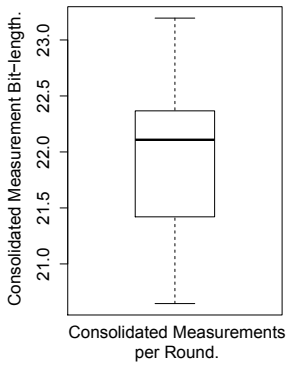

Figure 5. The boxplot shows the range in bits of the consolidated measurements $\mathfrak{m}_j$, i.e., the logarithm $\mathfrak{m}_j$ to the base 2.

and of the exponents, respectively. In our simulation, we used parameters that offer a similar level of security for all protocols involved, i.e., 80 bits of security. Had we selected a higher level of security, e.g., 128 bits of security, the difference of performance in favor of iKUP would also increase.

B. Comparison with Other Privacy-Enhancing Protocols

A function similar to Commit, in Eq. (5), is used in [20], but the protocol in [20] works with integers instead of ECC, and its goal is to return the consolidated measurements instead of the verification. Thus, the protocol in [20] is not scalable.

A comparison of the characteristics of the most relevant privacy-enhancing protocols for the Smart Grid, in the context of our work, is presented in Table I. In the first column of the table, we identify the protocols and their primitives in the second one, i.e., if they are based on the IFP or the ECDLP, and if they encrypt (Enc) or commit (Commit) measurements. The third column (Setup) indicates the complexity, with respect to the number of messages, to distribute the cryptographic keys, where $|\mathcal{M}|$ is the total of number meters. The same column

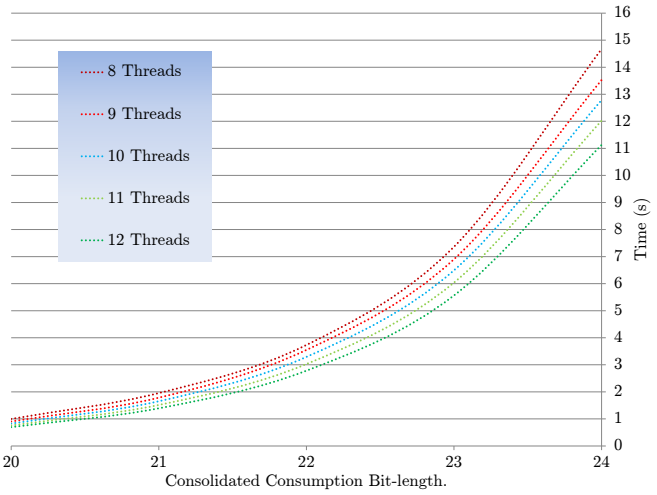

Figure 6. The amount of time needed to recover $\mathfrak{m}_j$ from $\mathfrak{C}_j$ with a brute-force attack, i.e., without knowing encrypted consolidated consumption $\mathfrak{c}_j$.

indicates if a TTP is used for key distribution, which means that its complexity cannot be assured. The fourth column (Keys) lists how many keys are stored in the meter, excluding the public–private key pair used for signing messages. In [14], there are no keys, with the exception of the signing key, but meters need to store the sum of random values generated during run time.

The fifth, sixth and seventh columns show, respectively, if the protocol allows the measurements to be consolidated, the consolidated measurements to be verified, and the billing to be verified.

A protocol that allows verification of consolidated consumption and billing using commitment scheme was proposed in [21], but it is not efficient because it requires groups four times larger than [8]. A protocol for verifying that the consolidated measurement equals the billing is presented in [22]. However, it allows meters to report arbitrary measurements that cannot be verified and there are no considerations regarding the expected consumption values. In addition, [22] performs worse than [21], as it needs to solve the discrete logarithm problem.

C. Additional Advantages of iKUP

iKUP can identify faulty meters in $O(\log |\mathcal{M}|)$ steps. If the utility is not able to open the commitment $\mathfrak{C}_j$, i.e., $\mathfrak{A} \neq \mathfrak{m}_j$, it can detect the meter contributed with an incorrect $m_{i,j}$ as follows. It assigns meters to subsets and requests them to rerun the consolidation of their measurements for round j within the subsets. The utility calculates the sum of the commitments for the subsets and verify which commitments it can open. Hence, the utility can identify in which subset or subsets the faulty meters are. By repeating the process and dividing the subsets even further, the utility is able to pinpoint the faulty meters. The number of divisions grows logarithmically with respect to the number of meters.

VIII. Conclusions

In this paper, we have introduced iKUP, an efficient and comprehensive privacy-preserving protocol for the Smart Grid that covers the entire cycle of power provisioning, consumption, billing, and verification. iKUP allows (*i*) utility providers to obtain a consolidated energy consumption value that relates to the consumption of a user set, (*ii*) utility providers to verify the correctness of this consolidated value, and (*iii*) the verification of the correctness of the billing information by both utility providers and users. iKUP prevents utilities from identifying individual contributions to the consolidated value and, therefore, enhances the users' privacy.

We evaluated the computational performance of iKUP with simulations using real-world data set with over 157 million measurements collected from 6,345 smart meters. We implemented and simulated two privacy-enhancing protocols, each based on a different security mechanism, the LOP and a general protocol based on the Paillier's scheme, and compared their results against iKUP. Our evaluation shows that its performance is worse for cryptographic operations that happen

Table I
COMPARISON WITH THE RELATED WORK.

Protocol	Primitive	Setup	Keys	Allows Consolidated Measurements	Verification of Consolidated Measurements	Billing Verification
[6]	DC-Net	$O(\|\mathcal{M}\|^2)$	$\|\mathcal{M}\|-1$	Yes	No	No
[7]	IFP - DC-Net	$O(\|\mathcal{M}\|^2)$	$\|\mathcal{M}\|-1$	Yes	No	No
[9]	IFP - Enc	$O(\|\mathcal{M}\|)$	1	Yes	No	No
[10]	IFP - Enc	TTP	1	Yes	No	No
[11]	IFP - Enc	$O(\|\mathcal{M}\|)$	1	Yes	No	No
[12]	IFP - Enc	TTP	1	Yes	No	No
[13]	ECDLP - Enc	$O(\|\mathcal{M}\|)$	1	Yes	Yes	No
[14]	IFP - Commit	n/a	n/a	No	No	Yes
[18]	ECDLP - Enc	$O(\|\mathcal{M}\|)$	1	Yes	Yes	No
[20]	IFP - Commit	TTP	1	Yes	No	No
[21]	IFP - Commit with matching Enc	$O(\|\mathcal{M}\|)$	1	Yes	Yes	Yes
[22]	ECDLP - Commit and Enc	$O(\|\mathcal{M}\|)$	1	Yes	Limited	Limited
iKUP	ECDLP - Commit and DC-Net - Enc	$O(\|\mathcal{M}\|)$	1	Yes	Yes	Yes

in the utility but iKUP heavily outperforms the others on encryption, which runs in the meters and has the highest impact on the overall protocol performance.

ACKNOWLEDGMENTS

The authors are thankful to the Horst-Görtz Foundation and SMARTSOCIETY, a project of the Seventh Framework Programme for Research of the European Community under grant agreement no. 600854, for funding this work, the Irish Social Science Data Archive and the Commission for Energy Regulation (CER), Electricity Customer Behaviour Trial, issued by The Research Perspective Ltd on the 12–03–2012.

APPENDIX

The value of b is "`0x 64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1`."

The base point $P = (x, y)$ is given by $x =$ "`0x 188DA80E B03090F6 7CBF20EB 43A18800 F4FF0AFD 82FF1012`" and $y =$ "`0x 07192B95 FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811`," and the order of P is given by $n =$ "`0x FFFFFFFF FFFFFFFF FFFFFFFF 99DEF836 146BC9B1 B4D22831`."

REFERENCES

[1] "Directive 2012/27/EU of the European Parliament and of the Council of 25 October 2012 on energy efficiency, amending directives 2009/125/EC and 2010/30/EU and repealing directives 2004/8/EC and 2006/32/EC," Official Journal L No.315, 25 Oct 2012.

[2] D. Chaum, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *J. of Cryptology*, vol. 1, pp. 65–75, 1988.

[3] M. Waidner, "Unconditional sender and recipient untraceability in spite of active attacks," in *Proc of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, ser. EUROCRYPT '89. Springer-Verlag New York, 1990, pp. 302–319.

[4] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography, Second Edition*, 2nd ed. Chapman & Hall/CRC, 2012.

[5] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '91. London, UK, UK: Springer-Verlag, 1992, pp. 129–140.

[6] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Privacy Enhancing Technologies*, ser. LNCS. Springer, 2011, vol. 6794, pp. 175–191.

[7] Z. Erkin and G. Tsudik, "Private computation of spatial and temporal power consumption with smart meters," in *ACNS*, ser. LNCS, vol. 7341. Springer, 2012, pp. 561–577.

[8] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT 1999*, ser. LNCS. Springer, 1999, vol. 1592, pp. 223–238.

[9] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Smart Grid Communications (SmartGridComm), 2010 1st IEEE Int Conf on*, Oct. 2010, pp. 327 –332.

[10] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *Parallel and Distributed Systems, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2012.

[11] S. Ruj and A. Nayak, "A decentralized security framework for data aggregation and access control in smart grids," *Smart Grid, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –10, 2013.

[12] M. Joye and B. Libert, Eds., *A Scalable Scheme for Privacy-Preserving Aggregation of Time-Series Data*. Springer, 2013.

[13] F. Li and B. Luo, "Preserving data integrity for smart grid data aggregation," in *IEEE Int Conf on Smart Grid Communications (SmartGridComm)*, 2012.

[14] M. Jawurek, M. Johns, and F. Kerschbaum, "Plug-in privacy for smart metering billing," in *Privacy Enhancing Technologies*, ser. LNCS. Springer, 2011, vol. 6794, pp. 192–210.

[15] A. Menezes, S. Vanstone, and T. Okamoto, "Reducing elliptic curve logarithms to logarithms in a finite field," in *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, ser. STOC '91. New York, NY, USA: ACM, 1991, pp. 80–89.

[16] N. P. Smart, "The discrete logarithm problem on elliptic curves of trace one," *J. Cryptology*, vol. 12, no. 3, pp. 193–196, 1999.

[17] R. Schoof, "Counting points on elliptic curves over finite fields," *Journal de théorie des nombres de Bordeaux*, vol. 7, no. 1, pp. 219–254, 1995. [Online]. Available: http://eudml.org/doc/247664

[18] L. Yang and F. Li, "Detecting false data injection in smart grid in-network aggregation," in *Smart Grid Communications (SmartGridComm), 2013 IEEE Int Conf on*, Oct 2013, pp. 408–413.

[19] A. Molina-Markham, G. Danezis, K. Fu, P. J. Shenoy, and D. E. Irwin, "Designing privacy-preserving smart meters with low-cost microcontrollers," in *Financial Cryptography*, ser. LNCS, vol. 7397. Springer, 2012, pp. 239–253.

[20] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *NDSS*. The Internet Society, 2011.

[21] F. Borges, D. Demirel, L. Böck, J. Buchmann, and M. Mühlhäuser, "A Privacy-Enhancing protocol that provides In-Network data aggregation and verifiable smart meter billing," in *19th IEEE Symp. on Computers and Communications (IEEE ISCC 2014)*, Madeira, Portugal, Jun. 2014.

[22] K. Ohara, Y. Sakai, F. Yoshida, M. Iwamoto, and K. Ohta, "Privacy-preserving smart metering with verifiability for both billing and energy management," in *Proc. of the 2nd ACM Workshop on ASIA Public-key Cryptography*, ser. ASIAPKC'14. New York, NY, USA: ACM, 2014.