DEP2SA: A Decentralised Efficient Privacy-Preserving and Selective Aggregation Scheme in Advanced Metering Infrastructure

Mustafa A. Mustafa, *Student Member, IEEE,* Ning Zhang, Georgios Kalogridis, *Member, IEEE,* and Zhong Fan, *Member, IEEE*

***Abstract*—This paper proposes a novel solution, called a decentralised, efficient, privacy-preserving and selective aggregation (DEP2SA) scheme, designed to support secure and user privacy-preserving data collection in the advanced metering infrastructure. DEP2SA is more efficient and applicable in real-life deployment, as compared with the state of the art, by adopting and adapting a number of key technologies: (1) it uses a multi-recipient system model, making it more applicable to a liberalised electricity market; (2) it uses the homomorphic Paillier encryption and selective aggregation methods to protect users' consumption data against both external and internal attacks, thus making it more secure; (3) it aggregates data at the gateways that are closest to the data originator, thus saving bandwidth and reducing the risk of creating a performance bottleneck in the system; and (4) it uses short signature and batch signature verification methods to further reduce computational and communication overheads imposed on aggregating nodes. The scheme has been analysed in terms of security, computational and communication overheads, and the results show that it is more secure, efficient and scalable than related schemes.**

***Index Terms*—Smart grid, AMI, security, homomorphic encryption, privacy preserving, selective aggregation, data leakage.**

I. Introduction

SMART GRID (SG) is a next generation electrical grid that, as shown in Fig. 1, supports two-way electricity flows and communications among grid entities [1]. One component of SG is the advanced metering infrastructure (AMI) that ensures communications for meter applications. AMI may also connect users with other entities via users' smart meters (SMs) using a hierarchical network structure consisting of building area networks (BANs), neighbourhood area networks (NANs) and wide area networks (WANs).

One anticipated application of AMI is the automated meter reading [1], in which each SM measures its user's electricity consumption data (CD) during a short time slot and sends the CD to authorised entities. Having access to users' CDs for each time slot will allow grid operators manage the grid more efficiently and suppliers forecast their customers' demand for electricity more accurately. As a result, the grid's reliability and efficiency can be improved. The more fine-grained the CDs sent to entities are, the more the SG reliability and efficiency may be improved (without considering the extra costs incurred as the result of the additional processing/communication).

However, uncontrolled access to fine-grained CDs may put users' privacy at risk. Entities that have access to CDs may, for example, use non-intrusive load monitoring (NILM) techniques [2] to build individual users' electricity consumption patterns, breaching users' privacy. The more fine-grained the CDs, the greater the risks, as far as the users' privacy is concerned. Thus, it is important to protect users' CDs from unauthorised access while collecting the data.

One way to achieve this is to aggregate users' CDs for each time slot before making the data available to authorised entities, assuming that the aggregated CD (ACD) obtained in each slot provides sufficient information to the entities. Also, intermediate nodes that aggregate the data should not be allowed to access the CDs. This can be achieved by using a homomorphic encryption technique [3]. Such a technique allows intermediate nodes to perform a specific linear algebraic operation on ciphertexts, which is equivalent to a different operation conducted on the corresponding plaintexts.

There are schemes [4]–[15] published in literature, which are designed to secure data aggregations and collections, but they assume that there is only a single recipient of ACD of all the users. In other words, these schemes are designed based on a single-recipient system model. However, in a liberalised electricity market (which is deployed in most European countries) there are multiple entities (e.g., grid operators, suppliers) that are authorised to access ACDs of different sets of users for legitimate purposes, and these access should be granted in conformance to the least privilege principle. Clearly, the existing schemes are not designed for a liberalised electricity market, and more work is necessary to allow a migration from the single-recipient system model to a multi-recipient one.

One naïve approach to realise this migration is to allow a 'prime' authorised entity (one which obtains the aggregated data in a single-recipient system model) to share the aggregated data with other 'secondary' authorised entities. However, this approach has two main drawbacks: (1) the prime entity knows the aggregated data of each subset of users requested by each of the secondary entities (and this might not be desirable in a liberalised market) and (2) the secondary entities can not

M. A. Mustafa is with the Department of Electrical Engineering, ESAT/COSIC, KU Leuven, B-3001 Leuven-Heverlee, Belgium, and also with iMinds, Belgium, e-mail: (mustafa.mustafa@esat.kuleuven.be).

N. Zhang is with the School of Computer Science, The University of Manchester, Manchester, M13 9PL, UK, e-mail: (ning.zhang@manchester.ac.uk).

G. Kalogridis and Z. Fan are with Toshiba Research Europe Limited, Telecommunications Research Laboratory, Bristol, BS1 4ND, UK, e-mail: ({george, zhong.fan}@toshiba-trel.com).

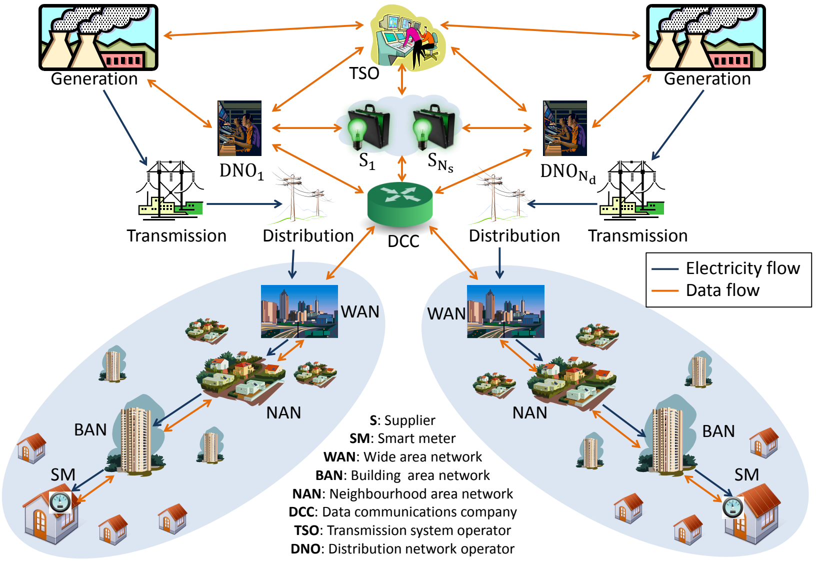


Fig. 1. A conceptual architecture of SG.

verify the correctness of the aggregated data, i.e., they can not verify that the data they receive from the prime entity is in fact the aggregated data of the requested subset of users.

One way to overcome these drawbacks would be to apply one of the existing solutions multiple times, i.e., to encrypt a user's CD multiple times, each time using a different authorised entity's homomorphic public key, generating multiple ciphertexts, one for each of the authorised entities. Then, the ciphertexts from different users that are intended for the same entity are aggregated and the result is sent to that entity. Thereby, all the authorised entities will only receive the ACD of the users under their managements. However, this naïve solution is not efficient as each SM will need to encrypt the same data multiple times to generate multiple ciphertexts. Therefore, there is a need for a new aggregation method that could serve the multi-recipient system model with less computational and communication overheads. In addition, considering the large number of SMs anticipated, having a single entity to perform the aggregation would place excessive computational burden on it, thus making the entity a potential performance bottleneck and an easy target for attacks. Hence, it is desirable to distribute the computational load of data aggregation across multiple entities, and the selection of these entities should be such that any additional communication costs introduced are minimal.

This paper proposes such a novel solution called a decentralised, efficient and privacy-preserving selective aggregation (DEP2SA) scheme. DEP2SA supports aggregation of CDs in respective users' suppliers and locations, so authorised entities can only get the fine-grained ACDs relevant to, and necessary for, their respective business dealings. In this way, users' privacy can better be preserved. Furthermore, DEP2SA allows grid operators (prime entities) to share their respective aggregated data with suppliers (the secondary entities) in such a way that the suppliers could verify the correctness of the received data with the assistance of a trusted entity. This work extends our previous research [16] in improving the aggregation method and proposing a method to quantify the level of private data leakage from ACDs. The main contributions of the paper are fourfold.

- First, it introduces a multi-recipient system model which is suited to liberalised electricity markets and a well-studied cyber threat model, and it specifies a set of functional and security requirements for the AMI.
- Secondly, it proposes a novel scheme (i.e., DEP2SA) that supports a selective and secure delivery of ACDs to respective multiple authorised recipients based on the need-to-know and least privilege principles. In comparison with related schemes, DEP2SA imposes less computational and communication overheads, while achieving privacy-preserving CD collection and distribution.
- Thirdly, it analyses ACDs of varying numbers of users and proposes a simple method to quantify the level of private data leakage from these ACDs. This method can be used to determine the minimum number of users whose CDs should be aggregated to ensure a given level of privacy preservation.
- Fourthly, it compares DEP2SA to two recent, most relevant work: EPPA [8] and a scheme [9] that also aggregates the data in a decentralised manner. The comparison results demonstrate that DEP2SA is more efficient, in terms of computational and communication costs, than these schemes.

The remainder of this paper is organised as follows. Section II discusses the related work. Sections, III and IV, respectively, present design preliminaries and main building blocks used in the design of DEP2SA. Section V describes the DEP2SA scheme in detail, which is followed by its security analysis in Section VI, users' private data leakage analysis in Section VII, and DEP2SA performance evaluation in Section VIII. Finally, Section IX concludes the paper. Table I lists the acronyms used in the paper.

TABLE I
ACRONYMS

AMI	advanced metering infrastructure	SG	smart grid
DCC	data communications company	SM	smart meter
DNO	distribution network operator	ACD	aggregated CD
TSO	transmission system operator	ECD	encrypted CD
CD	electricity consumption data	AECD	aggregated ECD
BAN	building area network	GW	gateway
NAN	neighbourhood area network	BG	BAN GW
WAN	wide area network	NG	NAN GW
TA	trusted authority	WG	WAN GW

II. Related Work

The importance of securing the SG and preserving users' privacy is well recognised by standardisation bodies, e.g., NIST [17], IETF [18] and ETSI [19], and the research community [20]–[26]. A number of efforts and proposals have been made to strengthen the protection. For example, Efthymiou et al. [27] proposed a method for anonymising users' fine-grained CDs sent by SMs, so authorised entities cannot link the received CDs to their originators. Lin et al. [28] proposed a system to allow users' CDs to be accessed at multiple time granularities, each identified by a random number. A random number is added to a user's fine-grained CD and the result is sent to, and stored in, a central database. The user can govern the time granularity at which her/his CD can be accessed by providing the corresponding random number(s) to data requesters. Mármol et al. [29] proposed a protocol to allow users to report their CDs to a supplier in a privacy-preserving manner. With this proposal, each user's SM encrypts the user's CD with a unique encryption key and then sends the encrypted CD (ECD) and the key to the supplier and a key aggregator, respectively. The key aggregator aggregates all the encryption keys received and sends the resulting key to the supplier. The supplier, then, aggregates all the received ECDs and uses the aggregated key to recover the ACD. Although the solutions proposed in [27]–[29] can preserve users' privacy, they are not scalable. As the number of users increases, the computational and communication overheads in the entire grid increase linearly.

Clearly, communicating ACDs instead of CDs helps to reduce communication overheads and preserve users' privacy. Our discussions here focus on privacy preservation through the use of homomorphic encryption. Li et al. [5] proposed an in-network aggregation scheme that uses SMs to aggregate users' ECDs en route for an authorised entity. The scheme achieves a good level of scalability. However, it only protects users' CDs against passive attacks. Deng et al. [6] overcame this limitation by proposing to digitally sign each ECD. Li et al. [7] improved [6] in terms of reducing overhead costs by using the Boneh-Lynn-Shacham (BLS) signature scheme that allows the batch verification of multiple signatures. They also introduced an incremental verification technique that allows the collector node to identify SMs feeding fake CDs. Li et al. [30] proposed an efficient and fault-diagnosable authentication architecture for AMI, which is also based on the BLS signature scheme. To further reduce overheads, Lu et al. [8] proposed a scheme which packs user's multidimensional CDs into a single ECD, whereas Ruj et al. [9] proposed a decentralised aggregation method, in which data are aggregated at local gateways en route for a central entity. A review and comparison of a number of aggregation schemes can be found in [14].

These existing solutions are designed for a single-recipient system model where one entity (per region) is assumed to do both, manage the grid and supply all the users (within the region) with electricity. These solutions may not be secure and efficient when being applied to a liberalised market, e.g., the UK market [31], where, to allow competition, grid management and electricity supply are done by different entities, and within one region, more than one entity may supply the users with electricity. To support this multi-entity model, Rottondi et al. [32] proposed an architecture containing additional functional entities, called privacy preserving nodes (PPNs). Each SM splits its user's CD into shares using a secret sharing scheme and sends these shares to different PPNs. PPNs perform aggregation of different sets of the shares based on the CDs' intended recipients. This solution has two drawbacks. First, it introduces the additional entities of PPNs, and this increases the SG complexity. Secondly, it employs a secret sharing scheme that requires the distribution of shares, and this increases communication overheads. The first drawback can be overcome by allocating the tasks of PPNs to existing SG nodes (e.g., gateways) [33]. However, the second drawback still remains. In addition, those existing schemes, which employ a homomorphic cryptosystem to protect the confidentiality of the collected data, have not considered any security threats imposed by authorised insiders, such as eavesdropping attacks by authorised entities. If an authorised entity could eavesdrop a user's ECD prior to data aggregation, it can recover the user's CD. To address these limitations, we here propose a novel data aggregation and collection solution, i.e., DEP2SA. DEP2SA is particularly designed for a multi-recipient system model and achieves security, privacy-preservation, efficiency and scalability.

It should be mentioned that the decentralised and selective aggregation method has been previously published in [16]. However, this paper extends the aggregation method to allow aggregating nodes to (i) detect and discard data coming from malfunctioning/malicious SMs, (ii) report such SMs to grid operators, and (iii) aggregate only the authentic data sent by legitimate SMs. The MUSP system [34] has also used the same aggregation method as in [16]. However, MUSP does not extend the aggregation method, rather it combines this method with other techniques to support additional services such as user billing and supplier and/or account holder switching.

III. Preliminaries

This section details the system and threat model, assumptions, notations and requirements used in the design of the DEP2SA scheme.

A. System Model

The system model, as shown in Fig. 2, is adapted to the UK's liberalised market [31] and consists of the following entities:

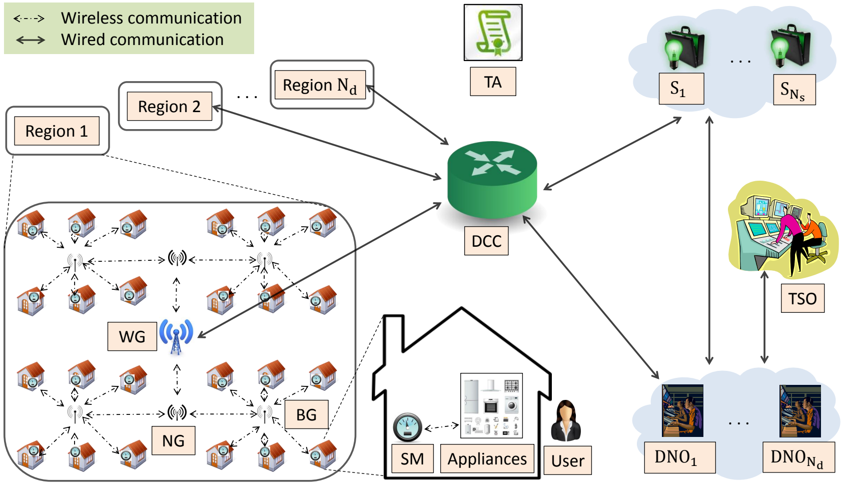


Fig. 2. A multi-recipient system model (architecture) used in the design of DEP2SA.

- Trusted authority (TA): Trusted entity that regulates electricity markets, e.g., in UK this is Ofgem [35].
- Grid operators: There is one transmission system operator (TSO) responsible for balancing the entire grid and $\mathrm{N_d}$ distribution network operators (DNOs) each responsible for maintaining the distribution network in a particular region and charging suppliers distribution network fees based on the CDs of the suppliers' customers in this region.
- Suppliers: There are $\mathrm{N_s}$ suppliers each responsible for supplying the electricity to its customers who may be located in different regions across the grid.
- User: A customer who demands, consumes and pays his/her supplier for the electricity consumed.
- Smart meter (SM): Advanced metering device that measures its user's CD on a per time slot, t_n, basis.
- Data communications company (DCC): A third party entity that is responsible for collecting and communicating users' data to authorised SG entities [36].
- Networking facility: It connects users' SMs to the DCC via a hierarchical network structure [21] consisting of BANs, NANs and WANs. Each BAN, NAN and WAN has a gateway (GW), i.e., a BAN GW (BG), an NAN GW (NG) and a WAN GW (WG). A higher level GW collects data received from a number of gateways at the level immediate below it. A GW at the lowest level, i.e., BG, collects data received from SMs that are connected to the GW. The DCC collects data from WGs.

B. Threat Model

The threat model used in the DEP2SA design is as follows:

- Users are untrustworthy and curious. They may try to modify CDs sent by their SMs in attempt to gain financial advantage and/or learn other users' CDs.
- DNOs are semi-trusted and curious. They report correct data to TSO (so TSO can keep the grid in balance) but they may manipulate data sent to suppliers in an attempt to gain financial advantage. They may also try to learn individual users' CDs and/or ACDs of any group of users located in other DNOs' regions.
- DCC is honest but curious. It follows protocol specifications but it may try to find out CDs of individual users and/or ACDs of any group of users. Also, the DCC is trusted by the authorised data recipients (i.e., TSO, DNOs and suppliers) to act as expected.
- Suppliers are suspicious and curious. They do not assume (or they do not trust) that DNOs would always charge them the right distribution network fees. They may also attempt to learn individual users' CDs and/or ACDs of any group of customers contracted by their competitors (i.e., other suppliers).
- External entities are untrustworthy or even malicious. They may eavesdrop data in transit trying to gain access to confidential data and/or modify the data in an attempt to disrupt the SG.

C. Assumptions

The following assumptions are used in the DEP2SA design:

- Each entity in the system model has a unique ID.
- SMs are tamper-proof and sealed. No one (including their users) could tamper with them without being detected.
- All entities are time synchronised.
- For the sake of simplicity, each BG collects data from $\mathrm{N_{sm}}$ (number of) SMs, each NG collects data from $\mathrm{N_{bg}}$ BGs and each WG from $\mathrm{N_{ng}}$ NGs. There are $\mathrm{N_{wg}}$ WGs in each region and $\mathrm{N^G_{wg}}$ WGs in the whole grid.

D. Notations

We denote the ith SM as $\mathrm{sm}_i \in \mathbb{SM}$, where $\mathbb{SM}$ is the set of all the SMs in the grid, and the CD during the nth time slot, t_n, measured by sm_i as $\mathrm{e}^{t_n}_{\mathrm{sm}_i} \in \mathbb{E}^{t_n}$, where $\mathbb{E}^{t_n}$ is the set of CDs during t_n measured by all the SMs (of all the users) in the grid. We denote the following subsets of $\mathbb{SM}$ and $\mathbb{E}^{t_n}$:

- $\mathbb{SM}_{\mathrm{d}_j} \subset \mathbb{SM}$ as the set of all the SMs operated by the jth DNO, d_j, (located in region j).

TABLE II
NOTATIONS

Symbol	Meaning
t_n	nth time slot, $n = \{1, \dots, \mathrm{N_t}\}$
d_j	jth DNO (operating in region j), $j = \{1, \dots, \mathrm{N_d}\}$
s_u	uth supplier, $u = \{1, \dots, \mathrm{N_s}\}$
sm_i	ith SM
$\mathbb{SM}$	set of all the SMs in the grid
$\mathbb{SM}_{\mathrm{d}_j}$	set of all the SMs operated by d_j
$\mathbb{SM}_{\mathrm{s}_u}$	set of all the SMs whose users are supplied by s_u
$\mathbb{SM}_{\mathrm{d}_j,\mathrm{s}_u}$	set of all the SMs operated by d_j and whose users are supplied by s_u
$\mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j}$	set of all the SMs connected to βth BG and operated by d_j
$\mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_u}$	set of all the SMs connected to bg_β, operated by d_j and whose users are supplied by s_u
$\mathrm{e}_{\mathrm{sm}_i}^{t_n}$	CD (of a user) during t_n measured by sm_i
$\mathbb{E}^{t_n}$, $\mathbb{E}_{\mathrm{d}_j}^{t_n}$, $\mathbb{E}_{\mathrm{s}_u}^{t_n}$	set of CDs during t_n measured by all the SMs belonging to $\mathbb{SM}$, $\mathbb{SM}_{\mathrm{d}_j}$, $\mathbb{SM}_{\mathrm{s}_u}$
$\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{t_n}$	set of CD during t_n measured by all the SMs belonging to $\mathbb{SM}_{\mathrm{d}_j,\mathrm{s}_u}$
ID_i, σ_i	identity of entity i, digital signature created by i
TS_i, TS^{t_n}	time stamp of entity i, of time slot t_n
x_i, y_i	secret, public key of entity i for signing, verifying
hpk_i, hsk_i	homomorphic public, private key pair of entity i
Cert_i	digital certificate of entity i
k_{bg_β}	key shared between bg_β and all its child SMs
$C_{\mathrm{sm}_i}^{t_n}$, c_{sm_i}	ciphertext of $\mathrm{e}_{\mathrm{sm}_i}^{t_n}$, generated by sm_i using k_{bg_β}
E_k, D_k	symmetric encryption, decryption using key k
Enc_{pk_i}	asymmetric encryption using entity i's public key
Dec_{sk_i}	asymmetric decryption using entity i's private key
$\mathrm{m^p}$, $\mathrm{m^r}$	message pending, received status of a SM
$\mathrm{m^a}$, $\mathrm{m^{agg}}$	message authentic, aggregated status of a SM
$\mathrm{N_{m^p}}$, $\mathrm{N_{m^r}}$	number of SMs with a status $\mathrm{m^p}$, $\mathrm{m^r}$
$\mathrm{N_{m^a}}$, $\mathrm{N_{m^{agg}}}$	number of SMs with a status $\mathrm{m^a}$, $\mathrm{m^{agg}}$
$\mathrm{N_{sm}}$	number of SMs connected to each BG
$\mathrm{N_{bg}}$	number of BGs connected to each NG
$\mathrm{N_{ng}}$	number of NGs connected to each WG
$\mathrm{N_{wg}}$, $\mathrm{N_{wg}^G}$	number of WGs in each DNO's region, in the grid

- $\mathbb{SM}_{\mathrm{s}_u} \subseteq \mathbb{SM}$ as the set of all the SMs whose users are supplied by the uth supplier, s_u.
- $\mathbb{SM}_{\mathrm{d}_j,\mathrm{s}_u} \subseteq \mathbb{SM}_{\mathrm{d}_j}$ and $\subseteq \mathbb{SM}_{\mathrm{s}_u}$ as the set of all the SMs operated by d_j and whose users are supplied by s_u.
- $\mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j} \subset \mathbb{SM}_{\mathrm{d}_j}$ as the set of all the SMs connected to βth BG, bg_β, and operated by d_j.
- $\mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_u} \subseteq \mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j}$ as the set of all the SMs connected to bg_β, operated by d_j and whose users are supplied by s_u.
- $\mathbb{E}_{\mathrm{d}_j}^{t_n}$, $\mathbb{E}_{\mathrm{s}_u}^{t_n}$ and $\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{t_n}$ as the sets of CDs during t_n measured by the SMs belonging to the sets $\mathbb{SM}_{\mathrm{d}_j}$, $\mathbb{SM}_{\mathrm{s}_u}$ and $\mathbb{SM}_{\mathrm{d}_j,\mathrm{s}_u}$, respectively.

Also, $\sum(X)$ denotes the aggregate (sum) value of all elements in the set X. More notations are given in Table II.

E. Design Requirements

The AMI application should satisfy the following functional and security requirements.

1) Functional Requirements:

(F1) Each malfunctioning and/or under-attack SM/GW should be identified as early as possible and reported to the regional DNO, so necessary actions can be taken.

(F2) At each t_n, each DNO, d_j, should be able to access

a) $\sum(\mathbb{E}_{\mathrm{d}_j}^{t_n})$, so it can better manage the distribution network in its region, and

b) $\sum(\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{t_n})$ for $u = \{1, \dots, \mathrm{N_s}\}$, so it can split distribution network fees fairly among suppliers.

(F3) At each t_n, each supplier, s_u, should be able to access

a) $\sum(\mathbb{E}_{\mathrm{s}_u}^{t_n})$, so it can predict its customers' demand accurately to avoid imbalance fines, and

b) $\sum(\mathbb{E}_{\mathrm{d}_j,\mathrm{s}_u}^{t_n})$ for $j = \{1, \dots, \mathrm{N_d}\}$, so it can be assured that it pays the correct distribution network fee to each DNO, i.e., it is not over(under)charged.

(F4) At each t_n, the TSO should be able to access

a) all $\sum(\mathbb{E}_{\mathrm{d}_j}^{t_n})$ for $j = \{1, \dots, \mathrm{N_d}\}$, and

b) $\sum(\mathbb{E}^{t_n})$, so it can balance the grid efficiently.

2) Security Requirements:

(S1) Message authenticity: The recipient should be assured that the message has not been altered during transit, is fresh and indeed from the claimed source.

(S2) Confidentiality of users' data: Users' fine-grained ACDs should only be accessed by authorised entities.

(S3) User privacy preservation: individual users' fine-grained CDs should not be revealed to any SG entity.

(S4) Authorisation: Entities should only be allowed to access the ACDs of their users, i.e., the users they operate in case of a DNO; the users they supply in case of a supplier.

(S5) Availability: Protocols should be designed such that they are resilient to denial-of-service (DoS) attacks.

IV. BUILDING BLOCKS

The bilinear pairing based BLS short signature scheme [37], the aggregate signature scheme [38] and the Paillier cryptosystem [3] are used as the building blocks in our design. This section reviews briefly these schemes.

A. Bilinear Pairing based Signature Schemes

The BLS short signature scheme [37] generates a signature with its length being only half of the size of a DSA signature for a similar level of security. It also allows the construction of an aggregate signature [38] from multiple signatures signed on different messages by different users and this aggregate signature can be batch verified.

Let (G, G_T, q, g, e, H) be the digital signature system parameters where G and G_T are two cyclic groups of the same prime order q, $g \in G$ is a generator, $e : G \times G \rightarrow G_T$ is a bilinear map (i.e., e is efficiently computable, $e(g, g) \neq 1$ and $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in Z$ [39]) and $H : \{0, 1\}^* \rightarrow G$ is a cryptographic hash function.

1) BLS Short Signature Scheme: The scheme comprises three algorithms: a key generation algorithm (KeyGen), a signature generation algorithm (SigGen) and a signature verification algorithm (SigVer).

- KeyGen: Select randomly $x \overset{\mathrm{R}}{\longleftarrow} Z_q$ and compute $y = g^x$. The secret key is $x \in Z_q$. The public key is $y \in G$.
- SigGen: Given a message $m \in \{0, 1\}^*$ and secret key x, compute the signature $\sigma = H(m)^x$, $\sigma \in G$.
- SigVer: Given the public key y, message m, and signature σ, accept if $e(g, \sigma) = e(y, H(m))$ holds.

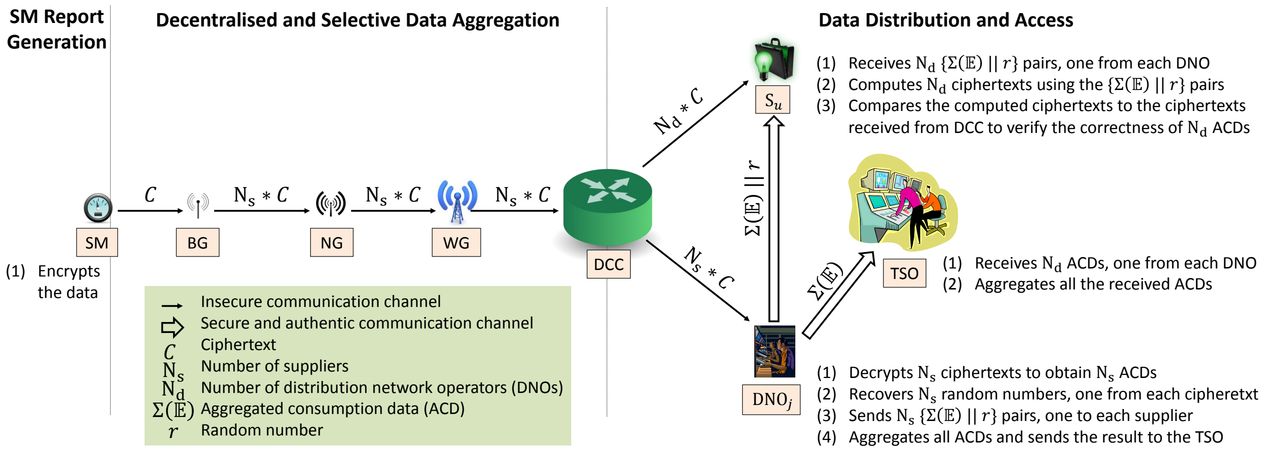

SM Report Generation
Decentralised and Selective Data Aggregation
Data Distribution and Access
C
$N_s * C$
$N_s * C$
$N_s * C$
$N_d * C$
$N_s * C$
$\Sigma(\mathbb{E}) \mid\mid r$
$\Sigma(\mathbb{E})$
SM
BG
NG
WG
DCC
S_u
TSO
DNO_j
(1) Encrypts the data
Insecure communication channel
Secure and authentic communication channel
C Ciphertext
N_s Number of suppliers
N_d Number of distribution network operators (DNOs)
$\Sigma(\mathbb{E})$ Aggregated consumption data (ACD)
r Random number
(1) Receives N_d $\{\Sigma(\mathbb{E}) \mid\mid r\}$ pairs, one from each DNO
(2) Computes N_d ciphertexts using the $\{\Sigma(\mathbb{E}) \mid\mid r\}$ pairs
(3) Compares the computed ciphertexts to the ciphertexts received from DCC to verify the correctness of N_d ACDs
(1) Receives N_d ACDs, one from each DNO
(2) Aggregates all the received ACDs
(1) Decrypts N_s ciphertexts to obtain N_s ACDs
(2) Recovers N_s random numbers, one from each cipheretxt
(3) Sends N_s $\{\Sigma(\mathbb{E}) \mid\mid r\}$ pairs, one to each supplier
(4) Aggregates all ACDs and sends the result to the TSO

Fig. 3. An overview of DEP2SA.

2) Aggregate Signature Scheme: The scheme comprises four algorithms: a key generation algorithm (KeyGen), a signature generation algorithm (SigGen), a signature aggregation algorithm (SigAgg) and a signature verification algorithm (SigVer).

- KeyGen & SigGen: These two algorithms are the same as the algorithms described in Section IV-A1. Suppose that there are n distinct users. Each user u_i, where $i = \{1, \ldots, n\}$, generates its secret key, x_i, and public key, y_i. Then, u_i signs its message m_i and obtains its signature, σ_i.
- SigAgg: An aggregated signature is computed by multiplying individual signatures, $\sigma^{agg} = \prod_{i=1}^{n} \sigma_i$.
- SigVer: Given n users' public keys, $y_1, \ldots, y_n$, their messages, $m_1, \ldots, m_n$, and the aggregate signature on the messages, σ^{agg}, compute $H(m_i)$. Accept if all the messages are distinct and $e(g, \sigma^{agg}) = \prod_{i=1}^{n} e(y_i, H(m_i))$ holds.

B. Paillier Cryptosystem

The Paillier cryptosystem [3] has an additive homomorphism property, and it is relatively efficient and semantically secure. It comprises three algorithms: a key generation algorithm (KeyGen), an encryption algorithm (Enc) and a decryption algorithm (Dec).

- KeyGen: Choose two large prime numbers (p_1, q_1). Calculate $n = p_1.q_1$, $\lambda = lcm\,(p_1 - 1, q_1 - 1)$. Define $L(u) = (u-1)/n$. Choose a generator $g \in Z^*_{n^2}$. Calculate $\mu = (L(g^\lambda \, mod\, n^2))^{-1} \, mod\, n$. The public key is $hpk = (n, g)$ and the private key is $hsk = (\lambda, \mu)$.
- Enc: Given a message $m \in Z_n$, choose a random number $r \in Z^*_n$. Compute the ciphertext $C = Enc(m) = g^m.r^n \, mod\, n^2$.
- Dec: Given the ciphertext $C \in Z^*_{n^2}$, recover the message $m = Dec(C) = L(C^\lambda \, mod\, n^2).\,\mu \, mod\, n$.

The Paillier cryptosystem has the following two properties:

- Additive homomorphism: Multiplying the ciphertexts of x messages results in a ciphertext of the sum of the messages, e.g.,

$$\begin{aligned} C(m_1).C(m_2) &= (g^{m_1}.r_1^n).(g^{m_2}.r_2^n) \, mod\, n^2 \\ &= g^{(m_1+m_2)}.(r_1.r_2)^n \, mod\, n^2 \quad (1) \\ &= C(m_1 + m_2). \end{aligned}$$

- Random number recovery: Given a message, m, its ciphertext C and the private key hsk, the random number r used in the encryption of m can be recovered by computing the following equation.

$$r = (Cg^{-m} \, mod\, n)^{n^{-1} \, mod\, \lambda} \, mod\, n. \qquad (2)$$

V. The DEP2SA Scheme

This section describes our novel data aggregation scheme, the DEP2SA scheme. Prior to the detailed description, we first give an overview of the scheme (which is also shown in Fig. 3) and outline the system initialisation process.

A. Overview of DEP2SA

Each DNO has a homomorphic private/public key pair, so each of them acts as an independent prime authorised entity. Each SM encrypts its data with the homomoprhic public key of its regional DNO and sends the ciphertext (attached with the ID of the DNO and the ID of its user's contracted supplier) to its local gateway. Each gateway aggregates the received ciphertexts based on the attached supplier ID and forwards the resulted ciphertexts to the next level gateway (DCC). The DCC aggregates the received ciphertexts based on the attached DNO ID and supplier ID, producing respective region-supplier-based ciphertexts. Then, the DCC forwards selections of these ciphretexts to their respective recipients, i.e. the corresponding DNOs and suppliers. In this way, each DNO/supplier only receives the aggregated ciphertexts attached with its own ID. It is worth noting that the suppliers can not decrypt these cipertexts; they use these ciphertexts for verification purposes.

Upon the receipt of the region-supplier-based ciphertexts, each DNO performs the following tasks: (1) decrypts the

TABLE III
KEYS AND CERTIFICATES OF ENTITIES

Entity	Secret Keys	Public Keys	Certificate
TA	x_{ta}	y_{ta}	$\text{Cert}_{\text{ta}} = \{\text{ID}_{\text{ta}}, y_{\text{ta}}\}$
TSO	x_{tso}, sk_{tso}	y_{tso}, pk_{tso}	$\text{Cert}_{\text{tso}} = \{\text{ID}_{\text{tso}}, y_{\text{tso}}, pk_{\text{tso}}\}$
DNO	x_{d_j}, sk_{d_j}, hsk_{d_j}	y_{d_j}, pk_{d_j}, hpk_{d_j}	$\text{Cert}_{\text{d}_j} = \{\text{ID}_{\text{d}_j}, y_{\text{d}_j}, pk_{\text{d}_j}, hpk_{\text{d}_j}\}$
Supplier	x_{s_u}, sk_{s_u}	y_{s_u}, pk_{s_u}	$\text{Cert}_{\text{s}_u} = \{\text{ID}_{\text{s}_u}, y_{\text{s}_u}, pk_{\text{s}_u}\}$
DCC	x_{dcc}	y_{dcc}	$\text{Cert}_{\text{dcc}} = \{\text{ID}_{\text{dcc}}, y_{\text{dcc}}\}$
BG	x_{bg_β}	y_{bg_β}	$\text{Cert}_{\text{bg}_\beta} = \{\text{ID}_{\text{bg}_\beta}, y_{\text{bg}_\beta}\}$
NG	x_{ng_η}	y_{ng_η}	$\text{Cert}_{\text{ng}_\eta} = \{\text{ID}_{\text{ng}_\eta}, y_{\text{ng}_\eta}\}$
WG	x_{wg_ω}	y_{wg_ω}	$\text{Cert}_{\text{wg}_\omega} = \{\text{ID}_{\text{wg}_\omega}, y_{\text{wg}_\omega}\}$
SM	x_{sm_i}	y_{sm_i}	$\text{Cert}_{\text{sm}_i} = \{\text{ID}_{\text{sm}_i}, y_{\text{sm}_i}\}$

ciphertexts to obtain the corresponding supplier-based ACDs of the users located in its region of operation, (2) uses the ACDs along with its homomorphic private key to recover the random number embedded in each of these ciphertexts, (3) sends each pair of the recovered ACD and the random number to their respective suppliers (secondary authorised entities), and (4) aggregates all of its recovered supplier-based ACDs and sends the resulted ACD to the TSO.

Upon the receipt of each ACD and random number pair from each of the DNOs, each supplier computes the ciphertexts using these ACDs, random numbers and the homomorphic public key of the respective DNOs, and then verifies the correctness of the received ACDs by comparing the computed ciphertexts to the ciphertexts received from the DCC. Finally, the TSO aggregates all the received ACDs.

It should be emphasized that, as the result of using the decentralised and selective data aggregation approach, DEP2SA can offer significant bandwidth savings compared to the centralised aggregation approach. For example, with the decentralised aggregation approach, each gateway sends a single message containing only N_s ciphertexts, where N_s is the number of suppliers in a liberalised electricity market. In contrast, with the centralised aggregation approach, the ciphertexts generated by every single SM will have to be sent to the central aggregating entity (the DCC).

B. System Initialisation

The system initialisation comprises three phases: (1) system parameters setup, (2) key generation and distribution and (3) SM/GW installation and key establishment.

1) System Parameters Setup: A trusted authority (TA) generates the system's parameters, (G, G_T, q, g, e), defines a hash function, H, selects a random number $x_{\text{ta}} \overset{\text{R}}{\leftarrow} Z_q$ and computes $y_{\text{ta}} = g^{x_{\text{ta}}}$. Here, $x_{\text{ta}} \in Z_q$ is the system's master secret key and $y_{\text{ta}} \in G$ is the system's master public key. TA keeps x_{ta} secret, but publishes all other system parameters, i.e., $\{G, G_T, q, g, e, y_{\text{ta}}, H\}$.

2) Key Generation and Distribution: This phase is divided into three steps outlined below.

Step 1 is executed during a license acquisition process:

- The DCC, TSO, DNOs and suppliers each generates a distinct BLS public/secret key pair, $\{y_i, x_i\}$, using the KeyGen algorithm described in Section IV-A1. These keys are used for data verification/signing.
- The TSO, DNOs and the suppliers each generates a distinct public/private key pair, $\{pk_i, sk_i\}$, using a standard public-key algorithm such as RSA. These keys are for data encryption and decryption.
- DNOs each generates a distinct homomorphic public/private key pair, $\{hpk_{\text{d}_j}, hsk_{\text{d}_j}\}$, using the KeyGen algorithm described in Section IV-B.
- The TA signs all the public keys generated by the DCC, TSO, DNOs and suppliers with its secret key, x_{ta}. These are done through the generation of a digital certificate for all such keys of each entity.

Step 2 is executed during SM manufacturing process:

- Each SM generates a distinct BLS public/secret key pair, $\{y_{\text{sm}_i}, x_{\text{sm}_i}\}$.
- The TA generates a digital certificate for the public key of each SM.
- Each SM is equipped with the digital certificate certifying its public key. The corresponding secret key is kept secret and tamper-proof.

Step 3 is executed during GW manufacturing process:

- Each GW generates a distinct BLS public/secret key pair, $\{y_{\text{bg}_\beta}, x_{\text{bg}_\beta}\}$.
- The DCC generates and signs a certificate for the public key generated by each GW.
- Each GW is equipped with the certificate certifying its public key. The corresponding secret key is kept secret and tamper-proof.

All the entities' keys and certificates are listed in Table III. For simplicity, the table only lists the entity's ID and certified keys contained in each certificate. A certificate typically contains a number of data items including: version number, serial number, issuing certification authority's (CA's) ID, CA's digital signature, subject/owner, owner's public key, validity period, certificate usage, signature algorithm and extensions. Also for the sake of simplicity, it is assumed that, if an entity has more than one public key, all the public keys are certified in a single certificate.

3) SM/GW Installation and Key Establishment: This phase is also divided into three steps outlined below.

Step 1: During an SM installation, the digital certificates of the SM's regional DNO, its local BG and its user's contracted supplier are installed onto the SM.

Step 2: During a GW installation, each BG is installed with the certificates of its child SMs and its parent NG. Similarly, each NG is installed with its child BGs' and its parent WG's

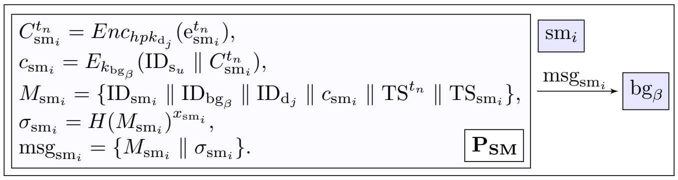

$C^{t_n}_{\mathrm{sm}_i} = Enc_{hpk_{\mathrm{d}_j}}(\mathrm{e}^{t_n}_{\mathrm{sm}_i}),$
$c_{\mathrm{sm}_i} = E_{k_{\mathrm{bg}_\beta}}(\mathrm{ID}_{\mathrm{s}_u} \parallel C^{t_n}_{\mathrm{sm}_i}),$
$M_{\mathrm{sm}_i} = \{\mathrm{ID}_{\mathrm{sm}_i} \parallel \mathrm{ID}_{\mathrm{bg}_\beta} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel c_{\mathrm{sm}_i} \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{sm}_i}\},$
$\sigma_{\mathrm{sm}_i} = H(M_{\mathrm{sm}_i})^{x_{\mathrm{sm}_i}},$
$\mathrm{msg}_{\mathrm{sm}_i} = \{M_{\mathrm{sm}_i} \parallel \sigma_{\mathrm{sm}_i}\}.$
$\mathbf{P_{SM}}$
sm_i
$\mathrm{msg}_{\mathrm{sm}_i}$
bg_β

Fig. 4. SM report generation: the $\mathbf{P_{SM}}$ processing step.

certificates, and each WG with its child NGs' and DCC's certificates. Each GW is installed with its regional DNO's certificate and has a list of its child SMs/GWs.

Step 3: After the installations, each BG establishes a secure channel with each of its child SMs. This can be done by establishing a shared secret (i.e., a symmetric key), e.g., k_{bg_β}, between the entities using a standard security protocol such as TLS [40]. This secret should be updated regularly.

C. DEP2SA in Detail

The DEP2SA scheme consists of three parts: (1) SM report generation, (2) decentralised and selective data aggregation and (3) data distribution and access.

1) SM Report Generation: At every time slot t_n, each SM generates a report that contains its user's ECD and sends it to its local BG. This processing step ($\mathbf{P_{SM}}$) is described next and shown in Fig. 4.

$\mathbf{P_{SM}}$: At the start of a slot, say t_{n+1}, each SM, sm_i, constructs a message that contains its user's ECD consumed during the previous slot, t_n, and sends the message to its local BG, bg_β. In detail, sm_i performs the following operations.

1. It reads its user's CD in slot t_n, $\mathrm{e}^{t_n}_{\mathrm{sm}_i}$, from its register.
2. It encrypts $\mathrm{e}^{t_n}_{\mathrm{sm}_i}$ with its regional DNO's homomorphic public key generating $C^{t_n}_{\mathrm{sm}_i} = Enc_{hpk_{\mathrm{d}_j}}(\mathrm{e}^{t_n}_{\mathrm{sm}_i})$. This encryption is to protect $\mathrm{e}^{t_n}_{\mathrm{sm}_i}$ against eavesdropping attacks by unauthorised entities.
3. It encrypts $\{\mathrm{ID}_{\mathrm{s}_u} \parallel C^{t_n}_{\mathrm{sm}_i}\}$ with the symmetric key it shares with bg_β generating $c_{\mathrm{sm}_i} = E_{k_{\mathrm{bg}_\beta}}(\mathrm{ID}_{\mathrm{s}_u} \parallel C^{t_n}_{\mathrm{sm}_i})$, where $\mathrm{ID}_{\mathrm{s}_u}$ is the ID of the user's supplier, s_u. This encryption is to protect the confidentiality of (i) $\mathrm{e}^{t_n}_{\mathrm{sm}_i}$ against eavesdropping attacks by authorised entities (i.e., the regional DNO that holds the homomorphic private key, hsk_{d_j}) and (ii) $\mathrm{ID}_{\mathrm{s}_u}$ against eavesdropping attacks by unauthorised entities.
4. It constructs $M_{\mathrm{sm}_i} = \{\mathrm{ID}_{\mathrm{sm}_i} \parallel \mathrm{ID}_{\mathrm{bg}_\beta} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel c_{\mathrm{sm}_i} \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{sm}_i}\}$, where $\mathrm{ID}_{\mathrm{sm}_i}$, $\mathrm{ID}_{\mathrm{bg}_\beta}$ and $\mathrm{ID}_{\mathrm{d}_j}$ are the IDs of the SM, local BG and regional DNO, respectively, TS^{t_n} is the time stamp of the slot t_n, used to uniquely identify the slot (e.g., date-t_n), and $\mathrm{TS}_{\mathrm{sm}_i}$ is the SM's local time stamp used to resist replay attacks.
5. It signs on M_{sm_i} to generate a signature, $\sigma_{\mathrm{sm}_i} = H(M_{\mathrm{sm}_i})^{x_{\mathrm{sm}_i}}$, which is used to resist active attacks (any forgery or unauthorised modification of data).
6. It constructs and sends $\mathrm{msg}_{\mathrm{sm}_i} = \{M_{\mathrm{sm}_i} \parallel \sigma_{\mathrm{sm}_i}\}$ to its local BG, e.g., bg_β.

2) Decentralised and Selective Data Aggregation: At every time slot, t_n, users' ECDs are grouped and aggregated at various levels in the network. In other words, data aggregations are performed progressively at different nodes as the data traverse across the different networks. At each GW, data aggregation is performed respectively based on the users' suppliers, and at the DCC, based on the users' suppliers as well as locations. This part consists of four processing steps: $\mathbf{P_{BG}}$, $\mathbf{P_{NG}}$, $\mathbf{P_{WG}}$ and $\mathbf{P_{DCC,1}}$.

$\mathbf{P_{BG}}$: Each BG receives messages from its child SMs, verifies and groups them based on the users' suppliers, aggregates the ECDs contained in the messages in each group, and sends the aggregated ECDs (AECDs) to its upstream NG. If some of the received messages fail to arrive or fail the verifications after multiple attempts, the BG reports the SMs that dispatch the unsuccessful messages to its regional DNO, and aggregates only the ECDs carried in messages that have passed successfully the verifications. A flowchart of the BG's operations is shown in Fig. 5 and explained below.

1. At the start of slot t_{n+1}, each BG, bg_β, changes the status of all its child SMs on its list to $\mathrm{m}^{\mathbf{P}}$ (i.e., it is *pending* receipt of messages), resets its registers and starts a countdown timer, tmr_1. This timer sets the maximum time period bg_β should wait for any pending messages before it performs a batch verification of signatures on the received and partially verified messages. The value of this period should be chosen such that, by the expiration of this value, bg_β should have received messages from all of the SMs it connects, so that it could batch verify them. This batch verification is used to reduce the computational cost at bg_β (more details are given shortly).
2. bg_β checks the timers' status (more details are given below).
3. bg_β checks if it has received a new message.
4. For each received message, $\mathrm{msg}_{\mathrm{sm}_i}$, bg_β verifies the data contained in the message in terms of:
   a) freshness (V^{fr}), i.e., it checks if the difference between its local time stamp and the time stamp contained in $\mathrm{msg}_{\mathrm{sm}_i}$ is less than a predefined value ($t_\triangle$), i.e., if $|\mathrm{TS}_{\mathrm{bg}_\beta} - \mathrm{TS}_{\mathrm{sm}_i}| \leq t_\triangle$,
   b) recipient (V^{rec}), i.e., it checks if the ID of the intended recipient of $\mathrm{msg}_{\mathrm{sm}_i}$, $\mathrm{ID}_{\mathrm{bg}_\beta}$, is the same as its own ID contained in its certificate,
   c) sender (V^{sen}), i.e., it checks if the ID of the claimed sender of $\mathrm{msg}_{\mathrm{sm}_i}$, $\mathrm{ID}_{\mathrm{sm}_i}$, is the same as the ID of one of its child SMs on its list,
   d) status (V^{st}), i.e., it checks if sm_i's status on its list is $\mathrm{m}^{\mathbf{P}}$,
   e) time slot (V^{ts}), i.e., it checks if TS^{t_n} is the expected one (in sequence).

   V^{fr} is used to resist replay attacks. V^{rec} and V^{sen} are used to eliminate messages that are not destined (expected) to (by) bg_β. V^{st} and V^{ts} are used to resist faulty/malicious SMs which send (i) more than one authentic messages during t_n and (ii) messages containing CDs measured at a time slot different than t_n, respectively. These verification methods are lightweight and aim to detect faulty/malicious SMs as early as possible, reduce computational costs at BGs and ensure that the correct data are aggregated.

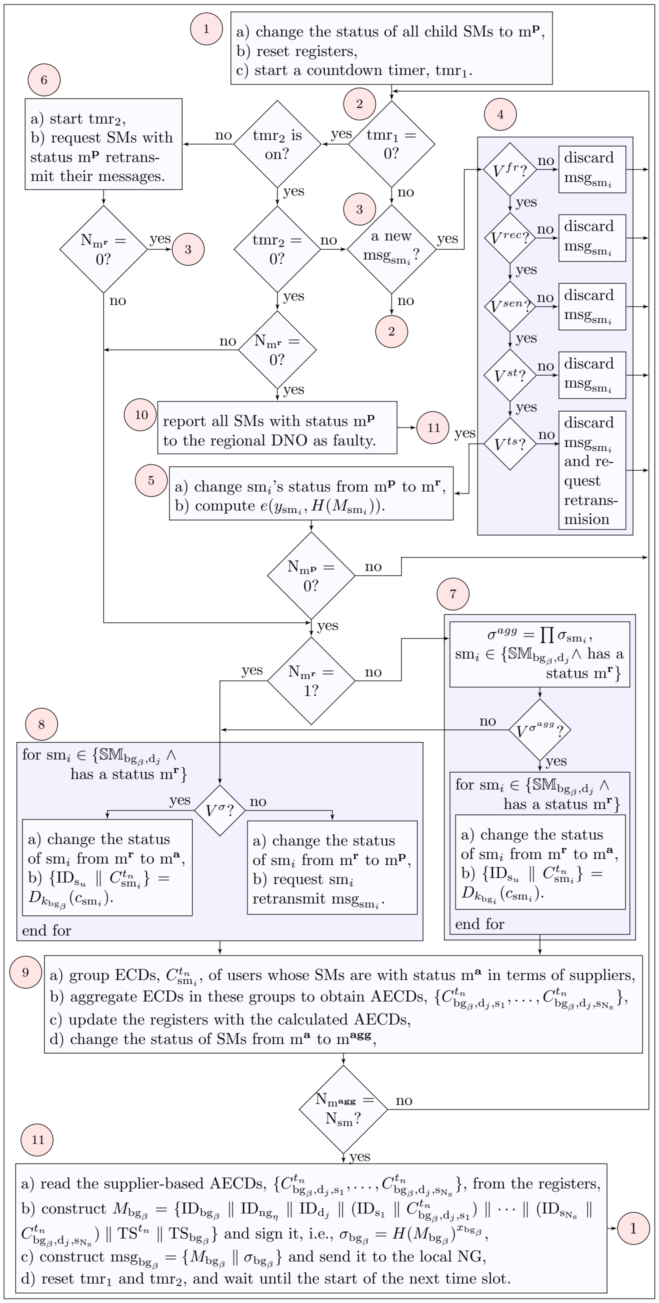

1
a) change the status of all child SMs to $\mathrm{m}^{\mathbf{p}}$,
b) reset registers,
c) start a countdown timer, tmr_1.
2
$\mathrm{tmr}_1 =$ 0?
yes
tmr_2 is on?
no
6
a) start tmr_2,
b) request SMs with status $\mathrm{m}^{\mathbf{p}}$ retransmit their messages.
$\mathrm{N}_{\mathrm{m}^{\mathbf{r}}} =$ 0?
yes
3
no
yes
$\mathrm{tmr}_2 =$ 0?
no
3
a new $\mathrm{msg}_{\mathrm{sm}_i}$?
no
2
yes
4
V^{fr}?
no
discard $\mathrm{msg}_{\mathrm{sm}_i}$
yes
V^{rec}?
no
discard $\mathrm{msg}_{\mathrm{sm}_i}$
yes
V^{sen}?
no
discard $\mathrm{msg}_{\mathrm{sm}_i}$
yes
V^{st}?
no
discard $\mathrm{msg}_{\mathrm{sm}_i}$
yes
V^{ts}?
no
discard $\mathrm{msg}_{\mathrm{sm}_i}$ and request retransmision
yes
yes
$\mathrm{N}_{\mathrm{m}^{\mathbf{r}}} =$ 0?
no
yes
10
report all SMs with status $\mathrm{m}^{\mathbf{p}}$ to the regional DNO as faulty.
11
5
a) change sm_i's status from $\mathrm{m}^{\mathbf{p}}$ to $\mathrm{m}^{\mathbf{r}}$,
b) compute $e(y_{\mathrm{sm}_i}, H(M_{\mathrm{sm}_i}))$.
$\mathrm{N}_{\mathrm{m}^{\mathbf{p}}} =$ 0?
no
yes
7
$\sigma^{agg} = \prod \sigma_{\mathrm{sm}_i}$, $\mathrm{sm}_i \in \{\mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j} \wedge$ has a status $\mathrm{m}^{\mathbf{r}}\}$
$\mathrm{N}_{\mathrm{m}^{\mathbf{r}}} =$ 1?
yes
no
$V^{\sigma^{agg}}$?
no
yes
8
for $\mathrm{sm}_i \in \{\mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j} \wedge$ has a status $\mathrm{m}^{\mathbf{r}}\}$
V^{σ}?
yes
no
a) change the status of sm_i from $\mathrm{m}^{\mathbf{r}}$ to $\mathrm{m}^{\mathbf{a}}$,
b) $\{\mathrm{ID}_{\mathrm{s}_u} \parallel C^{t_n}_{\mathrm{sm}_i}\} = D_{k_{\mathrm{bg}_\beta}}(c_{\mathrm{sm}_i})$.
a) change the status of sm_i from $\mathrm{m}^{\mathbf{r}}$ to $\mathrm{m}^{\mathbf{p}}$,
b) request sm_i retransmit $\mathrm{msg}_{\mathrm{sm}_i}$.
end for
for $\mathrm{sm}_i \in \{\mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j} \wedge$ has a status $\mathrm{m}^{\mathbf{r}}\}$
a) change the status of sm_i from $\mathrm{m}^{\mathbf{r}}$ to $\mathrm{m}^{\mathbf{a}}$,
b) $\{\mathrm{ID}_{\mathrm{s}_u} \parallel C^{t_n}_{\mathrm{sm}_i}\} = D_{k_{\mathrm{bg}_i}}(c_{\mathrm{sm}_i})$.
end for
9
a) group ECDs, $C^{t_n}_{\mathrm{sm}_i}$, of users whose SMs are with status $\mathrm{m}^{\mathbf{a}}$ in terms of suppliers,
b) aggregate ECDs in these groups to obtain AECDs, $\{C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_1}, \ldots, C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}\}$,
c) update the registers with the calculated AECDs,
d) change the status of SMs from $\mathrm{m}^{\mathbf{a}}$ to $\mathrm{m}^{\mathbf{agg}}$,
$\mathrm{N}_{\mathrm{m}^{\mathbf{agg}}} =$ N_{sm}?
no
yes
11
a) read the supplier-based AECDs, $\{C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_1}, \ldots, C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}\}$, from the registers,
b) construct $M_{\mathrm{bg}_\beta} = \{\mathrm{ID}_{\mathrm{bg}_\beta} \parallel \mathrm{ID}_{\mathrm{ng}_\eta} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel (\mathrm{ID}_{\mathrm{s}_1} \parallel C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_1}) \parallel \cdots \parallel (\mathrm{ID}_{\mathrm{s}_{\mathrm{N_s}}} \parallel C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}) \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{bg}_\beta}\}$ and sign it, i.e., $\sigma_{\mathrm{bg}_\beta} = H(M_{\mathrm{bg}_\beta})^{x_{\mathrm{bg}_\beta}}$,
c) construct $\mathrm{msg}_{\mathrm{bg}_\beta} = \{M_{\mathrm{bg}_\beta} \parallel \sigma_{\mathrm{bg}_\beta}\}$ and send it to the local NG,
d) reset tmr_1 and tmr_2, and wait until the start of the next time slot.
1

Fig. 5. A flowchart of the operations at a BG: the $\mathbf{P_{BG}}$ processing step.

5. It changes sm_i's status on its list to $\mathrm{m}^{\mathbf{r}}$ (*received*) and performs a pairing operation, i.e., $e(y_{\mathrm{sm}_i}, H(M_{\mathrm{sm}_i}))$. An SM with status $\mathrm{m}^{\mathbf{r}}$ means that its message passed the initial lightweight verifications described in the previous step.
6. If some messages fail to arrive or fail to pass any of the verifications before the expiration of tmr_1, bg_β starts tmr_2 (i.e., extends the period set by tmr_1) and requests SMs with status $\mathrm{m}^{\mathbf{p}}$ on its list to resend their messages. Note that if bg_β receives messages from all the SMs it connects before the expiration of tmr_1 and if these messages are authentic, bg_β skips this step.
7. It computes an aggregate signature of the messages from SMs with status $\mathrm{m}^{\mathbf{r}}$, i.e., $\sigma^{agg} = \prod \sigma_{\mathrm{sm}_i}$, where $\mathrm{sm}_i \in \{\mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j} \wedge$ has a status $\mathrm{m}^{\mathbf{r}}\}$, and verifies σ^{agg} using the batch verification method ($V^{\sigma^{agg}}$), i.e., if $e(g, \sigma^{agg}) = \prod e(y_{\mathrm{sm}_i}, H(M_{\mathrm{sm}_i}))$. If $V^{\sigma^{agg}}$ is positive, bg_β accepts the messages, changes the SMs' status from $\mathrm{m}^{\mathbf{r}}$ to $\mathrm{m}^{\mathbf{a}}$ (*authentic*), decrypts c_{sm_i} in the messages using k_{bg_β} and skips the next step. Note that this step is not necessary, but it is good for efficiency as, to batch verify x messages, bg_β has to perform $x+1$ computationally expensive pairing operations, as opposed to $2x$, if it verifies them one by one. An SM with status $\mathrm{m}^{\mathbf{a}}$ means that its message passed all the verifications and the ciphertext in the message can be aggregated.
8. If $V^{\sigma^{agg}}$ is not positive or is not performed, then for each $\mathrm{msg}_{\mathrm{sm}_i}$ from SMs with status $\mathrm{m}^{\mathbf{r}}$, bg_β performs signature verification (V^{σ}), i.e., checks if $e(g, \sigma_{\mathrm{sm}_i}) = e(y_{\mathrm{sm}_i}, H(M_{\mathrm{sm}_i}))$. If V^{σ} is positive, bg_β changes the status of sm_i to $\mathrm{m}^{\mathbf{a}}$ and decrypts c_{sm_i} contained in $\mathrm{msg}_{\mathrm{sm}_i}$. Otherwise, if V^{σ} is negative, it changes the status to $\mathrm{m}^{\mathbf{p}}$ and requests sm_i to retransmit $\mathrm{msg}_{\mathrm{sm}_i}$.
9. It groups and aggregates users' ECDs (from SMs with status $\mathrm{m}^{\mathbf{a}}$) based on the users' suppliers (i.e., the ECDs destined to the same supplier are aggregated into one AECD), i.e., $C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_u} = \prod C^{t_n}_{\mathrm{sm}_i}$ for $u = \{1, \ldots, \mathrm{N_s}\}$, where $\mathrm{sm}_i \in \{\mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_u} \wedge$ has a status $\mathrm{m}^{\mathbf{a}}\}$. If its registers are reset, bg_β stores $C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_u}$. Otherwise it multiplies them with the ones already stored and updates its registers with the result. Then, it changes the status of the SMs from $\mathrm{m}^{\mathbf{a}}$ to $\mathrm{m}^{\mathbf{agg}}$ (*aggregated*), i.e., the ciphertexts in their messages were aggregated.
10. If tmr_2 times out, bg_β reports all the SMs with status $\mathrm{m}^{\mathbf{p}}$ to the regional DNO using a standard protocol such as TLS (satisfying (F1)). Note that bg_β skips this step if it receives messages from all the SMs it connects before expiration of $\{\mathrm{tmr}_1 + \mathrm{tmr}_2\}$ and if all the messages pass V^{fr}, V^{rec}, V^{sen}, V^{st}, V^{ts} and V^{σ} verifications.
11. It reads the supplier-based AECDs from its registers, constructs $\mathrm{msg}_{\mathrm{bg}_\beta} = \{M_{\mathrm{bg}_\beta} \parallel \sigma_{\mathrm{bg}_\beta}\}$, where $M_{\mathrm{bg}_\beta} = \{\mathrm{ID}_{\mathrm{bg}_\beta} \parallel \mathrm{ID}_{\mathrm{ng}_\eta} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel (\mathrm{ID}_{\mathrm{s}_1} \parallel C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_1}) \parallel \ldots \parallel (\mathrm{ID}_{\mathrm{s}_{\mathrm{N_s}}} \parallel C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}) \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{bg}_\beta}\}$, $\sigma_{\mathrm{bg}_\beta} = H(M_{\mathrm{bg}_\beta})^{x_{\mathrm{bg}_\beta}}$, sends $\mathrm{msg}_{\mathrm{bg}_\beta}$ to its local NG, resets its timers, and waits until the start of the next slot.

Note that SMs' status helps a BG keep track of the SMs whose (i) messages are pending and (ii) data are aggregated. So, DEP2SA allows further information to be included in the aggregated messages along the network such as the aggregate value includes x out y SMs (y-x failed to report). This number would be different at each level, as appropriate.

$\mathbf{P_{NG}}$ & $\mathbf{P_{WG}}$: The operations performed by each NG (WG) are similar to those carried out by BGs except that the messages processed are from BGs (NGs), and the symmetric decryption in Steps 7 and 8 is skipped as none of the GWs perform encryption tasks. For example, bg_β, in contrast to sm_i, sends $\{\mathrm{ID}_{\mathrm{s}_u} \parallel C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_u}\}$ as it is.

$\mathbf{P_{DCC,1}}$: The operations performed by the DCC are similar to those performed by NGs/WGs except that the messages

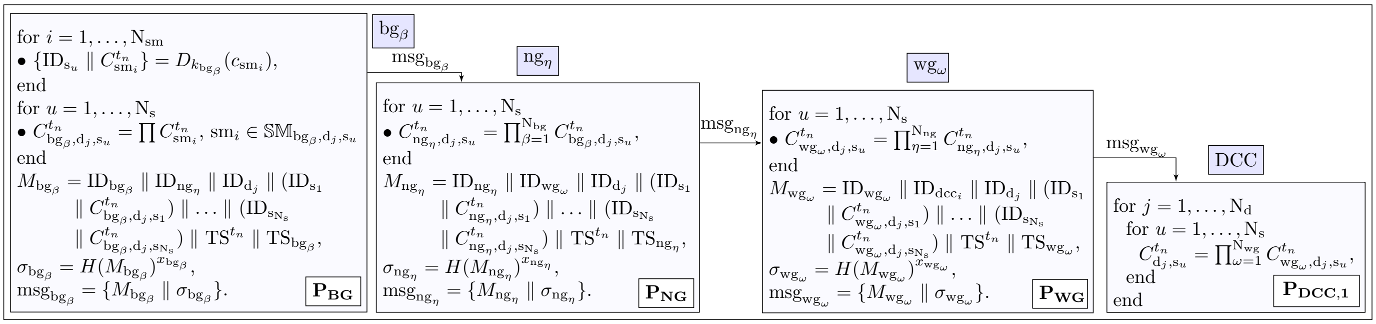

for $i = 1, \ldots, \mathrm{N_{sm}}$
• $\{\mathrm{ID}_{\mathrm{s}_u} \parallel C^{t_n}_{\mathrm{sm}_i}\} = D_{k_{\mathrm{bg}_\beta}}(c_{\mathrm{sm}_i})$,
end
for $u = 1, \ldots, \mathrm{N_s}$
• $C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_u} = \prod C^{t_n}_{\mathrm{sm}_i}$, $\mathrm{sm}_i \in \mathbb{SM}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_u}$
end
$M_{\mathrm{bg}_\beta} = \mathrm{ID}_{\mathrm{bg}_\beta} \parallel \mathrm{ID}_{\mathrm{ng}_\eta} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel (\mathrm{ID}_{\mathrm{s}_1} \parallel C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_1}) \parallel \ldots \parallel (\mathrm{ID}_{\mathrm{s}_{\mathrm{N_s}}} \parallel C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}) \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{bg}_\beta}$,
$\sigma_{\mathrm{bg}_\beta} = H(M_{\mathrm{bg}_\beta})^{x_{\mathrm{bg}_\beta}}$,
$\mathrm{msg}_{\mathrm{bg}_\beta} = \{M_{\mathrm{bg}_\beta} \parallel \sigma_{\mathrm{bg}_\beta}\}$.
$\mathbf{P_{BG}}$
bg_β
$\mathrm{msg}_{\mathrm{bg}_\beta}$
ng_η
for $u = 1, \ldots, \mathrm{N_s}$
• $C^{t_n}_{\mathrm{ng}_\eta,\mathrm{d}_j,\mathrm{s}_u} = \prod_{\beta=1}^{\mathrm{N_{bg}}} C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_u}$,
end
$M_{\mathrm{ng}_\eta} = \mathrm{ID}_{\mathrm{ng}_\eta} \parallel \mathrm{ID}_{\mathrm{wg}_\omega} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel (\mathrm{ID}_{\mathrm{s}_1} \parallel C^{t_n}_{\mathrm{ng}_\eta,\mathrm{d}_j,\mathrm{s}_1}) \parallel \ldots \parallel (\mathrm{ID}_{\mathrm{s}_{\mathrm{N_s}}} \parallel C^{t_n}_{\mathrm{ng}_\eta,\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}) \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{ng}_\eta}$,
$\sigma_{\mathrm{ng}_\eta} = H(M_{\mathrm{ng}_\eta})^{x_{\mathrm{ng}_\eta}}$,
$\mathrm{msg}_{\mathrm{ng}_\eta} = \{M_{\mathrm{ng}_\eta} \parallel \sigma_{\mathrm{ng}_\eta}\}$.
$\mathbf{P_{NG}}$
$\mathrm{msg}_{\mathrm{ng}_\eta}$
wg_ω
for $u = 1, \ldots, \mathrm{N_s}$
• $C^{t_n}_{\mathrm{wg}_\omega,\mathrm{d}_j,\mathrm{s}_u} = \prod_{\eta=1}^{\mathrm{N_{ng}}} C^{t_n}_{\mathrm{ng}_\eta,\mathrm{d}_j,\mathrm{s}_u}$,
end
$M_{\mathrm{wg}_\omega} = \mathrm{ID}_{\mathrm{wg}_\omega} \parallel \mathrm{ID}_{\mathrm{dcc}_i} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel (\mathrm{ID}_{\mathrm{s}_1} \parallel C^{t_n}_{\mathrm{wg}_\omega,\mathrm{d}_j,\mathrm{s}_1}) \parallel \ldots \parallel (\mathrm{ID}_{\mathrm{s}_{\mathrm{N_s}}} \parallel C^{t_n}_{\mathrm{wg}_\omega,\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}) \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{wg}_\omega}$,
$\sigma_{\mathrm{wg}_\omega} = H(M_{\mathrm{wg}_\omega})^{x_{\mathrm{wg}_\omega}}$,
$\mathrm{msg}_{\mathrm{wg}_\omega} = \{M_{\mathrm{wg}_\omega} \parallel \sigma_{\mathrm{wg}_\omega}\}$.
$\mathbf{P_{WG}}$
$\mathrm{msg}_{\mathrm{wg}_\omega}$
DCC
for $j = 1, \ldots, \mathrm{N_d}$
for $u = 1, \ldots, \mathrm{N_s}$
$C^{t_n}_{\mathrm{d}_j,\mathrm{s}_u} = \prod_{\omega=1}^{\mathrm{N_{wg}}} C^{t_n}_{\mathrm{wg}_\omega,\mathrm{d}_j,\mathrm{s}_u}$,
end
end
$\mathbf{P_{DCC,1}}$

Fig. 6. Decentralised and selective data aggregation: the main operations of the $\mathbf{P_{BG}}$, $\mathbf{P_{NG}}$, $\mathbf{P_{WG}}$ and $\mathbf{P_{DCC,1}}$ processing steps.

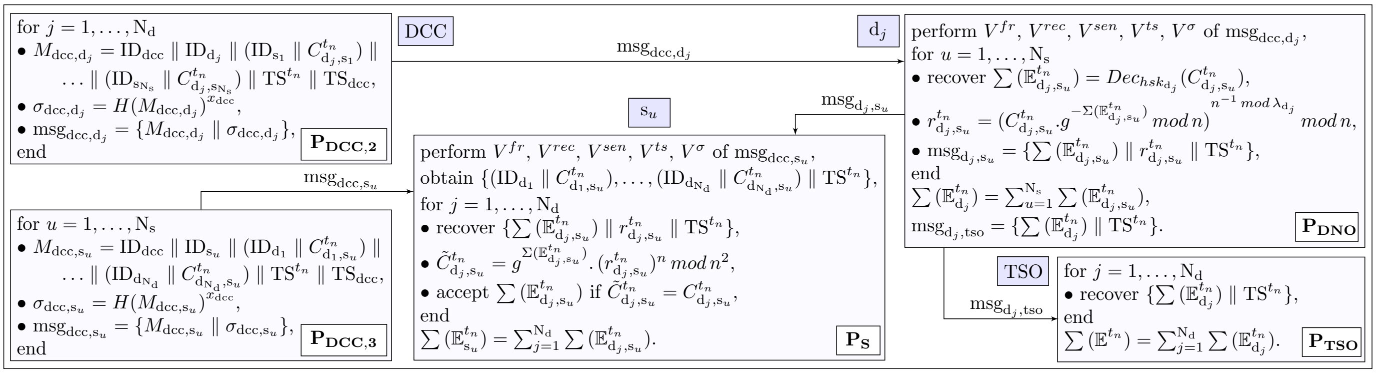

for $j = 1, \ldots, \mathrm{N_d}$
• $M_{\mathrm{dcc},\mathrm{d}_j} = \mathrm{ID_{dcc}} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel (\mathrm{ID}_{\mathrm{s}_1} \parallel C^{t_n}_{\mathrm{d}_j,\mathrm{s}_1}) \parallel \ldots \parallel (\mathrm{ID}_{\mathrm{s}_{\mathrm{N_s}}} \parallel C^{t_n}_{\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}) \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS_{dcc}}$,
• $\sigma_{\mathrm{dcc},\mathrm{d}_j} = H(M_{\mathrm{dcc},\mathrm{d}_j})^{x_{\mathrm{dcc}}}$,
• $\mathrm{msg}_{\mathrm{dcc},\mathrm{d}_j} = \{M_{\mathrm{dcc},\mathrm{d}_j} \parallel \sigma_{\mathrm{dcc},\mathrm{d}_j}\}$,
end
$\mathbf{P_{DCC,2}}$
DCC
$\mathrm{msg}_{\mathrm{dcc},\mathrm{d}_j}$
d_j
perform V^{fr}, V^{rec}, V^{sen}, V^{ts}, V^{σ} of $\mathrm{msg}_{\mathrm{dcc},\mathrm{d}_j}$,
for $u = 1, \ldots, \mathrm{N_s}$
• recover $\sum(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u}) = Dec_{hsk_{\mathrm{d}_j}}(C^{t_n}_{\mathrm{d}_j,\mathrm{s}_u})$,
• $r^{t_n}_{\mathrm{d}_j,\mathrm{s}_u} = (C^{t_n}_{\mathrm{d}_j,\mathrm{s}_u}.g^{-\Sigma(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u})}\, mod\, n)^{n^{-1}\, mod\, \lambda_{\mathrm{d}_j}}\, mod\, n$,
• $\mathrm{msg}_{\mathrm{d}_j,\mathrm{s}_u} = \{\sum(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u}) \parallel r^{t_n}_{\mathrm{d}_j,\mathrm{s}_u} \parallel \mathrm{TS}^{t_n}\}$,
end
$\sum(\mathbb{E}^{t_n}_{\mathrm{d}_j}) = \sum_{u=1}^{\mathrm{N_s}} \sum(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u})$,
$\mathrm{msg}_{\mathrm{d}_j,\mathrm{tso}} = \{\sum(\mathbb{E}^{t_n}_{\mathrm{d}_j}) \parallel \mathrm{TS}^{t_n}\}$.
$\mathbf{P_{DNO}}$
$\mathrm{msg}_{\mathrm{d}_j,\mathrm{s}_u}$
s_u
$\mathrm{msg}_{\mathrm{dcc},\mathrm{s}_u}$
for $u = 1, \ldots, \mathrm{N_s}$
• $M_{\mathrm{dcc},\mathrm{s}_u} = \mathrm{ID_{dcc}} \parallel \mathrm{ID}_{\mathrm{s}_u} \parallel (\mathrm{ID}_{\mathrm{d}_1} \parallel C^{t_n}_{\mathrm{d}_1,\mathrm{s}_u}) \parallel \ldots \parallel (\mathrm{ID}_{\mathrm{d}_{\mathrm{N_d}}} \parallel C^{t_n}_{\mathrm{d}_{\mathrm{N_d}},\mathrm{s}_u}) \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS_{dcc}}$,
• $\sigma_{\mathrm{dcc},\mathrm{s}_u} = H(M_{\mathrm{dcc},\mathrm{s}_u})^{x_{\mathrm{dcc}}}$,
• $\mathrm{msg}_{\mathrm{dcc},\mathrm{s}_u} = \{M_{\mathrm{dcc},\mathrm{s}_u} \parallel \sigma_{\mathrm{dcc},\mathrm{s}_u}\}$,
end
$\mathbf{P_{DCC,3}}$
perform V^{fr}, V^{rec}, V^{sen}, V^{ts}, V^{σ} of $\mathrm{msg}_{\mathrm{dcc},\mathrm{s}_u}$,
obtain $\{(\mathrm{ID}_{\mathrm{d}_1} \parallel C^{t_n}_{\mathrm{d}_1,\mathrm{s}_u}), \ldots, (\mathrm{ID}_{\mathrm{d}_{\mathrm{N_d}}} \parallel C^{t_n}_{\mathrm{d}_{\mathrm{N_d}},\mathrm{s}_u}) \parallel \mathrm{TS}^{t_n}\}$,
for $j = 1, \ldots, \mathrm{N_d}$
• recover $\{\sum(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u}) \parallel r^{t_n}_{\mathrm{d}_j,\mathrm{s}_u} \parallel \mathrm{TS}^{t_n}\}$,
• $\tilde{C}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u} = g^{\Sigma(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u})}.(r^{t_n}_{\mathrm{d}_j,\mathrm{s}_u})^n\, mod\, n^2$,
• accept $\sum(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u})$ if $\tilde{C}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u} = C^{t_n}_{\mathrm{d}_j,\mathrm{s}_u}$,
end
$\sum(\mathbb{E}^{t_n}_{\mathrm{s}_u}) = \sum_{j=1}^{\mathrm{N_d}} \sum(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u})$.
$\mathbf{P_S}$
TSO
$\mathrm{msg}_{\mathrm{d}_j,\mathrm{tso}}$
for $j = 1, \ldots, \mathrm{N_d}$
• recover $\{\sum(\mathbb{E}^{t_n}_{\mathrm{d}_j}) \parallel \mathrm{TS}^{t_n}\}$,
end
$\sum(\mathbb{E}^{t_n}) = \sum_{j=1}^{\mathrm{N_d}} \sum(\mathbb{E}^{t_n}_{\mathrm{d}_j})$.
$\mathbf{P_{TSO}}$

Fig. 7. Data distribution and access: the $\mathbf{P_{DCC,2}}$, $\mathbf{P_{DCC,3}}$, $\mathbf{P_{DNO}}$, $\mathbf{P_S}$ and $\mathbf{P_{TSO}}$ processing steps.

processed are from WGs, and AECDs are aggregated based on users' suppliers and users' locations (i.e., DNOs) (rather than just based on the users' suppliers).

Without loss of generality, Fig. 6 summarises the main operations in the $\mathbf{P_{BG}}$, $\mathbf{P_{NG}}$, $\mathbf{P_{WG}}$ and $\mathbf{P_{DCC,1}}$ processing steps assuming that all the messages in these steps are authentic and received on time.

3) Data Distribution and Access: DCC distributes different sets of AECDs to their respective authorised recipients, i.e., DNOs and suppliers. The DNOs recover users' ACDs from the AECDs and send respective sets of the ACDs to the TSO and the corresponding suppliers. These steps are summarised in Fig. 7 and explained below.

$\mathbf{P_{DCC,2}}$: For each DNO, e.g., d_j, DCC constructs a message that contains the supplier-based AECDs of the users in the region managed by the DNO, i.e., $\{C^{t_n}_{\mathrm{d}_j,\mathrm{s}_1}, \ldots, C^{t_n}_{\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}\}$. It then signs the message and sends it to the DNO.

$\mathbf{P_{DCC,3}}$: For each supplier, e.g., s_u, DCC constructs a message that contains the region-based AECDs of the users supplied by the supplier, i.e., $\{C^{t_n}_{\mathrm{d}_1,\mathrm{s}_u}, \ldots, C^{t_n}_{\mathrm{d}_{\mathrm{N_d}},\mathrm{s}_u}\}$. It then signs the message and sends it to the supplier.

As the number of grid operators and suppliers in the grid is small, the TSO, DNOs and suppliers can use a standard protocol such as TLS to establish secure and authentic communication channels between each pair of them (or among themselves). Thus, in the next processing steps we only present the data which these entities send to each other.

$\mathbf{P_{DNO}}$: Each DNO verifies the message received from the DCC, recovers the supplier-based ACDs ($\mathrm{N_s}$ included in the message in total) using its homomorphic private key (satisfying (F2b)), recovers also the random number embedded in each AECD using (2), and sends both items to the respective suppliers. Then, it calculates the total ACD in its region, i.e., $\sum(\mathbb{E}^{t_n}_{\mathrm{d}_j}) = \sum_{u=1}^{\mathrm{N_s}} \sum(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u})$ (satisfying (F2a)), and sends the result to the TSO.

$\mathbf{P_S}$: Each supplier verifies the message received from the DCC and obtains the region-based AECDs. It then, upon receiving a message from each DNO, recovers the ACD and the random number, and use them (together with the homomorphic public key of the corresponding DNO) to compute the AECD. If the AECD computed is the same as the AECD received from the DCC, the supplier accepts the ACD as authentic (satisfying (F3b)). It then computes the ACD of all its customers, i.e., $\sum(\mathbb{E}^{t_n}_{\mathrm{s}_u}) = \sum_{j=1}^{\mathrm{N_d}} \sum(\mathbb{E}^{t_n}_{\mathrm{d}_j,\mathrm{s}_u})$ (satisfying (F3a)).

$\mathbf{P_{TSO}}$: Upon receiving a message from each DNO, the TSO recovers the ACD for each DNO, e.g., $\sum(\mathbb{E}^{t_n}_{\mathrm{d}_j})$, (satisfying (F4a)). It then calculates the ACD of all the users in the grid, i.e., $\sum(\mathbb{E}^{t_n}) = \sum_{j=1}^{\mathrm{N_d}} \sum(\mathbb{E}^{t_n}_{\mathrm{d}_j})$ (satisfying (F4b)).

VI. Security Analysis

In this section we analyse the security and privacy properties of the DEP2SA scheme.

A. Protocol Message Authenticity

Each message in the DEP2SA scheme contains a BLS short signature which is proven secure under chosen-message attack in the random oracle model assuming that the Computational Diffie-Hellman problem is hard [37], [38]. In addition, the signature signing keys are stored in tamper-proof devices, and the corresponding signature verification keys are certified by the TA. Hence, DEP2SA provides assurance of source authentication, non-repudiation of origin and integrity of each protocol message (satisfying (S1)). Any active attacks on the data in transit can be detected and the modified data discarded. Including a time stamp in each message also ensures that all received messages are fresh.

B. *Confidentiality of Users' CDs*

In DEP2SA, users' CDs are encrypted at their source (SMs) using the Paillier cryptosystem, then the ECDs are progressively aggregated, and the AECDs are delivered to DNOs, where ACDs are recovered and selections of them are delivered, respectively, to their need-to-know entities (TSO and suppliers). As the Paillier cryptosystem is semantically secure against chosen plaintext attacks (assuming that the Composite Residuosity Class problem is hard [3]) and communication channels connecting the TSO, DNOs and suppliers are secure and authentic (e.g., established using TLS), only authorised entities (i.e., the TSO, DNOs and suppliers) can access the ACDs of users (satisfying (S2)). All the external entities and unauthorised internal entities (including GWs, the DCC), should they eavesdrop messages in transit, would only be able to access the ECDs or AECDs (but not ACDs) of the users.

C. *User Privacy-preservation*

With DEP2SA, a DNO receives only the supplier-based AECDs of the users located in its region of operation. In other words, the most fine grained CDs which a DNO and a supplier have access to is the ACDs of a set of users located in a particular region and supplied by a particular supplier, and usually the size of this set is on the order of thousands. Even authorised entities (i.e., the TSO, DNOs, suppliers) do not have access to individual users' CDs. Moreover, unlike other schemes, DEP2SA is also resistant against eavesdropping attacks by authorised entities as users' CDs are double encrypted (first with the regional DNO's homomorphic public key and then with the key shared between the SMs and their local BG) while in transit between the SMs and the BG. For the similar reason, if DNOs' homomorphic private keys are compromised, DEP2SA can still operate and protect users' privacy as the most fine-grained data attackers could access is the AECDs sent by BGs. As long as these AECDs contain the ACDs of a sufficient number of users (more details are given in Section VII), it is hard for attackers to work out individual users' CDs (satisfying (S3)). Also, as the IDs of users' contracted suppliers are encrypted while in transit between SMs and BGs, eavesdroppers can not figure out which supplier is contracted by which user.

D. *Authorisation to Access Users' ACDs*

The 'principle of least privilege' (i.e., only allow an entity to have access to data just sufficient for it to carry out its duties (business responsibilities)) has been applied in the design of DEP2SA. The use of the region-based cryptographic key deployment (i.e., each DNO has its own homomorphic public/private key pair) combined with the recipient-based selective aggregation of users' ECDs ensures that only the DNOs that need to know a set of users' CDs can actually decrypt the ACD of this set, thus making the scheme resilient to attacks mounted by external and authorised internal entities such as the elevation of privilege attacks (satisfying (S4)). Also, the use of the recipient-based selective distribution of ACDs ensures that suppliers receive only the ACDs of their customers.

TABLE IV
SECURITY LEVEL COMPARISON

	[5]	[6]	[7]	[8]	[9]	DEP2SA
Message authenticity		√	√	√		√
Confidentiality	√	√	√	√	√	√
User privacy preservation						√
Authorisation						√
Availability						√

E. *Availability*

DEP2SA is designed with resilience to DoS attacks in mind. As it uses a decentralised approach to message verifications and data aggregation, there is no entity in the system that bares an imbalanced processing load during an execution of the scheme, thus avoiding the creation of a performance bottleneck. This approach brings us an extra advantage, i.e., malicious or unauthorised messages can be detected by a node that is one-hop away from their originators and can immediately be discarded upon the detections. In addition, the verifying entities first deploy lightweight verification methods to detect any unauthorised messages, thus reducing the risks of any DoS attacks affecting the performance of DEP2SA (satisfying (S5)).

F. *Security Level Comparison*

The security properties achieved by DEP2SA in comparison with related schemes [5]–[9] are summarised in Table IV. Compared to these schemes, DEP2SA achieves the highest level of protections.

G. *Cloud-based DCC*

Delegating the operations of the DCC to a semi-trusted cloud service provider should not affect the security and privacy properties of the DEP2SA scheme. The DCC only handles, and operates on, ciphertexts, so its operations can be delegated to a semi-trusted cloud service provider. This provider will not have access to any of the CDs and/or ACDs of users. To ensure the correct operation of the DEP2SA scheme, the provider should aggregate the ciphertexts generated by subsets of users (the ciphertexts attached with the same DNOs and suppliers IDs form one subset) and then distribute the aggregated ciphertexts to their respective authorised recipients (i.e., DNOs and suppliers). Considering that a cloud service provider is usually paid based on the amount of data it processes, and that the amount of data a service provider is expected to process (in our system model) at any given time slot will be constant (as the number of SMs served would remain the same for a given time slot), there is no financial incentive for the cloud service provider to group and aggregate the ciphertexts incorrectly as this will not affect the amount of money payable to the service provider. On the contrary, there is every incentive for the cloud service provider to perform the aggregations and to provide the DCC services correctly and truthfully, as this will enhance its reputation, which, in turn, can increase its business standing and market share.

VII. User Private Data Leakage from Aggregated Consumption Data

Although the authorised entities in DEP2SA receive only ACDs (not individual users' CDs), they may still manage to obtain some of the individual users' private data by analysing the received ACDs. In this section we discuss potential private data leakage from ACDs and propose a simple method to quantify it. Such method can be used for finding out the minimum number of users whose CDs should be aggregated such that the resulted ACD provides sufficient user privacy preservation.

A. Problem Description and Our Aim

Suppose that there are two sets of users. The first set has 10000 users and the set of CDs of these users is denoted as $\mathbb{E}_{10^4}$. The second set has only two users and the set of CDs of these two users is denoted as $\mathbb{E}_2$. If an entity has access only to the ACD of the first set of users, denoted as $\sum(\mathbb{E}_{10^4})$, it would be difficult (if not impossible) for the entity to disaggregate $\sum(\mathbb{E}_{10^4})$ into CDs of individual users. The entity may only learn some statistical patterns of this set of users, but not individual users' consumption patterns or some relevant behaviours or activities.

However, this is not the case for the second set of users. Owing to the small number of users in it, if an entity has access to the ACD of this set of users, i.e., $\sum(\mathbb{E}_2)$, it may be feasible for the entity to disaggregate $\sum(\mathbb{E}_2)$ into individual components and learn each of the users' raw CDs. The entity then, with the help of NILM techniques, can translate such users' raw CDs into users' specific behaviours/activities, breaching the users' privacy. It is worth noting that with a small set of users, even the ACD such as $\sum(\mathbb{E}_2)$ can be translated to users' specific activities, as $\sum(\mathbb{E}_2)$ may not be sufficient to disguise some specific appliance load signatures.

The example above clearly shows that the level of private data leakage from an ACD is dependent on individual users' CDs and the number of users whose CDs formed the ACD. How to measure/quantify such leakage is an open question.

Our aim is to propose a method that can quantify the level of private data leakage from an ACD, thus to provide the designers of SG/AMI with a tool to find out the minimum number of users whose CDs should be aggregated so that the risk of inferring a user's private behavior or activities from the ACD can be controlled at an acceptable level.

B. Definitions and Experimental Dataset

We define a notation of 'strong' user privacy if none of the entities (described in Section III-A) can access individual users' raw CDs and/or detect some specific human activities or appliance operations and link them to individual users. We also define a notation of 'strong' adversary capability if the adversary (including authorised entities) has a NILM algorithm that can decompose an user's CD into a set of appliance CDs with 100% accuracy.

Our analysis is based on a real-life dataset, "Electricity Customer Behaviour Trial" [41], that contains 6,287 users' CDs collected at 30-minute intervals for 536 days.

C. Our Hypothesis

Usually the ACD of a large number of users is available in the public domain. For example, the real-time electricity demand data of a country (or a region in a county) are regularly published by the county's grid operators (e.g., the real-time demand data of UK is available at [42]). These data can give some indications of the electricity consumption patterns of the country's entire population. However, owing to the large number of users in the set, it is hard for an adversary to learn a particular user's consumption pattern (specific activities) from the ACD. The only useful information the adversary may get from the ACD is the users' overall behaviour.

Assuming that an adversary has access to the ACD of a large number of users, $\sum(\mathbb{E}_{\mathrm{x}})$, we argue that the ACD of a subset of these users, $\sum(\mathbb{E}_{\mathrm{y}})$, would leak no or minimum information with regards to an individual user's private data as long as $\sum(\mathbb{E}_{\mathrm{y}})$ follows the same trend as $\sum(\mathbb{E}_{\mathrm{x}})$, i.e., as long as the difference between the two ACD distributions over a period of time, $P_{\Sigma(\mathbb{E}_{\mathrm{x}}^{t_n})}$ and $P_{\Sigma(\mathbb{E}_{\mathrm{y}}^{t_n})}$, is negligible. The difference between the two distributions can be measured by the K-divergence [43], i.e.,

$$K = \sum_{n=1}^{\mathrm{N_t}} P_{\Sigma(\mathbb{E}_{\mathrm{y}}^{t_n})} \log \frac{2P_{\Sigma(\mathbb{E}_{\mathrm{y}}^{t_n})}}{P_{\Sigma(\mathbb{E}_{\mathrm{y}}^{t_n})} + P_{\Sigma(\mathbb{E}_{\mathrm{x}}^{t_n})}}. \tag{3}$$

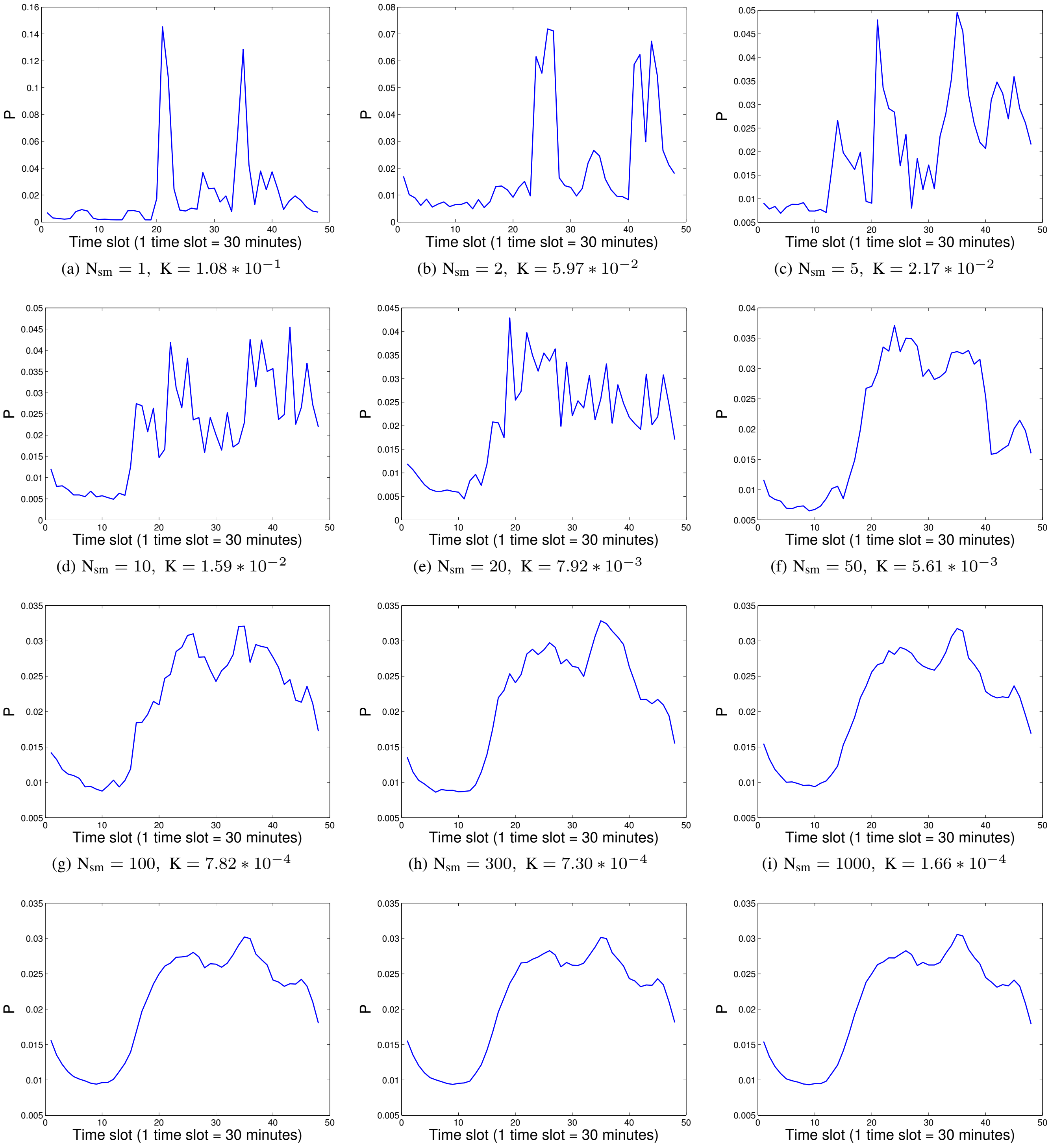
An advantageous property of the K-divergence as compared to other distance measures, such as the the relative entropy, is that its value is between zero and one [43].

D. The Experiment and Results

Here we use the dataset from [41] to analyse the statistical differences between the ACD distribution of all the users in a neighbourhood and the ACD distributions of various subsets of these users over a 24-hour time period. The ACD of users in a given set can be computed by:

$$\sum(\mathbb{E}_{\mathrm{N_{sm}}}^{t_n}) = \sum_{i=1}^{\mathrm{N_{sm}}} \mathrm{e}_{\mathrm{sm}_i}^{t_n}, \text{for } n = \{1, \ldots \mathrm{N_t}\}, \tag{4}$$

where $\mathrm{e}_{\mathrm{sm}_i}^{t_n}$ is the CD during t_n of the ith user in the set, $\mathrm{N_t}$ is the resolution of CDs (e.g., in a given period of 24 hours, if CDs are taken at every 30-minute interval, then $\mathrm{N_t} = 48$) and $\mathrm{N_{sm}}$ is the size of the set (which is the number of users whose CDs are aggregated).



(a) $\mathrm{N_{sm}} = 1,\ \mathrm{K} = 1.08 * 10^{-1}$

(b) $\mathrm{N_{sm}} = 2,\ \mathrm{K} = 5.97 * 10^{-2}$

(c) $\mathrm{N_{sm}} = 5,\ \mathrm{K} = 2.17 * 10^{-2}$

(d) $\mathrm{N_{sm}} = 10,\ \mathrm{K} = 1.59 * 10^{-2}$

(e) $\mathrm{N_{sm}} = 20,\ \mathrm{K} = 7.92 * 10^{-3}$

(f) $\mathrm{N_{sm}} = 50,\ \mathrm{K} = 5.61 * 10^{-3}$

(g) $\mathrm{N_{sm}} = 100,\ \mathrm{K} = 7.82 * 10^{-4}$

(h) $\mathrm{N_{sm}} = 300,\ \mathrm{K} = 7.30 * 10^{-4}$

(i) $\mathrm{N_{sm}} = 1000,\ \mathrm{K} = 1.66 * 10^{-4}$

(j) $\mathrm{N_{sm}} = 3000,\ \mathrm{K} = 1.83 * 10^{-5}$

(k) $\mathrm{N_{sm}} = 5000,\ \mathrm{K} = 8.85 * 10^{-6}$

(l) $\mathrm{N_{sm}} = 6287,\ \mathrm{K} = 0$

Fig. 8. The ACD distributions of subsets with an increasing number of users from a dataset, i.e., $P_{\Sigma(\mathbb{E}_1^{t_n})}$ (8a), ..., $P_{\Sigma(\mathbb{E}_{5000}^{t_n})}$ (8k), and the ACD distribution of all the users in the dataset, i.e., $P_{\Sigma(\mathbb{E}_{6287}^{t_n})}$ (8l), over a 24-hour period.

The experiment is carried out as follows. First a 24-hour period is randomly chosen, a set containing the CDs during this period of all the users from the dataset, i.e., $\mathbb{E}_{6287}$ ($\mathrm{N_{sm}} = 6287$, equivalent to a typical neighbourhood), is formed and the ACD of all the users in the set, i.e., $\sum(\mathbb{E}_{6287}^{t_n})$, is computed using (4). Then a subset of $\mathbb{E}_{6287}$ containing the CD of only one randomly chosen user from the dataset is formed. Next, iteratively, the CD of another randomly chosen user from the remaining dataset is included in the subset. After each iteration the ACD of all the users in the subset is computed using (4). Then the distributions of the ACD of all the users in (i) different versions of the subset, i.e., $P_{\Sigma(\mathbb{E}_{\mathrm{N_{sm}}}^{t_n})}$

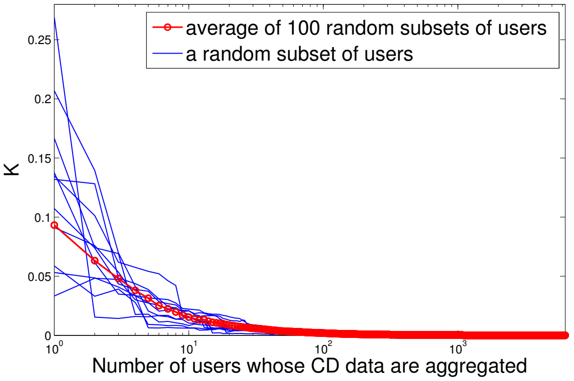


Fig. 9. The K-divergence between the ACD distribution of all the users (i.e., 6287 users) in a dataset and the ACD distributions of random subsets with varying number of users from the same dataset.

for $\mathrm{N_{sm}} = \{1, \ldots, 6286\}$ (some shown in Fig. 8a to Fig. 8k), and (ii) in $\mathbb{E}_{6287}$, i.e., $P_{\Sigma(\mathbb{E}^{t_n}_{6287})}$ (Fig. 8l), over the 24-hour period are computed by:

$$P_{\Sigma(\mathbb{E}^{t_n}_{\mathrm{N_{sm}}})} = \frac{\sum(\mathbb{E}^{t_n}_{\mathrm{N_{sm}}})}{\sum\limits_{n=1}^{\mathrm{N_t}} \sum(\mathbb{E}^{t_n}_{\mathrm{N_{sm}}})}. \tag{5}$$

The K-divergence between each of $P_{\Sigma(\mathbb{E}^{t_n}_{\mathrm{N_{sm}}})}$ ($\mathrm{N_{sm}} = \{1, \ldots, 6286\}$) and $P_{\Sigma(\mathbb{E}^{t_n}_{6287})}$ are computed using (3). As shown in Fig. 8, as $\mathrm{N_{sm}}$ increases, $P_{\Sigma(\mathbb{E}^{t_n}_{\mathrm{N_{sm}}})}$ becomes more similar to $P_{\Sigma(\mathbb{E}^{t_n}_{6287})}$, thus the K-divergence between them decreases accordingly. The more similar the two distributions are, the less individual users' private data is leaked from $\sum(\mathbb{E}^{t_n}_{\mathrm{N_{sm}}})$. Therefore, the K-divergence value can be used as a measure of the level of private data leakage from ACDs. The smaller K is, the less private data about individual users could be leaked.

To minimise the effect of the order in the selection of the members of the subset $\mathbb{E}_{\mathrm{N_{sm}}}$ ($\mathrm{N_{sm}} = \{1, \ldots, 6286\}$), we run our experiment 100 times with 100 different and randomly chosen such subsets and compute the average K-divergence value (as shown in Fig. 9). We observe that as we increase $\mathrm{N_{sm}}$, the K-divergence (the difference) between $P_{\Sigma(\mathbb{E}^{t_n}_{\mathrm{N_{sm}}})}$ and $P_{\Sigma(\mathbb{E}^{t_n}_{6287})}$ decreases rapidly until it reaches a value (i.e., $\mathrm{K_T} = 5 * 10^{-3}$) after which further increasing $\mathrm{N_{sm}}$ has a negligible effect on K. $\mathrm{K_T}$ can be used as a threshold value for preserving individual users' privacy when users' ACDs are used. In other words, to ensure that the ACD of a set of users does not leak individual users' private data (i.e., to preserve a sufficient level of privacy for the user in the set), the minimum number of users included in the set should be such that the resulting K value is lower than $\mathrm{K_T}$.

VIII. Performance Evaluation

In this section we evaluate the DEP2SA scheme in terms of computational complexity imposed on various entities in the system and the communication overheads incurred between SMs and the DCC. We also compare the performance of DEP2SA with the performances of two state-of-the-art aggregation schemes: the efficient privacy-preserving aggregation scheme [8], named as EPPA, and the decentralised security framework [9], named as DSF. As in DEP2SA, both schemes use homomorphic enctryption technique (the Paillier cryptosystem) to protect users' privacy. However, EPPA aggregates users' CDs only at BGs, whereas DSF, similar to DEP2SA, aggregates the data also at NGs and WGs.

A. Computational Complexity

Computationally expensive operations used in DEP2SA are exponentiation operation in Z_{n^2} (EOZ), exponentiation operation in G (EOG) and pairing operation (PO).

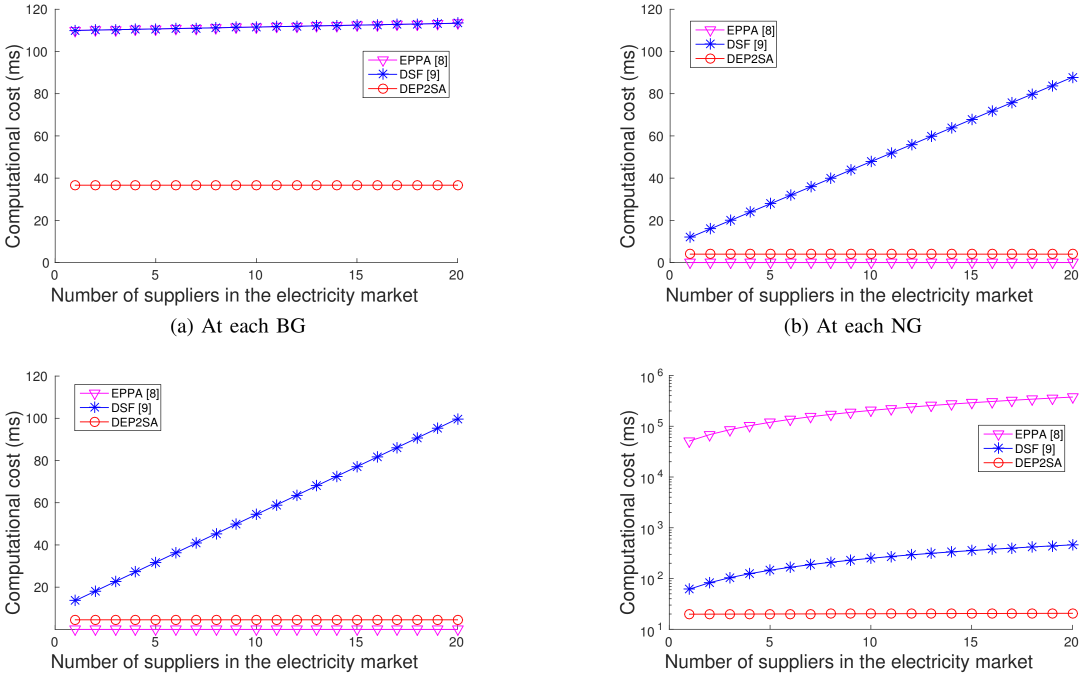
In DEP2SA, and in each time slot, an SM performs one EOZ to encrypt its user's CD, and one EOG to sign its message; a BG does $(\mathrm{N_{sm}} + 1)$ POs to perform a batch verification of messages from its child SMs and one EOG to sign its message; similarly, an NG and a WG performs $(\mathrm{N_{bg}} + 1)$ POs and one EOG, and $(\mathrm{N_{ng}} + 1)$ POs and one EOG, respectively; DCC performs $(\mathrm{N^G_{wg}} + 1)$ POs and $(\mathrm{N_d} + \mathrm{N_s})$ EOGs to sign its message for each DNO and supplier.

As EPPA [8] and DSF [9] use a single-recipient system model, and to meet the functional requirements, in each time slot, an SM performs three EOZs to encrypt its CD three times using the homomorphic public keys of three different recipients (the TSO, its regional DNO and supplier) and three EOGs to sign three different messages destined to each of these recipients; a BG does $2(\mathrm{N_{sm}} + 1)$ and $\mathrm{N_s}(\frac{\mathrm{N_{sm}}}{\mathrm{N_s}} + 1)$ POs to perform batch verifications of the messages destined to the TSO and DNO, and $\mathrm{N_s}$ number of suppliers, respectively, and $(2 + \mathrm{N_s})$ EOGs to sign $(2 + \mathrm{N_s})$ different messages (to TSO, DNO and each of the suppliers). As EPPA [8] aggregates data only at BGs, the computational costs at NGs and WGs are negligible (the messages are simply forwarded), whereas DCC performs $(\mathrm{N^G_{wg}}\mathrm{N_{ng}}\mathrm{N_{bg}}(2 + \mathrm{N_s}) + \mathrm{N_d}(1 + \mathrm{N_s}) + 1)$ POs to verify the messages sent from the BGs and destined to TSO, DNOs and suppliers, and $(\mathrm{N_d} + \mathrm{N_s} + 1)$ EOGs to sign a different message to the TSO, each DNO and each supplier. As DSF [9] aggregates data also at NGs and WGs, each NG and WG perform $(\mathrm{N_{bg}}+1)(2+\mathrm{N_s})$ and $(\mathrm{N_{ng}}+1)(2+\mathrm{N_s})$ POs, and $(2 + \mathrm{N_s})$ and $(2 + \mathrm{N_s})$ EOGs, respectively, whereas DCC performs $(\mathrm{N^G_{wg}}(2+\mathrm{N_s})+\mathrm{N_d}(1+\mathrm{N_s})+1)$ POs and $(\mathrm{N_d}+\mathrm{N_s}+1)$ EOGs.

We denote EOZ, EOG and PO as o_{ez}, o_{eg} and o_p, respectively, and summarise the computational complexities of DEP2SA, EPPA [8] and DSF [9] in Table V. We have also conducted experiments with the PBC [44] and MIRACL [45] libraries on a 3.0 GHz-processor and 4 GB-memory machine to study the operational costs of EOZ, EOG and PO. The experimental results show that EOZ ($|n^2| = 2,048$) costs 84.4 μs, EOG in G with 160 bits 43.5 μs and PO 136.1 μs. We set $\mathrm{N_{sm}} = 268$, $\mathrm{N_{bg}} = 28$, $\mathrm{N_{ng}} = 32$, $\mathrm{N^G_{wg}} = 140$ and $\mathrm{N_d} = 14$, so AMI could cover the entire grid in UK [35]. We depict the variations of computational costs in terms of $\mathrm{N_s}$ (the number of suppliers in a liberalised market) in Fig. 10. Compared to EPPA [8], DEP2SA has slightly more computational costs at an NG and a WG (as EPPA only forwards messages at NGs and WGs), but significantly less computational costs at a BG, and specifically at the DCC. Compared to DSF [9], DEP2SA introduces significantly less computational costs at each entity.

TABLE V
COMPUTATION COMPLEXITY COMPARISON

	EPPA [8]
SM	$3o_{ez} + 3o_{eg}$
BG	$(3\mathrm{N_{sm}} + 2 + \mathrm{N_s}) * o_p + (2 + \mathrm{N_s}) * o_{eg}$
NG	negligible
WG	negligible
DCC	$(\mathrm{N_{wg}^G N_{ng} N_{bg}}(2 + \mathrm{N_s}) + \mathrm{N_d}(1 + \mathrm{N_s}) + 1) * o_p + (\mathrm{N_d} + \mathrm{N_s} + 1) * o_{eg}$
	DSF [9]
SM	$3o_{ez} + 3o_{eg}$
BG	$(3\mathrm{N_{sm}} + 2 + \mathrm{N_s}) * o_p + (2 + \mathrm{N_s}) * o_{eg}$
NG	$(\mathrm{N_{bg}} + 1)(2 + \mathrm{N_s}) * o_p + (2 + \mathrm{N_s}) * o_{eg}$
WG	$(\mathrm{N_{ng}} + 1)(2 + \mathrm{N_s}) * o_p + (2 + \mathrm{N_s}) * o_{eg}$
DCC	$(\mathrm{N_{wg}^G}(2 + \mathrm{N_s}) + \mathrm{N_d}(1 + \mathrm{N_s}) + 1) * o_p + (\mathrm{N_d} + \mathrm{N_s} + 1) * o_{eg}$
	DEP2SA
SM	$o_{ez} + o_{eg}$
BG	$(\mathrm{N_{sm}} + 1) * o_p + o_{eg}$
NG	$(\mathrm{N_{bg}} + 1) * o_p + o_{eg}$
WG	$(\mathrm{N_{ng}} + 1) * o_p + o_{eg}$
DCC	$(\mathrm{N_{wg}^G} + 1) * o_p + (\mathrm{N_d} + \mathrm{N_s}) * o_{eg}$



(a) At each BG

(b) At each NG

(c) At each WG

(d) At the DCC

Fig. 10. Computational cost comparison at a BG, NG, WG and DCC.

B. Communication Overheads

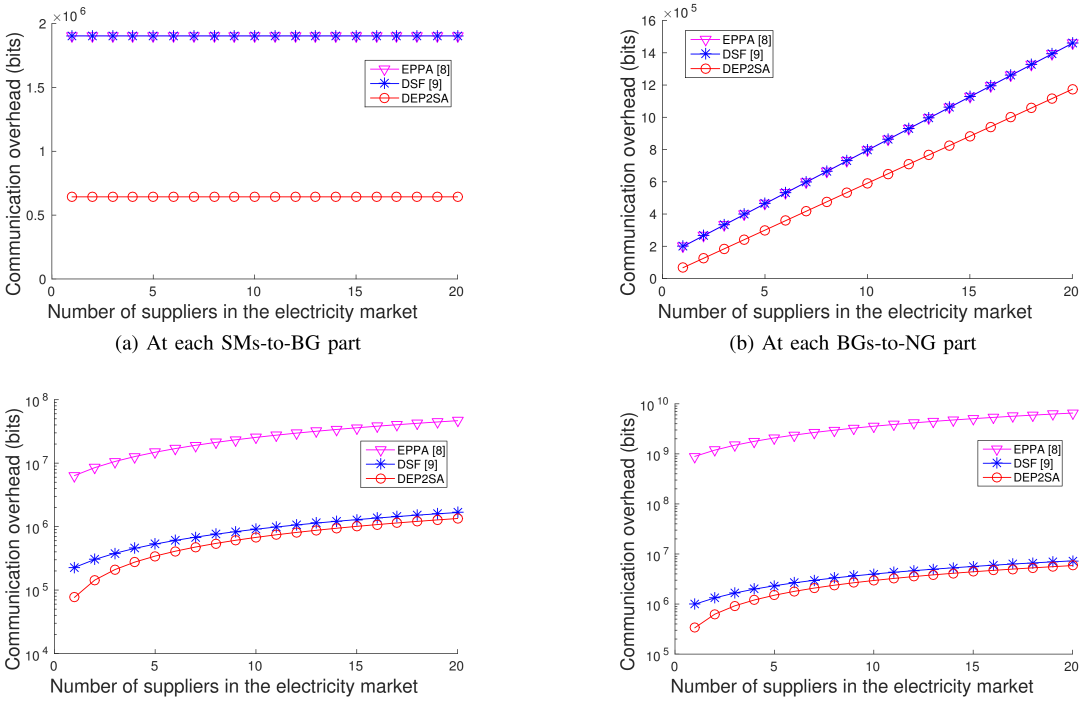
The communication overheads introduced by DEP2SA are largely in four parts: the overhead incurred (1) when SMs send data to a BG (denoted as SMs-to-BG), (2) when BGs send data to an NG (BGs-to-NG), (3) when NGs send data to a WG (NGs-to-WG), and (4) when WGs send data to the DCC (WGs-to-DCC).

With DEP2SA in each time slot, each SM sends its report, $\mathrm{msg}_{\mathrm{sm}_i} = \{\mathrm{ID}_{\mathrm{sm}_i} \parallel \mathrm{ID}_{\mathrm{bg}_\beta} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel c_{\mathrm{sm}_i} \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{sm}_i} \parallel \sigma_{\mathrm{sm}_i}\}$, to its local BG. Considering that each BG collects data from $\mathrm{N_{sm}}$ SMs, the SMs-to-BG communication overhead is $\mathrm{N_{sm}} * (3|\mathrm{ID}| + 2|\mathrm{TS}| + |\sigma| + |c|)$. Similarly, each BG sends only a single message, $\mathrm{msg}_{\mathrm{bg}_\beta} = \{\mathrm{ID}_{\mathrm{bg}_\beta} \parallel \mathrm{ID}_{\mathrm{ng}_\eta} \parallel \mathrm{ID}_{\mathrm{d}_j} \parallel (\mathrm{ID}_{\mathrm{s}_1} \parallel C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_1}) \parallel \ldots \parallel (\mathrm{ID}_{\mathrm{s}_{\mathrm{N_s}}} \parallel C^{t_n}_{\mathrm{bg}_\beta,\mathrm{d}_j,\mathrm{s}_{\mathrm{N_s}}}) \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{bg}_\beta} \parallel \sigma_{\mathrm{bg}_\beta}\}$, to its local NG. Considering that each NG collects data from $\mathrm{N_{bg}}$ BGs, the BGs-to-NG communication overhead is $\mathrm{N_{bg}} * (3|\mathrm{ID}| + 2|\mathrm{TS}| + |\sigma| + \mathrm{N_s} * (|\mathrm{ID}| + |C|))$. Similarly, the NGs-to-WG and WGs-to-DCC communication overheads are $\mathrm{N_{ng}} * (3|\mathrm{ID}| + 2|\mathrm{TS}| + |\sigma| + \mathrm{N_s} * (|\mathrm{ID}| + |C|))$ and $\mathrm{N_{wg}^G} * (3|\mathrm{ID}| + 2|\mathrm{TS}| + |\sigma| + \mathrm{N_s} * (|\mathrm{ID}| + |C|))$, respectively.

In contrast to DEP2SA, in EPPA [8] and DSF [9], each SM sends 3 different messages of the form $\{\mathrm{ID}_{\mathrm{sm}_i} \parallel \mathrm{ID}_{\mathrm{bg}_\beta} \parallel \mathrm{ID}_r \parallel C^{t_n}_{\mathrm{sm}_i} \parallel \mathrm{TS}^{t_n} \parallel \mathrm{TS}_{\mathrm{sm}_i} \parallel \sigma_{\mathrm{sm}_i}\}$ to three different recipients (i.e., $r = \{\mathrm{d}_j, \mathrm{tso}, \mathrm{s}_u\}$). Hence, the SMs-to-BG communication overhead is $3\mathrm{N_{sm}} * (3|\mathrm{ID}| + 2|\mathrm{TS}| + |\sigma| + |C|)$. Each BG sends $(2 + \mathrm{N_s})$ different messages of the same form to $(2 + \mathrm{N_s})$ different recipients (i.e., $r = \{\mathrm{d}_j, \mathrm{tso}, \mathrm{s}_1, \ldots, \mathrm{s}_{\mathrm{N_s}}\}$). Hence, the BGs-to-NG communication overhead is $\mathrm{N_{bg}} * (2 + \mathrm{N_s}) * (3|\mathrm{ID}| +$

TABLE VI
COMMUNICATION OVERHEAD COMPARISON

	EPPA [8]
SMs-to-BG	$3\mathrm{N_{sm}} * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$
BGs-to-NG	$\mathrm{N_{bg}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$
NGs-to-WG	$\mathrm{N_{ng}N_{bg}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$
WGs-to-DCC	$\mathrm{N_{wg}^{G}N_{ng}N_{bg}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$
	DSF [9]
SMs-to-BG	$3\mathrm{N_{sm}} * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$
BGs-to-NG	$\mathrm{N_{bg}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$
NGs-to-WG	$\mathrm{N_{ng}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$
WGs-to-DCC	$\mathrm{N_{wg}^{G}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$
	DEP2SA
SMs-to-BG	$\mathrm{N_{sm}} * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert c\rvert)$
BGs-to-NG	$\mathrm{N_{bg}} * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \mathrm{N_s} * (\lvert\mathrm{ID}\rvert + \lvert C\rvert))$
NGs-to-WG	$\mathrm{N_{ng}} * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \mathrm{N_s} * (\lvert\mathrm{ID}\rvert + \lvert C\rvert))$
WGs-to-DCC	$\mathrm{N_{wg}^{G}} * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \mathrm{N_s} * (\lvert\mathrm{ID}\rvert + \lvert C\rvert))$



(a) At each SMs-to-BG part

(b) At each BGs-to-NG part

(c) At each NGs-to-WG part

(d) At the WGs-to-DCC part

Fig. 11. Communication overhead comparison at the SMs-to-BG, BGs-to-NG, NGs-to-WG and WGs-to-DCC parts.

$2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$. Similarly, in DSF [9], the NGs-to-WG and WGs-to-DCC communication overheads are $\mathrm{N_{ng}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$ and $\mathrm{N_{wg}^{G}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$, respectively. As in EPPA [8] the NGs and WGs simply forward messages without performing any aggregation, the NGs-to-WG and WGs-to-DCC communication overheads are $\mathrm{N_{ng}N_{bg}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$ and $\mathrm{N_{wg}^{G}N_{ng}N_{bg}} * (2 + \mathrm{N_s}) * (3\lvert\mathrm{ID}\rvert + 2\lvert\mathrm{TS}\rvert + \lvert\sigma\rvert + \lvert C\rvert)$, respectively.

The communication overheads introduced by DEP2SA, EPPA [8] and DSF [9] are summarised in Table VI. Furthermore, using the setting n_{d_j} 1024-bit, $\lvert\mathrm{ID}\rvert$ and $\lvert\mathrm{TS}\rvert$ 32-bit and G 160-bit long [8], we depict the communication overheads in terms of $\mathrm{N_s}$ in Fig. 11. It can be seen that DEP2SA introduces less communication overheads than EPPA and DSF in every communication part. The total communication overhead between SMs and DCC is shown in Fig. 12. DEP2SA has significantly less total communication overhead compared to EPPA [8] and DSF [9].

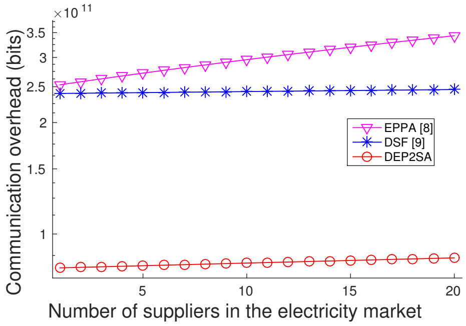


Fig. 12. Communication overhead comparison at the SMs-to-DCC part.

IX. Conclusion

In this paper, we have proposed a novel data aggregation scheme, DEP2SA, that allows grid operators and suppliers to collect users' aggregated consumption data efficiently and with user privacy preservation. DEP2SA combines the use of homomorphic encryption and selective data aggregation based on the geographic locations of the data source (i.e., SMs) and the intended recipients of the aggregated data. As a result, DEP2SA allows authorised entities to access ACDs of users on the need-to-know basis, i.e., different authorised entities can only gain access to the ACDs of the subgroups of users under their respective managements, thus making the scheme secure in terms of preserving users' privacy and readily applicable to liberalised electricity markets. Also, the adoption of a decentralised and progressive data verification and aggregation model makes the scheme both computationally and bandwidth-wise efficient. Particularly, with this approach, the computation complexity is independent of the number of SMs in the system, thus making DEP2SA scalable. Analytical and numerical results have shown that, in comparison to existing related schemes, DEP2SA offers significant improvements in terms of computational complexity and scalability. Security analysis has demonstrated that DEP2SA satisfies the security and privacy-preservation requirements specified, including the principle of least privilege.

Acknowledgment

This research was supported by the Engineering and Physical Sciences Research Council (EPSRC) and Toshiba Research Europe Limited under Grant [EP/I501541/1].

References

[1] H. Farhangi, "The path of the smart grid," *Power and Energy Magazine, IEEE*, vol. 8, no. 1, pp. 18–28, Jan. 2010.

[2] E. L. Quinn, "Privacy and the new energy infrastructure," *Social Sience Research Networks (SSRN)*, Feb. 2009.

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

[4] X. He, M.-O. Pun, and C.-C. Kuo, "Secure and efficient cryptosystem for smart grid using homomorphic encryption," in *the 3rd IEEE PES Conf. on Innovative Smart Grid Technologies (ISGT)*, 2012.

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

[6] P. Deng and L. Yang, "A secure and privacy-preserving communication scheme for advanced metering infrastructure," in *the 3rd IEEE PES Conf. on Innovative Smart Grid Technologies (ISGT)*, 2012.

[7] F. Li and B. Luo, "Preserving data integrity for smart grid data aggregation," in *the 3rd IEEE Int. Conf. on Smart Grid Communications (SmartGridComm)*, pp. 366–371, 2012.

[8] 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. 23, no. 9, pp. 1621–1631, Sep. 2012.

[9] S. Ruj and A. Nayak, "A decentralized security framework for data aggregation and access control in smart grids," *Smart Grid, IEEE Transactions on*, vol. 4, no. 1, pp. 196–205, Mar. 2013.

[10] F. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Security and Trust Management*, ser. LNCS. Springer Berlin Heidelberg, vol. 6710, pp. 226–238, 2011.

[11] F. Borges, D. Demirel, L. Bock, J. Buchmann, and M. Muhlhauser, "A privacy-enhancing protocol that provides in-network data aggregation and verifiable smart meter billing," in *the 19th IEEE Symposium on Computers and Communication (ISCC)*, pp. 1–6, 2014.

[12] F. Borges and M. Muhlhauser, "EPPP4SMS: Efficient privacy-preserving protocol for smart metering systems and its simulation using real-world data," *Smart Grid, IEEE Transactions on*, vol. 5, no. 6, pp. 2701–2708, Nov. 2014.

[13] X. Liang, X. Li, R. Lu, X. Lin, and X. Shen, "UDP: Usage-based dynamic pricing with privacy preservation for smart grid," *Smart Grid, IEEE Transactions on*, vol. 4, no. 1, pp. 141–150, Mar. 2013.

[14] Z. Erkin, J. Troncoso-Pastoriza, R. Lagendijk, and F. Perez-Gonzalez, "Privacy-preserving data aggregation in smart metering systems: An overview," *Signal Processing Magazine, IEEE*, vol. 30, no. 2, pp. 75–86, Mar. 2013.

[15] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *Parallel and Distributed Systems, IEEE Trans. on*, vol. 25, no. 8, pp. 2053–2064, Aug. 2014.

[16] M. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan, "DESA: A decentralized, efficient and selective aggregation scheme in AMI," in *IEEE Conf. on Innovative Smart Grid Technologies (ISGT)*, 2014.

[17] NIST, "Guidelines for smart grid cyber security," NISTIR 7628 Revision 1, vol. 1-3, Sep. 2014. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf

[18] IETF. (2011, Jun.) Internet protocols for the smart grid. RFC 6272. [Online]. Available: https://tools.ietf.org/html/rfc6272

[19] ETSI, "Machine-to-machine communications (M2M); threat analysis and counter-measures to M2M service layer," ETSI TR 103 167 V1.1.1, Tech. Rep., Aug. 2011.

[20] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *Security Privacy, IEEE*, vol. 7, pp. 75–77, 2009.

[21] Z. Fadlullah, M. Fouda, N. Kato, X. Shen, and Y. Nozaki, "An early warning system against malicious activities for smart grid communications," *Network, IEEE*, vol. 25, no. 5, pp. 50–55, 2011.

[22] H. Khurana, M. Hadley, N. Lu, and D. Frincke, "Smart-grid security issues," *Security Privacy, IEEE*, vol. 8, pp. 81–85, Jan. 2010.

[23] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid - The new and improved power grid: A survey," *Communications Surveys Tutorials, IEEE*, no. 4, pp. 944–980, Fourth quarter 2011.

[24] Z. Fan, P. Kulkarni, S. Gormus, C. Efthymiou, G. Kalogridis, M. Sooriyabandara, Z. Zhu, S. Lambotharan, and W. H. Chin, "Smart grid communications: Overview of research challenges, solutions, and standardization activities," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 1, pp. 21–38, First quarter 2013.

[25] J. Liu, Y. Xiao, S. Li, W. Liang, and C. L. P. Chen, "Cyber security and privacy issues in smart grids," *Communications Surveys Tutorials, IEEE*, vol. 14, no. 4, pp. 981–997, Fourth quarter 2012.

[26] M. Erol-Kantarci and H. Mouftah, "Smart grid forensic science: applications, challenges, and open issues," *Communications Magazine, IEEE*, vol. 51, no. 1, pp. 68–74, Jan. 2013.

[27] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *the 1st IEEE Int. Conf. on Smart Grid Communications (SmartGridComm)*, pp. 1–6, 2010.

[28] H.-Y. Lin, S.-T. Shen, and B. Lin, "A privacy preserving smart metering system supporting multiple time granularities," in *the 6th IEEE Int. Conf. on Software Security and Reliability Companion (SERE-C)*, pp. 119–126, 2012.

[29] F. Mármol, C. Sorge, O. Ugus, and G. Pérez, "Do not snoop my habits: preserving privacy in the smart grid," *Communications Magazine, IEEE*, vol. 50, no. 5, pp. 166–172, May 2012.

[30] D. Li, Z. Aung, J. R. Williams, and A. Sanchez, "Efficient and fault-diagnosable authentication architecture for ami in smart grid," *Security and Communication Networks*, vol. 8, no. 4, pp. 598–616, 2015. [Online]. Available: http://dx.doi.org/10.1002/sec.1006

[31] The electricity trading arrangements: A beginners guide. [Online]. Available: https://www.elexon.co.uk/wp-content/uploads/2015/10/beginners_guide_to_trading_arrangements_v5.0.pdf

[32] C. Rottondi, G. Verticale, and C. Krauss, "Privacy-preserving smart metering with multiple data consumers," *Computer Networks*, vol. 57, no. 7, pp. 1699–1713, May 2013.

[33] ——, "Distributed privacy-preserving aggregation of metering data in smart grids," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 7, pp. 1342–1354, July 2013.

[34] M. A. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan, "MUSP: Multi-service, user self-controllable and privacy-preserving system for smart metering," in *Communications (ICC), IEEE International Conference on*, 8-12 June, London, UK 2015, pp. 788–794.

[35] Ofgem. [Online]. Available: www.ofgem.gov.uk/electricity

[36] DCC. [Online]. Available: http://www.smartdcc.co.uk/

[37] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weil pairing,” *Journal of Cryptology*, vol. 17, pp. 297–319, 2004.

[38] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and verifiably encrypted signatures from bilinear maps,” in *Advances in Cryptology EUROCRYPT*, ser. LNCS. Springer Berlin Heidelberg, vol. 2656, pp. 416–432, 2003.

[39] D. Boneh and M. Franklin, “Identity-based encryption from the weil pairing,” in *Advances in Cryptology CRYPTO*, ser. LNCS. Springer Berlin Heidelberg, vol. 2139, pp. 213–229, 2001.

[40] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246 Std., Aug. 2008. [Online]. Available: https://www.ietf.org/rfc/rfc5246.txt

[41] Electricity customer behaviour trial. [Online]. Available: http://www.ucd.ie/issda/data/commissionforenergyregulationcer/

[42] [Online]. Available: http://www2.nationalgrid.com/uk/

[43] G. Kalogridis and S. Denic, “Data mining and privacy of personal behaviour types in smart grid,” in *the 11th IEEE Int. Conf. on Data Mining Workshops (ICDMW)*, pp. 636–642, 2011.

[44] PBC library. [Online]. Available: http://crypto.stanford.edu/pbc/

[45] MIRACL. [Online]. Available: http://certivox.org/display/EXT/MIRACL

Mustafa A. Mustafa received the B.Sc. degree in communications equipment and technologies from Technical University of Varna, Varna, Bulgaria, in 2007, the M.Sc. degree in communications and signal processing from Newcastle University, Newcastle upon Tyne, U.K., in 2010, and the Ph.D. degree in Computer Science from The University of Manchester, Manchester, U.K., in 2015. He is a Postdoctoral Research Fellow at the COSIC Research Group of the Electrical Engineering Department, KU Leuven, Belgium. His research interests include information security, data privacy, smart grid, e-health and IoT.

Ning Zhang received the B.Sc. degree from Dalian Maritime University, Dalian, China, and the Ph.D. degree from the University of Kent, Canterbury, U.K., all in electronics engineering. She is a Senior Lecturer in the School of Computer Science at the University of Manchester, Manchester, U.K. Her current research interests include security in networked and distributed systems, applied cryptography, data privacy, trust and digital right managements. She has authored papers and acted as referees and reviewers in these topic areas.

Georgios Kalogridis received the Dipl. El. Comp. Eng. from the University of Patras, Greece, in 2000, the M.Sc. degree in Advanced Computing from the University of Bristol, U.K., in 2001, and the Ph.D. degree in Mathematics from Royal Holloway, University of London, U.K., in 2011. He is a Principal Research Engineer and a Team Leader at Toshiba Telecommunications Research Laboratory, where he has been working since 2001. His research has spanned the areas of information security, data privacy, machine learning, wireless networking, smart grid communications and IoT. In these areas, he has authored papers, invented patents, developed prototypes, and contributed to collaborative research projects and standards.

Zhong Fan received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, and the Ph.D. degree in telecommunication networks from Durham University, Durham, U.K. He is a Chief Research Fellow at Toshiba Research Europe, Bristol, U.K. Prior to joining Toshiba, he worked as a Research Fellow at Cambridge University, Cambridge, U.K., a Lecturer at Birmingham University, Birmingham, U.K., and a Researcher at Marconi Labs, Cambridge, U.K. He was also awarded a BT Short-Term Fellowship to work at BT Labs. His research interests are wireless networks, IP networks, and machine-to-machine and smart grid communications.