MUSP: A Multi-service, User Self-controllable and Privacy-preserving System for Smart Metering

DOI:
10.1109/ICC.2015.7248418

Document Version
Accepted author manuscript

Link to publication record in Manchester Research Explorer

Citation for published version (APA):

Published in:
2015 IEEE International Conference on Communications (ICC)

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester's Takedown Procedures [http://man.ac.uk/04Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.
MUSP: Multi-service, User Self-controllable and Privacy-preserving System for Smart Metering

Mustafa A. Mustafa and Ning Zhang
School of Computer Science
The University of Manchester
Oxford Road, Manchester, M13 9PL, UK
Email: {mustafm, nzhang}@cs.man.ac.uk

Georgios Kalogridis and Zhong Fan
Toshiba Research Europe Limited
Telecommunications Research Laboratory
32 Queen Square, Bristol, BS1 4ND, UK
Email: {george, zhong.fan}@toshiba-trel.com

Abstract—This paper proposes a Multi-service, User Self-controllable and Privacy-preserving (MUSP) system for secure smart metering. This system has a number of novel properties. Firstly, it can report users’ fine-grained consumption data to grid operators and suppliers securely and with user privacy preservation capability. These are achieved by using a homomorphic encryption technique in conjunction with selective data aggregation and distribution methods, so only the aggregated data are delivered to the authorised data recipients only on a need-to-know basis. Secondly, it allows suppliers to access their customers’ attributable meter readings regularly. To protect users’ privacy, suppliers, by default, can access new data only at a low frequency (e.g. once a month). However, MUSP allows users (1) to adjust (control) this frequency and (2) to release new data by demand (e.g. when change of tariff occurs), thus putting users’ privacy preservation in their own hands. Thirdly, it is equipped with an easy and user friendly supplier switching facility to allow users to switch providers easily and conveniently. Security analysis and performance evaluation demonstrate that the MUSP system can protect users’ privacy while providing these services in an efficient and scalable manner.

I. INTRODUCTION

Advanced Metering Infrastructure (AMI) is part of the emerging Smart Grid (SG) that supports metering data communications [1], [2]. An important component of AMI is Smart Meters (SMs). SMs are devices capable of measuring, recording and communicating users’ detailed electricity consumption data to different SG entities such as grid operators and suppliers. By making use of such data, these entities would be able to operate the SG more efficiently and reliably.

However, along with the benefits, the SG also introduces new security and privacy issues [2]–[8]. For example, SG entities having access to detailed consumption data may use non-intrusive load monitoring techniques to build individual users’ electricity consumption patterns, thus breaching users’ privacy [9]. To prevent such breaches, none of the SG entities should be allowed to access individual users’ metering data at a high frequency. However, to allow account management, suppliers must access their customers’ (individual users’) attributable metering data\(^1\). Also, some users may not have stringent privacy requirements; they may not mind their suppliers to access their attributable metering data more frequently in return for incentives. So, a metering system should be flexible (able to support diverse requirements). One way to facilitate this flexibility is to allow users to control the release frequency of their attributable data. As electricity markets are liberalized (users are free to choose their suppliers), the system should also allow users to switch among different suppliers easily and securely. Thus, a smart metering system should support the following services in a secure, privacy-preserving and efficient manner:

(S1) Report aggregated consumption data to operators/suppliers at a high frequency (for efficient grid management);
(S2) Report meter readings to its user at a high frequency (for making the user aware of his/her electricity usage);
(S3) Allow suppliers to access attributable meter readings without users’ intervention (for account management);
(S4) Allow users to adjust the release frequency of their attributable meter readings (for privacy level control);
(S5) Allow users to release their attributable meter readings on demand (for accommodating cases like tariff change);
(S6) Allow users to easily switch among different suppliers.

This paper proposes such a system, a novel Multi-service, User Self-controllable and Privacy-preserving (MUSP) smart metering system. MUSP utilizes a homomorphic encryption technique to aggregate users’ detailed consumption data and only distribute the aggregated data to the respective grid operators and suppliers on a need-to-know basis. The latter property requires that users’ data are grouped depending on their intended recipients and that the homomorphic encryption is applied accordingly. MUSP also, with the help of a trusted party, allows users (1) to adjust (control) the frequency with which their suppliers can access their new attributable meter readings and (2) to securely switch suppliers. To the best of our knowledge, this is the first work on designing a smart metering system that supports all of the above mentioned services.

The rest of the paper is organized as follows: §II reviews related work. §III gives the design preliminaries. §IV describes our novel MUSP system, §V its security analysis and §VI its performance evaluation. §VII concludes the paper.

II. RELATED WORK

Efthymiou et al. [10] proposed a method for anonymizing users’ metering data sent at high frequency, so data recipients cannot link data to their owners. Lin et al. [11] proposed a system to allow users’ data to be accessed at multiple time

\(^1\)Attributable metering data: data that are tied to an individual SM/user.
granularities. Már mol et al. [12] proposed a system whereby a supplier aggregates the encrypted data sent by users and uses an aggregated key sent by a key aggregator to recover the aggregated data. Although these solutions preserve users’ privacy, they impose high levels of communication overheads.

To reduce the communication overheads, individual users’ data should be aggregated and only the aggregated data be delivered to the final recipients. To preserve users’ privacy during a data aggregation process, homomorphic encryption based aggregation schemes have been proposed [13]–[18]. However, these schemes use a single-recipient system model, thus they are not efficient when applied to liberalized electricity markets where multiple recipients are expected to access the aggregated data of different sets of users.

To address this limitation, Rottondi et al. [19] proposed an architecture with additional functional entities, Privacy Preserving Nodes (PPNs), that collect users’ data encrypted by a Secret Sharing Scheme (SSS) and perform aggregation of different sets of the data for their intended recipients governed by access rights. This solution has two drawbacks: (1) increased SG complexity due to the use of PPNs, and (2) increased communication overheads due to the use of SSS. The first drawback was overcome by allocating the tasks of PPNs to existing SG nodes (gateways) [20], and the second one by using an aggregated key sent by a key aggregator to recover the aggregated data. Although these solutions preserve users’ privacy, they impose high levels of communication overheads.

### C. Threat Model

- **SMs are tamper-proof.** It is hard for entities (including their users) to tamper with them successfully.
- **DCC, DNOs, TSO and suppliers are honest-but-curious.** They follow protocol specifications but may try to find out confidential data of users or (other) operators/suppliers.
- **External entities are not trustworthy.** They may intercept data in transit trying to access confidential (private) data.

### D. Smart Meter Architecture

MUSP introduces a single functional component, an accounting register $R_{acc}$, into the existing SM architecture (Fig. 2). Different from operational registers, $R_{op_1}, \ldots, R_{op_n}$, which are used for storing raw metering data, $R_{acc}$ stores user’s attributable metering data. Each operational register stores Meter Reading (MR) and timeslot, e.g. $\{MR_n \parallel T_n\}$, but $R_{acc}$ also stores the SM’s digital signature on the data, e.g. $\{MR_n \parallel T_n \parallel \text{Sig}_{SM}(MR_n \parallel T_n)\}$. $R_{op_1}, \ldots, R_{op_n}$ are updated per timeslot when the content of $R_{op}$ is shifted to $R_{op_{i+1}}$ and the freshest $\{MR_n \parallel T_n\}$ is stored in $R_{op_1}$, thus $R_{op_1}$ ($R_{op_n}$) stores the most (least) recent data available on the SM. $R_{acc}$ is regularly updated with data from $R_{op_1}$ and the update rate ($UR_{R_{acc}}$) is used as an input value of the application software.
Each SM is installed with the following secret keys: a symmetric key shared between the SM and its IHD (K_{BD}), a symmetric key shared between the SM and its local BG (K_{BG}) and its signature signing key (SK_{SM}). It is also loaded with the certificates of the regional DNO (Cert_{DNO}), the DCC (Cert_{DCC}), its user’s supplier (Cert_{S}), the local BG (Cert_{BG}) and its own certificate (Cert_{SM}). Each certificate contains the owner’s ID and public key. Each DNO’s certificate also contains the DNO’s homomorphic public key. In this work, we use the Paillier Cryptosystem [22] as it has an additive homomorphism property and is relatively efficient and semantically secure. Information about homomorphic encryption (including Paillier) schemes can be found in [23].

IV. THE MUSP SYSTEM

The MUSP system is operated by five protocols used for: (a) aggregated consumption data reporting to SG entities, (b) MR reporting to users, (c) attributable MR reporting to suppliers, (d) accounting register update and (e) supplier switching.

A. Aggregated Consumption Data Reporting to SG Entities

This protocol is designed to distribute users’ aggregated consumption data to authorised SG entities on a need-to-know basis at a high frequency. It is invoked at the beginning of each timeslot, and each protocol message contains the IDs of the message originator and recipient, timestamps along with the originator’s signature on the message. The message, upon reception, is verified by the recipient. For simplicity, the description below omits these details, but rather it focuses on the measures taken to ensure secure aggregation of users’ consumption data and secure reporting the aggregated data to the need-to-know SG entities (grid operators and suppliers). The protocol involves six major steps described next.

1) Consumption data report GEN (by SMs): Each SM, say SM_i, generates a report containing its user’s consumption data for the past timeslot and sends it to its local BG, i.e. SM_i
   a) reads MR_n and MR_{n-1} from registers R_{op1} and R_{op2} respectively, and calculates the user’s consumption data, CD_n = MR_n - MR_{n-1}, for the past timeslot;
   b) generates a random number, r_n;
   c) encrypts CD_n using its regional DNO’s homomorphic public key and r_n (used as an input value to the encryption algorithm too), i.e. C_{SM} = Enc(HPK_{DNO}, CD_n, r_n); this is to resist passive attacks by external entities;
   d) appends the user’s supplier’s ID to C_{SM} and encrypts the result using K_{BG}, i.e. C_{SM} = E(K_{BG}, {ID_S || C_{SM}}); this is to resist passive attacks by authorised entities;
   e) appends the ID of the regional DNO to C_{SM} and sends the result, i.e. M_{SM} = {ID_{DNO} || C_{SM}}, to the local BG.

2) Supplier-based data aggregation (by gateways): Each gateway, say BG_s, aggregates the respective sets of received ciphertexts based on the suppliers’ IDs attached to them, i.e. upon the receipt of a message M_{SM}, BG_s
   a) decrypts C_{SM}, i.e. {ID_{S} || C_{SM}} = D(K_{BG}, C_{SM});
   b) groups the ciphertexts with the same supplier ID;
   c) aggregates the ciphertexts in each group to form a single aggregated ciphertext for the group, i.e. C_{BG, S} = \prod C_{SM} and appends the corresponding supplier’s ID to C_{BG, S};
   d) constructs a message containing regional DNO’s ID and aggregated ciphertexts each appended with the corresponding supplier’s ID, M_{BG} = \{ID_{DNO} || ID_{S}, C_{BG, S_1}, \ldots, ID_{S_N} || C_{BG, S_N}\}, and sends M_{BG} to the local BG.

Each NG/WG does similar operations as those carried out by each BG except that it processes messages from BGs/NGs and skips the decryption step as gateways do not encrypt data.

3) Region-supplier-based data aggregation and distribution (by DCC): DCC, based on the DNOs’ and suppliers’ IDs contained in the received messages, aggregates the ciphertexts contained in the messages, and distributes the aggregated ciphertexts to the respective DNOs/suppliers. In detail, DCC
   a) sorts the received messages containing the same DNO ID into one group, thus forming N_d different groups;
   b) sorts the ciphertexts contained in the messages in each group into subgroups based on the supplier’s ID attached to each, thus forms N_i different subgroups in each group;
   c) aggregates the ciphertexts in each subgroup, e.g. C_{DNO, S} = \prod C_{WG, S}, thus forming (N_d x N_i) different aggregated ciphertexts, where C_{DNO, S} is the aggregated ciphertext of all the users located in the region operated by DNO_s and supplied by S_u;
   d) constructs a message for each DNO, that contains the supplier-based aggregated ciphertexts of the users living in the DNO’s region, e.g. M_{DCC,DNO} = \{ID_{DNO_s} || C_{DNO, S_1}, \ldots, ID_{S_N} || C_{DNO, S_N}\}, and sends it to the DNO;
   e) constructs a message for each supplier, which contains the region-based aggregated ciphertexts of all the supplier’s customers, e.g. M_{DCC,S} = \{ID_{DNO_s} || ID_{DNO, S_1} \ldots, ID_{S_N} || C_{DNO, S_N}\}, and sends it to the supplier.

Note that suppliers cannot decrypt these ciphertexts as they do not know the corresponding homomorphic private keys. These ciphertexts are used by the suppliers for data verifications (more details in Section IV-A5).

4) Region-supplier-based data recovery (by DNOs): DNOs recover aggregated consumption data (ACD) and send sets of data to the corresponding suppliers and TSO through secure channels. For this, each DNO, say DNO_s, does the following:
   a) for each supplier, e.g. S_u, it
      i. recovers the ACD by decrypting the respective aggregated ciphertext using its homomorphic private key, i.e. ACD_{DNO, S} = Dec(HSK_{DNO, C_{DNO, S}});
      ii. recovers the random number embedded in C_{DNO, S}, i.e. \{r_{DNO, S} \equiv Rn(HSK_{DNO, C_{DNO, S}}, ACD_{DNO, S})
      iii. constructs a message M_{DNO, S} = \{r_{DNO, S} || ACD_{DNO, S}\} and sends it to the supplier;
   b) sums all the ACD recovered, i.e. M_{DNO, TSO} = \sum ACD_{DNO, S}, and sends the result to the TSO.

5) Supplier-region-based data access and ciphertext-based data verification (by suppliers): Suppliers access and verify the ACD of their customers. Each supplier, e.g. S_u,
a) obtains the message, \( M_{\text{DCC},S_u} = \{ \| ID_{\text{DNO}_1} \| C_{\text{DNO}_1}, S_u \}, \ldots, \{ \| ID_{\text{DNO}_n} \| C_{\text{DNO}_n}, S_u \} \}, \) sent from the DCC;

b) for each DNO, e.g. \( \text{DNO}_1 \), it

i. obtains \( M_{\text{DNO}_1,S_u} = \{ \text{DNO}_1, \| ACD_{\text{DNO}_1}, S_u \} \}, \)

ii. encrypts \( ACD_{\text{DNO}_1}, S_u \) using the homomorphic public key of \( \text{DNO}_1 \) and \( r_{\text{DNO}_1}, S_u \), i.e. \( C'_{\text{DNO}_1} = \text{Enc}(HPK_{\text{DNO}_1}, ACD_{\text{DNO}_1}, S_u, r_{\text{DNO}_1}, S_u) \).

iii. verifies and accepts \( ACD_{\text{DNO}_1}, S_u \) if \( C'_{\text{DNO}_1} \) received in step a) is identical to \( C'_{\text{DNO}_1} \) computed above;

iv. sums the verified \( ACD, ACD_{\text{DNO}_1} = \sum_{j=1}^{N} ACD_{\text{DNO}_j} \).

6) Region-based data access (by TSO): Upon receiving the \( \text{ACD}s \) from all the \( \text{DNO}s \), TSO sums the \( \text{ACD}s \). \( ACD_{\text{TSO}} = \sum_{j=1}^{N} ACD_{\text{DNO}_j} \), to get the \( \text{ACD} \) of all the users in the grid.

B. Meter Reading Reporting to Users

MRs can be used for improving electricity usage awareness too by making them available to their respective users at a very high frequency, e.g. once every 30 sec. As shown in Fig. 3, this protocol has one message: a MR Report (\( \text{Rep}_{\text{MR}} \)).

1) \( \text{Rep}_{\text{MR}} \text{ GEN} \): SM generates and sends \( \text{Rep}_{\text{MR}} \) to its IHD.

In detail, SM reads the current MR, \( \text{MR}_n \), from its meteologic unit, encrypts it using the key shared with the IHD (\( K_{\text{IHD}} \)) to generate \( C_{\text{IHD}} = \text{Enc}(K_{\text{IHD}}, \text{MR}_n) \), and generates a key-hased message authentication code (\( \text{HAMAC} \)) on \( C_{\text{IHD}} \) also using \( K_{\text{IHD}} \), i.e. \( H_{\text{IHD}}(C_{\text{IHD}}) \). Then it sends the report, \( \text{Rep}_{\text{MR}} = \{ C_{\text{IHD}} \| H_{\text{IHD}}(C_{\text{IHD}}) \} \), to its IHD.

2) \( \text{Rep}_{\text{MR}} \text{ VER} \& \text{ MR}_n \text{ display} \): The IHD verifies \( \text{Rep}_{\text{MR}} \) and recovers \( \text{MR}_n \) using \( K_{\text{IHD}} \). It then displays (and stores) \( \text{MR}_n \). The user can access \( \text{MR}_n \) via IHD which is assumed to be in a physically secure place (inside the user's house).

C. Attributable Meter Reading Reporting to Suppliers

Attributable MRs are used for accounting (billing) purposes and released upon suppliers’ requests. As shown in Fig. 4, this protocol has two messages: an attributable MR Request (\( \text{Rep}_{\text{attr,MR}} \)) and an attributable MR Report (\( \text{Rep}_{\text{attr,MR}} \)).

1) \( \text{Rep}_{\text{attr,MR}} \text{ GEN} \): A supplier, \( \text{S} \), generates and dispatches \( \text{Rep}_{\text{attr,MR}} \) to an SM of its customer.

\( \text{S} \) generates randomly a session key (\( K \)), encrypts \( K \) using the SM’s public key, \( C_{S} = \text{Enc}(\text{PK}_{\text{SM}}, K) \), and signs the result using its private key to generate a request, \( \text{Rep}_{\text{attr,MR}} = \{ C_{S} \| \text{Sig}_{S}(C_{S}) \} \). Then it sends the request to the SM and starts a count-down timer that defines a timeslot in which \( S \) expects to receive a report from the SM. This measure is taken to resist denial-of-service attacks launched by malicious SMs.

2) \( \text{Rep}_{\text{attr,MR}} \text{ VER} \& \text{ Rep}_{\text{attr,MR}} \text{ GEN} \): SM verifies \( \text{Rep}_{\text{attr,MR}} \), generates \( \text{Rep}_{\text{attr,MR}} \) and sends it to \( \text{S} \).

Upon the receipt of \( \text{Rep}_{\text{attr,MR}} \), SM verifies its authenticity using the public key contained in the certificate of \( S \), \( \text{Cert}_S \), stored on SM, and recovers \( K \) using its private key, \( \text{SK}_{\text{SM}} \). If the request is authentic, SM then reads the data, \( \text{Data} = \{ \text{MR}_n \| T_n \| \text{Sig}_{\text{SM}}(\text{MR}_n \| T_n) \} \), from its \( \text{R}_{\text{acc}} \), encrypts the data using \( K \), i.e. \( C_{\text{SM}} = \text{Enc}(K, \text{Data}) \), and generates an HMAC on \( C_{\text{SM}} \) also using \( K \), i.e. \( H_{K}(C_{\text{SM}}) \). It sends the report, \( \text{Rep}_{\text{attr,MR}} = \{ C_{\text{SM}} \| H_{K}(C_{\text{SM}}) \} \), to \( \text{S} \).

3) \( \text{Rep}_{\text{attr,MR}} \text{ VER} \& \text{ Data access} \): \( \text{S} \) verifies \( \text{Rep}_{\text{attr,MR}} \) and recovers \( \text{Data} \) using \( K \). Note that \( \text{Data} \) contains attributable MR, i.e. MR timestamped and signed by the user’s SM.

It is worth noting that the approach used in this protocol simplifies data access management at SMs as an SM responds to every \( \text{Rep}_{\text{attr,MR}} \) sent by its user’s supplier and the generated report, \( \text{Rep}_{\text{attr,MR}} \), carries the same data until \( \text{R}_{\text{acc}} \) is updated regardless of the time and frequency of \( \text{Rep}_{\text{attr,MR}} \)’s arrivals.

D. Accounting Register Update

To balance privacy and functionality, MUSP is equipped with three ways of controlling the release of new attributable MRs to a supplier (via updating \( \text{R}_{\text{acc}} \) in an SM): \( \text{R}_{\text{acc}} \) Update Rate (\( \text{UR}_{\text{acc}} \) set by manufacturer), \( \text{UR}_{\text{acc}} \) set by user and \( \text{R}_{\text{acc}} \) update on demand.

1) \( \text{UR}_{\text{acc}} \text{ set by manufacturer} \): To ensure the basic functionality of the system with maximum protection of a user’s privacy, we should have a minimum \( \text{UR}_{\text{acc}} \) value set just sufficient for a supplier to accomplish its basic duty - generating accurate bills regularly. This value should be (i) agreed among regulators, suppliers and users, (ii) used as the default value and (iii) set by the SM manufacturer. For example, this value can be set to ‘once per month’, meaning that, by default, SM updates its \( \text{R}_{\text{acc}} \) only once per month, so the supplier can access the user’s new attributable MRs once every month.

2) \( \text{UR}_{\text{acc}} \text{ set by user} \): In addition to the minimum \( \text{UR}_{\text{acc}} \) value, there should also be a facility to allow an SM user to set \( \text{UR}_{\text{acc}} \) to a higher value, e.g. ‘once a day’. Obviously, the higher the \( \text{UR}_{\text{acc}} \) value, the more frequently the supplier can pull new attributable MRs from the SM, and the lower the privacy protection level. As depicted in Fig. 5, this facility is initiated by the user and notifies the supplier and DCC. Also, each message should be encrypted with the intended recipient’s public key and signed with the sender’s private key.

a) A user (User) and her supplier (S) establish a secure channel through which User requests a new \( \text{UR}_{\text{acc}} \) value.

b) S sends a message to DCC, and the message contains the user’s SM ID (\( \text{ID}_{\text{SM}} \)) and the requested \( \text{UR}_{\text{acc}} \) value.

c) DCC receives the message, generates a unique user code (\( U_{\text{code}} \)) and sends a message containing \( U_{\text{code}} \) and the new \( \text{UR}_{\text{acc}} \) value to the user’s IHD (via the user’s SM).

d) User’s IHD receives the message sent by DCC and displays \( U_{\text{code}} \) and the new \( \text{UR}_{\text{acc}} \) value.
Fig. 5. An outline of the accounting register UR-change protocol.

e) User verifies that the UR\_acc value displayed is the same as the value initially requested, and sends U\_code to S through a new or already established secure channel.
f) S receives U\_code and forwards it to DCC.
g) DCC receives U\_code sent by S, verifies that U\_code received is the same as U\_code sent to User, and sends an update to SM containing the new UR\_acc value.
h) SM replaces the existing UR\_acc value used as input into its application software with the new UR\_acc value and sends an acknowledgement (Ack\_upd) to DCC.
i) DCC forwards Ack\_upd to IHD and S.

Note that as the supplier is aware of the UR\_acc value, it does not need to be notified when a scheduled R\_acc update occurs, so MUSP introduces a low level of communication overheads. It also supports smart report timing as the supplier can schedule the times it sends requests to its customers’ SMs according to their R\_acc update rates, and still obtain the attributable MRs measured at the same time from all its customers.

3) R\_acc update on demand. There are times when a supplier may need to access new attributable MRs on demand (outside the schedule times), e.g. when there is a tariff/account holder change. In such cases, a new attributable MR release should be initiated by the user (as this should be approved by the user on per release basis) and the supplier should be notified of this release. Note that such updates will not affect the regular R\_acc update times set beforehand. The protocol is shown in Fig. 6.

a) User initiates the update on demand protocol by choosing the corresponding option from the menu of her IHD. This option could also be username/password protected.
b) IHD generates, encrypts and integrity protects (using K\_IMD) an update command (Cmd\_upd) and sends it to SM.
c) SM verifies Cmd\_upd, updates its R\_acc and sends an encrypted and signed notification (Notif\_upd) to S.

g) DCC receives Notif\_upd and forwards it to S.
h) S receives Notif\_upd, verifies that it is sent by the user (as this should be approved by the user), and forwards it to DCC.
i) DCC forwards Notif\_upd to SM.
j) SM replaces old UR\_acc with new UR\_acc, Ack\_upd.
k) SM sends an acknowledgement (Ack\_upd) to DCC.
l) DCC forwards Ack\_upd to User.
m) User receives Ack\_upd and forwards it to SM.

Fig. 6. An outline of the update-on-demand protocol.

E. Supplier Switch

MUSP also allows users to switch suppliers easily and securely. The protocol supporting this is outlined in Fig. 7.

a) A user (User) and a new supplier (S\_new) establish a secure channel through which User sends S\_new her data, including name (U\_name), address (U\_add), any contract details, e.g. binding duration, tariff, UR\_acc value.
b) S\_new sends a message to DCC containing the request to switch supplier (Req\_sw), U\_add and the agreed UR\_acc value.
c) DCC receives the message, identifies the user’s SM based on U\_add, generates a unique user code (U\_code) and sends a response to the user’s IHD containing U\_code, S\_new’s ID (ID\_new) and UR\_acc value.
d) IHD displays the response: U\_code || ID\_new || UR\_acc.
e) User verifies (1) ID\_new displayed on IHD is the ID of S\_new and (2) UR\_acc displayed on IHD is the same as UR\_acc initially agreed with S\_old.
f) S\_new sends U\_code and forwards it to DCC.
g) DCC receives U\_code, verifies that U\_code is the same as U\_code sent to User, and sends the user’s IHD (via SM), current supplier’s (S\_old) and S\_new information about the switch (Inf\_sw) including the date/time of the switch (T\_sw).
h) SM updates its registers (incl. R\_acc) at T\_sw as scheduled.
i) S\_old pulls the (final) MR from SM using the attributable MR reporting as in Section IV-C. This is successful as SM still stores its user’s current supplier’s certificate (Cert\_old).
j) DCC sends an update to SM containing the certificate of S\_new (Cert\_new) and the (new) UR\_acc value.
k) SM replaces (1) Cert\_old with Cert\_new on its storage and (2) the existing UR\_acc value with the new UR\_acc value, and sends an acknowledgement (Ack\_sw) to DCC.
l) DCC receives and forwards Ack\_sw to S\_new.
m) S\_new pulls the (initial) MR from SM successfully using the attributable MR reporting as SM now stores Cert\_new.
Note that the supplier switching process is easy and simplified as only the certificate of the current supplier stored on an SM needs to be replaced with the certificate of the new supplier. No change of cryptographic secret keys is required.

V. Security Analysis

A. Aggregated Consumption Data Reporting

Each individual user’s consumption data is double encrypted (first with the regional DNO’s homomorphic public key, then with the key shared between the user’s SM and its local BG, i.e. $K_{BG}$) while in transit between the SM and BG, thus (unlike other aggregating schemes) making our protocol resistant to passive eavesdropping attacks by both outsiders and authorised insiders. Each authorised insider (i.e. DNOs, suppliers, the TSO) can only access the aggregated data of their respective users. Assuming that the received aggregated data contains the individual data of sufficient number of users, such that it is difficult for the data recipients to decompose the aggregated data into individual users’ data, users’ privacy is protected.

B. Meter Reading Reporting

MRs from an SM are encrypted and integrity protected using a key, $K_{IHD}$, shared only between the SM and its IHD. So, only the users who have access to the IHD could access the MRs. Assuming that the IHD is located in a physically secure location (inside the user’s house), only authorised users can access MRs at a high frequency, thus preserving users’ privacy. Obviously, accessing IHD can also be username/password controlled, ensuring that no user, rather than the one knowing the password (i.e. account holder), can gain access to the MRs.

C. Attributable Meter Reading Reporting

Attributable MRs from $R_{acc}$ of an SM are encrypted and integrity protected using a session key, $K$, shared only between the SM and its IHD. So, only the users who have access to the IHD could access the MRs. Moreover, both the SM and the supplier are resistant to denial-of-service attacks as the SM only accepts requests sent by its user’s supplier and the supplier only accepts reports sent by the SM to which requests have been sent. However, as the supplier is allowed to obtain data from $R_{acc}$ as many times as it wants, to protect the user’s privacy, the content of $R_{acc}$ should be updated at a low frequency. This can be achieved by setting $UR_{acc}$ to a low value, e.g. one update per month, by default. This setting will assure the supplier that it can access its user’s new attributable MRs, at least once a month, without the user’s involvement. Also, having $R_{acc}$ allows users to adjust their own $UR_{acc}$ values, thus managing their own privacy protection levels.

D. Updates to SMs

Any change made to $UR_{acc}$ (or a supplier switch) is verified by the user with the help of a code generated by a trusted third party, i.e. DCC. As long as DCC generates an unique user code every time it receives a request made by a supplier, DCC can be assured that the request is indeed from a legitimate user.

VI. Performance Evaluation

This section evaluates the performance of the MUSP system. The computationally expensive operations used in the MUSP protocols are asymmetric encryption/decryption and digital signature generation/verification.

A. Aggregated Consumption Data Reporting

During each timeslot, an SM performs one asymmetric encryption and sends one message (thus it generates one digital signature) to its local BG, a gateway verifies one message per child entity and sends one message (thus it generates one digital signature) to its parent entity, and DCC verifies one message per child WG and sends one message to each DNO and each supplier (thus it generates $N_d + N_s$ digital signatures).

We ran experiments with pbc [24] and miracl [25] libraries on 3.0 GHz-processor and 4 GB-memory machine to study the operational costs. Our results indicate that asymmetric (homomorphic) encryption (i.e. Paillier encryption with $|n|^2 = 2,048$ [22]) costs 84.4 $\mu$s, digital signature (i.e. Boneh-Lynn-Shacham (BLS) signature [26]) generation 43.5 $\mu$s and verification 136.1 $\mu$s. Based on the electrical grid size in the UK [27], we set $N_d = 14$ ($N_d$ - number of DNOs in the grid), $N_{wg} = 140$ ($N_{wg}$ - number of WGs in the grid), $N_{ng} = 32$ ($N_{ng}$ - number of NGs connected to each WG), $N_{bg} = 28$ ($N_{bg}$ - number of BGS connected to each NG), and vary $N_{sm}$ from 100 to 420 ($N_{sm}$ - number of SMs connected to each BG) and $N_s$ from 1 to 33 ($N_s$ - number of suppliers in the electricity market). Fig. 8 depicts the variation of computational costs at different entities vs. $N_{sm}$ and $N_s$. Adding more SMs to the grid will affect (increase linearly) only the computational costs at BGs. Introducing new suppliers to the electricity market will only affect (increase slightly) the computational cost at DCC. Thus, our protocol, in contrast to other work [17], [18], scales well in terms of SMs and suppliers. It is also worth noting that each SM generates only one report in each timeslot, but still the consumption data included in the report is delivered (in an aggregated form) to three different entities: SM’s regional
at the WGs−to−DCC part
at each NGs−to−WG part
at each BGs−to−NG part

Communication overhead (bits)

}\end{figure}

This research is supported by the Engineering and Physical Sciences Research Council (EPSRC) and Toshiba Research Europe Limited under Grant [EP/IS01541/1].

REFERENCES


