VerSAMI: Versatile and Scalable key management for Smart Grid AMI systems

Mourad Benmalek, Yacine Challal, Abdelouahid Derhab, Abdelmadjid Bouabdallah

VerSAMI: Versatile and Scalable key management for Smart Grid AMI systems

Mourad Benmalek[a,*], Yacine Challal[a,b,c], Abdelouahid Derhab[d], Abdelmadjid Bouabdallah[c]

[a] *Laboratoire de Méthodes de Conception de Systèmes, Ecole nationale Supérieure d'Informatique, Algiers, Algeria*
[b] *Centre de Recherche sur l'Information Scientifique et Technique, Algiers, Algeria*
[c] *Université de Technologie de Compiègne, Heudiasyc UMR CNRS 7253, Compiègne, France*
[d] *Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia*

Abstract

In this paper, we propose four new key management schemes for Advanced Metering Infrastructure (AMI) to secure data communications in the Smart Grid (SG). The schemes are based on individual and batch rekeying operations using a novel multi-group key graph structure, are also versatile in the sense that they can support broadcast, unicast, as well as multicast communications. Security analysis shows that our schemes satisfy key management security properties. Furthermore, performance analysis and simulation results demonstrate that our schemes can scale to large AMI systems, and they incur low cost in terms of storage, communication overheads and time of key distribution compared to state-of-the-art schemes.

Keywords: Smart Grid, Advanced Metering Infrastructure, Cyber Security, Key Management Scheme, Individual rekeying, Batch rekeying.

1. Introduction

The Smart Grid (SG), also called intelligent grid, intelligrid or futuregrid [2], refers to a modernization of electric grid that incorporates information, advanced two-way communication and computational intelligence to the electricity generation, distribution and management, in order to improve control, efficiency, agility, economy, reliability, security and safety [2–4]. The smart grid provides many benefits such as: (1) the greater availability of electricity to homes at a lower cost, (2) the incorporation of renewable power generation such as wind and solar power into the grid, and (3) the improvement of the grid security and self-healing [5]. A succession of sub-systems should be realized in order to achieve an efficient, intelligent and sustainable grid [6]: Advanced Metering Infrastructure (AMI), Advanced Distribution Operations (ADO), Advanced Transmission Operations (ATO) and Advanced Asset Management (AAM).

*Corresponding author
Email addresses: m_benmalek@esi.dz (Mourad Benmalek), y_challal@esi.dz (Yacine Challal), abderhab@ksu.edu.sa (Abdelouahid Derhab), abdelmadjid.bouabdallah@utc.fr (Abdelmadjid Bouabdallah)

Advanced Metering Infrastructure (AMI) has been regarded as the essential component of the SG [7]. It is responsible for recording customer's consumption and transmitting that information back to the AMI host system for monitoring and billing purposes [8]. It is also responsible for implementing control commands (e.g., to disconnect-reconnect remotely and control the customers' devices and appliances in order to manage loads and demands) and price signals (e.g., to encourage customers to reduce their consumption in peak load periods according to these price signals sent from the AMI host system) [6]. An important feature in AMI systems that will make the smart grid more reliable and lead to mutual benefits for both customers and power utility is the Demand Response (DR) program [8]. DR programs can be considered as (a) tariffs or (b) utilities turn off customers' appliance, offered by power utilities to incentivize customers to individually and voluntarily manage their loads [9], e.g., by reducing their consumption at peak hours. A customer can subscribe to a variety of DR programs such as: Time of Use (TOU) program, Real Time Pricing (RTP) program, Critical-Peak Pricing (CPP) program, etc.

Given this cornerstone role of AMI, it becomes a privileged target for attackers, which potentially cause great damage against the infrastructures and the user's privacy. Consequently, security is one of the most challenging topics in AMI development. Many security requirements in AMI are the same as those of typical IT networks. Some of the key AMI security requirements are [7, 10, 11]: confidentiality, integrity, availability, and accountability (non-repudiation). To meet these security requirements and ensure secure communications in AMI, cryptographic countermeasures must be deployed and keys need to be set up to ensure authenticity and secrecy. However, cryptographic mechanisms for AMI require also an efficient key management (generation, distribution, and update). Inadequate key management can result in possible key disclosure to attackers, and even jeopardizing the entire goal of secure communications in AMI. Moreover, for the large number of devices (millions of smart meters) with constrained resources (e.g., the limited storage capacity of smart meters [12–15]), it is important for the key management protocol to be scalable. Furthermore, given the different modes of message transmission (unicast, multicast, and broadcast) normally used in AMI, versatility is required for the key management protocol, which means the ability to support all the three modes of message transmission. Many key management schemes (KMS) have been proposed [15–30], but none of them can completely support all the three modes of message transmission with efficiency and scalability.

Contributions. In this paper, we propose four new key management schemes that can simultaneously ensure *security*, *scalability*, *efficiency*, and *versatility*. The contributions of this work are as follows:

- We propose a new key **Ver**satile and **S**calable key management scheme for **AMI**, called, VerSAMI, to better support secure unicast, multicast, and broadcast communications for a large-scale AMI system. Moreover, VerSAMI supports the management of multiple DR programs; we envision that many customers should be able to subscribe to multiple DR programs simultaneously and flexibly

subscribe/unsubscribe to any DR program at any time (a DR program is equivalent to a multicast group). This is accomplished, unlike some other schemes based on independent-tree schemes, through our new multi-group key graph technique that efficiently handles the rekey operations, while meeting smart meters limitations in terms of memory and bandwidth capacities.

- We also propose an improved version of VerSAMI, called, VerSAMI+, which provides enhancement in the communication overhead.
- To overcome the possible problems of VerSAMI (resp. VerSAMI+) based on individual rekeying, and to reduce the number of rekeying operations, we propose an effective variant of VerSAMI (resp. VerSAMI+), called Batch-VerSAMI (resp. Batch-VarSAMI+), in which membership changes are handled in batches instead of handling individually.
- Unlike the earlier proposed schemes that validate their performance results through complexity analysis; we additionally propose a dynamic membership model; which simulates the AMI system behavior.
- A security and performance analysis, as well as simulations and comparison with state-of-the-art schemes have been conducted to prove the efficiency, and the security of the proposed schemes.

The rest of this paper is organized as follows: we discuss related work on key management in Section 2. In section 3, we provide background on the architecture of AMI system, interactive messages and key management function requirements. After that, we present our versatile and scalable key management schemes VerSAMI and VerSAMI+ in section 4. In section 5, we describe the proposed batch-rekeying schemes (Batch-VerSAMI and Batch-VerSAMI+), and discuss the security and performance analysis of all the schemes in sections 6 and 7. In section 8, we compare our schemes with two existing schemes [22, 23] and give a full discussion using both security and performance metrics. Finally, we draw our conclusions in section 9.

2. Related Work

According to [24], key management has been identified as a critical process to ensure the secure operations of AMI. As a recent NIST (National Institute of Standards and Technology) report [31], many secure communication scenarios in an AMI are required, and for all of them, key management needs to be set up to ensure authenticity and secrecy. In recent years, many key management schemes have been proposed to secure communications for AMI in smart grid. Here, we review these efforts and critically analyze some of them.

Kamto *et al.* [16] proposed a key distribution and management scheme for large customer networks to achieve authentication, privacy and data confidentiality in AMI. The proposed scheme is based on the Diffie-Hellman (DH) [32] key exchange and group ID-based mechanism [33], which induces a very high computation

overhead. Moreover, this scheme is not resistant to the man-in-the-middle (MITM) and desynchronnization attacks, and only secures communications between HAN (Home Area Network) devices and the gateway, and does not support other communication modes such as multicasting and broadcasting (i.e., is not versatile)

In order prevent message forgery attacks in multicast communications, Li and Cao [17] proposed a one-time signature scheme. The proposed scheme presents a significant reduction in the storage and communication overhead, but does not secure unicast and broadcast communications. Furthermore, this scheme does not address confidentiality and only focuses on communication integrity.

Nicanfar *et al.* [18] developed a key management protocol for data communication between the utility server and customers' smart meters based on an ID-based public/private key pair model [33]. Although this protocol aims to reduce the computation overhead, the synchronization process still demands considerable computation efforts. Wu and Zhou [19] proposed a novel key management scheme for smart grid combining a symmetric key technique based on the Needham-Schroeder authentication protocol [34] and an elliptic curve public key technique [35] to provide strong security, efficiency and fault-tolerance. The authors used three authentication protocols for different parts of AMI network. Xia and Wang [20] showed that Wu and Zhou scheme [19] is vulnerable to the MITM attack and proposed an improvement for this scheme based on a trusted third party. However, these two schemes do not support secure multicast communications that play an important role and have wide applications in smart grid. A new Integrated Authentication and Confidentiality (IAC) protocol is proposed by Ye Yan *et al.* [21] in order to offer trust services, data privacy, and integrity by mutual authentications.

Liu *et al.* [22] developed a new key management scheme to secure unicast, broadcast, and multicast communications in AMI. In spite of using key graph structure [36], the proposed scheme suffers from a lack of scalability due to inefficient key management, which results in non-negligible communication overhead for such a large-scale system. Furthermore, Liu's *et al.* [22] scheme is not tolerant to packet loss. Wan *et al.* [23] proposed an improvement for Liu's *et al.* scheme, called SKM, which combines identity-based cryptosystem [37] and One-way Function Trees (OFT) approach [38] for multicast key management. The use of an OFT separately for each DR program results in non-negligible overhead for keys storage at smart meters that have limited storage capacities. Nabeel et *et al.* [24] proposed a PUF-based (Physical Unclonable Function) key management scheme for AMI. Although their scheme supports decentralized key management, the scheme requires a PUF hardware device.

Lately, Liu *et al.* [15] developed a lightweight authenticated communication scheme based on the bit-wise exclusive-OR and Lagrange interpolation formula to ensure secure two-way communication between the smart meters and the neighborhood gateway. Mohammadali *et al.* [25] proposed a novel secure and lightweight identity-based key establishment protocol, called NIKE. The authors proposed two variations for different AMI setups, and prove that their scheme induce low computational overhead compared to some secure key establishment protocols.

Moreover, some scalable and efficient certificate revocation schemes have been proposed [26–30]. In [26], Mahmoud *et al.* proposed a scheme based on the compressed Certificate Revocation Lists (CRLs) for pseudonymous public key infrastructure. Authors showed that the proposed scheme is secure and the size of the CRL is linear with the number of revoked certificate series. This scheme has been applied to the vehicle-to-grid communication application [27]. In [28], Mahmoud *et al.* developed a Bloom filter based scheme for the efficient certificate revocation for AMI networks. The proposed scheme is constructed based on the Merkle tree to enable the gateway to provide proof for certificate revocation without contacting the certificate authority (CA) in achieving significant overhead savings. Rabieh *et al.* [29] proposed another certificate revocation scheme based on Bloom filters in AMI networks. The authors explained that the size of CRLs can be reduced by Bloom filter in order to improve clusters' size with acceptable overhead. Two revocation schemes were proposed for addressing the false positive issue of the Bloom filter. In another work, Akkaya *et al.* [30] proposed an efficient grouping algorithm to distribute CRLs in an 802.11s based AMI networks. This algorithm is based on the fact that smart meters are communicating with the gateway to send/receive utility information. Each smart meter uses the same route to communicate with the gateway (i.e. communicate with the same group of smart meters for forwarding its packets). Thus, instead of storing a large CRL in each smart meter, smart meters only keep the CRL of its group to minimize the communication and storage overhead of CRL.

Although there are a lot of key management schemes, each has tried to achieve a specific security goal. Moreover, some schemes are not versatile and do not secure multicast or broadcast communications that are frequently used in AMI networks, and some others result in non-negligible communication or storage overhead. In this work, we propose novel versatile, efficient, flexible and scalable key management schemes that meet the security requirements of AMI.

3. AMI Features and Key Management Design Objectives

In this section, we provide background on AMI system architecture, interactive messages exchanged via AMI networks to identify the basic requirements that are relevant to key management.

3.1. Basic Components of an AMI

AMI system consists of several cyber-physical components, that are connected through various communication networks at different levels of the infrastructure hierarchy. A typical AMI involves (Figure 1):

- **Smart Meters (SMs):** Which are advanced electrical meters that play an essential part in AMI's two-way communication between customers and the power utility: (1) they measure customer consumption during a certain time interval and transmit it to the power utility, and (2) they also allow dynamic pricing and accept control commands and price signals and act accordingly.

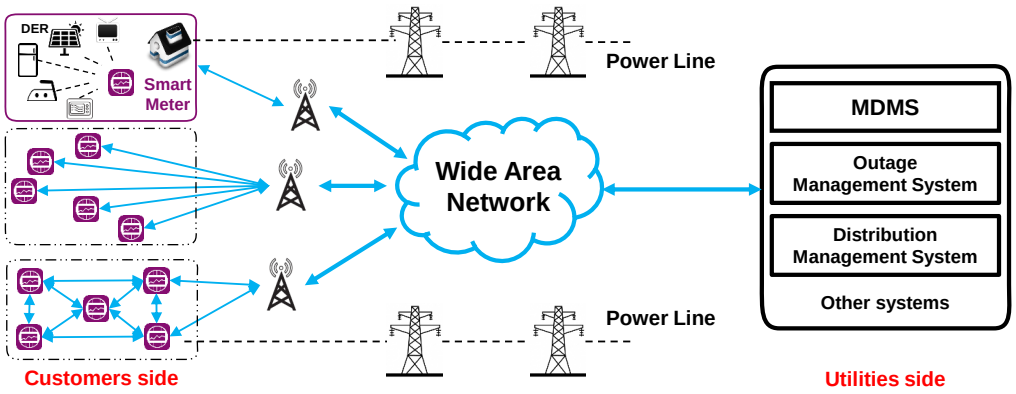


Figure 1: Basic Components of an AMI System.

- **AMI Wide Area Network:** It supports the two-way communication between the power utility control system and customers. The implementation of the wide area network for AMI could include different architectures and medias, such as cellular networks, power line communication systems, and IP-based networks [39].
- **Meter Data Management System (MDMS):** It is the central component of the management system. It acts as a database system for managing, storing and analyzing customer metering data. It should be able to address: (1) optimization and improvement of the utility operation, (2) optimization and improvement of the utility management, and (3) improvement of dynamic pricing and DR programs to enable better customer service.

3.2. Demand Response (DR) Programs

DR programs are tariffs offered by the power utility to customers to improve the grid efficiency and reliability [9]. In other words, it can be considered as changes in power consumption by customers from their normal consumption patterns in response to change in tariffs offered by the power utility to encourage customers to individually and voluntarily manage their energy loads. Even if several classifications of DR programs can be found, DR programs can mainly be divided into the two following groups:

1. **Incentive-based (event-based) DR programs:** In this category of DR programs, customers are rewarded for reducing their power loads upon the utility request or for giving the power utility some level of control over their power appliances (e.g., water heater, and air conditioner). Several DR programs can be cited as examples of programs in this category, such as: *Direct Load Control (DLC) program* [41], *Emergency Demand Reduction (EDR) program* [42], etc.
2. **Price-based (rate-based) DR programs:** In this category of DR programs, the power utility provides customers with different power prices at different times which, indirectly, make the customers

to change their power usage patters according to the changes of power prices. Several DR programs are proposed in this category, such as: *Critical Peak Pricing (CPP) program*, [43], *Real-Time Pricing (RTP) program* [44], etc.

3.3. AMI Interactive Messages Characteristics

Typical AMI applications include several interactive messages, such as meter reading, pricing signals, release of DR programs, joining/leaving a DR program, DR control commands, etc. These interactive messages among entities in an AMI system can be categorized into broadcast, multicast, and unicast according to their transmission mode. In Table 1, we summarize some important characteristics of AMI interactive messages that are relevant to key management [22, 40].

3.4. Key Management Design Objectives

As the key management function is a critical process in AMI security, we discuss in what follows the basic security and performance requirements of an effective key management scheme of AMI:

3.4.1. Security Design Objectives

- **Backward secrecy:** As the number of customers participating in DR programs is not fixed, any customer can join at any time any DR program. For this reason, it is obvious that this customer must not learn previously used secret keys [36].
- **Forward secrecy:** It means that a customer who leaves a DR program should not learn the future secret keys and messages [36].
- **Collusion freedom:** It means that any set of customers that leave a DR program should not be able to deduce the current used secret key through collusion.

3.4.2. Non-security Design Objectives

- **Versatility:** It means that the key management scheme should support the three modes of message transmission in AMI: unicast, broadcast, and multicast. Thus, for each mode, key generation, storage, distribution and update must be designed clearly.
- **Scalability:** The key management scheme should scale to the large size of SG that may comprise millions of SMs, without severly affecting the performance of the system.
- **Efficiency:** The key management process should be memory-usage and computationally efficient meeting the resources limitation of smart meters. Key generation, distribution, usage, and update processes should also induce low communication overhead, which is important to time-critical communication scenarios in AMI.

Table 1: AMI Interactive Messages Characteristics

Message	Sender	Receiver	Transmission mode	Latency Requirement
Meter reading	SM	Utility	Unicast	Low
Programs/Configuration update	Utility	SMs	Broadcast	Low
Join/Leave a DR program	SM	Utility	Unicast	Low
Electricity pricing	Utility	SMs	Broadcast / Multicast	Low
Remote Load Control commands	Utility	SMs	Multicast / Unicast	High
Notifications	Utility	SMs	Broadcast	High

4. VerSAMI : Versatile and Scalable Key Management for AMI

In this section, we present our new scalable and efficient key management scheme that we call **Ver**satile and **S**calable Key management for smart grid **AMI** networks. VerSAMI is based on a novel multi-group key graph structure that supports the management of multiple DR programs simultaneously for each customer.

In addition to secrecy, scalability, and efficiency, *flexibilty* is an indispensable feature: a customer can subscribe/unsubscribe to any DR program and make changes of his subscriptions at any time. We will demonstrate later that this new structure scales to large AMI with dynamic DR programs membership while meeting smart meters constraints in terms of storage and bandwidth capacities.

4.1. System Architecture and Assumptions

- The Advanced Metering Infrastructure complies with the architecture illustrated in Figure 1. The MDMS denotes the management side, and we assume that the MDMS is well protected from attacks and is responsible for key management services (key generation, distribution, and rekeying).
- A specific default DR program is mandatory for all customers of the AMI, i.e. all customers are subscribed to this default DR program, denoted DR_0. This mandatory DR program will be used by MDMS to broadcast some information messages or control commands to all customers of the SG.
- Except for the mandatory DR program, any customer can join or leave any DR program at any time.

4.2. Key Management Scheme Initialization

In order to initialize the key management scheme, let us consider a set of n customers $\{C_1, ..., C_n\}$, and m proposed DR programs. Initially:

- A specific method of securely exchanging cryptographic keys over a public channel is used to establish individual keys $\{k_1, ..., k_n\}$ between the MDMS and smart meters. For instance, Elliptic Curve Diffie-Hellman (ECDH) key agreement [45] can be used since it is known to induce less overhead compared to

many existing end-to-end key establishment using standard Diffie-Hellman protocol. These individual keys (which are refreshed periodically) will be used to secure unicast communications between the MDMS and smart meters.

- The established individual keys will also be used to generate the multi-group key graph for secure multicast communications between the MDMS and the smart meters. $\{GK_1, ..., GK_m\}$ denotes the m group keys of DR programs.
- In VerSAMI, GK_0 denotes the group key of the default DR program, this group key (which is refreshed periodically) must be generated by the MDMS and transmitted through secure channels for each smart meter, to be used for secure communications in the broadcast mode.

In Table 2, we summarize the terminology that we will use throughout the remaining of this paper.

4.3. Group Key Management

4.3.1. Logical Key Hierarchy (LKH)

Group key management is an important functional building block for any secure multicast architecture. Logical Key Hierarchy (LKH) protocol [36] is one of the relevant group key management protocols. In LKH, the key server maintains a tree of keys. Each member holds a copy of his leaf secret key and all the keys corresponding to the nodes in the path from his leaf to the root. The key corresponding to the root of the tree is the group key. For a balanced binary tree, each member stores at most $1 + log_2(n)$ keys, where n is the number of group members. This key hierarchy allows to reduce the number of re-key messages to $O(log\ n)$.

To briefly describe how LKH works, let us consider a group of six customers $\{C_1, C_2, C_3, C_4, C_5, C_6\}$. The key server builds a hierarchy of keys as shown in Figure 2. Each customer owns a secret key which is a leaf in the tree as well as all the keys on its path to the root. The root represents the group key shared by the customers. The other keys are used to reduce the required rekeying messages. According to Figure 2: C_1 owns $\{k_1, k_{12}, k_{14}, GK\}$, C_2 owns $\{k_2, k_{12}, k_{14}, GK\}$, C_3 owns $\{k_3, k_{34}, k_{14}, GK\}$, C_4 owns $\{k4, k_{34}, k_{14}, GK\}$, C_5 owns $\{k_5, k_{56}, GK\}$ and C_6 owns $\{k_6, k_{56}, GK\}$. Let us assume that C_5 leaves the group, the server changes k_{56} to k'_{56}, sends k'_{56} to C_6 encrypted with k_6. GK is changed to GK' and sent to $\{C_1, C_2, C_3, C_4\}$ encrypted with k_{14} and to C_6 encrypted with k'_{56} and hence only three messages are required.

4.3.2. Multi-group Key Graph Structure

We adopt LKH for its security and efficiency for group key management in terms of storage and communication overhead, and also for its flexibility approach that allows customers to join and leave DR programs

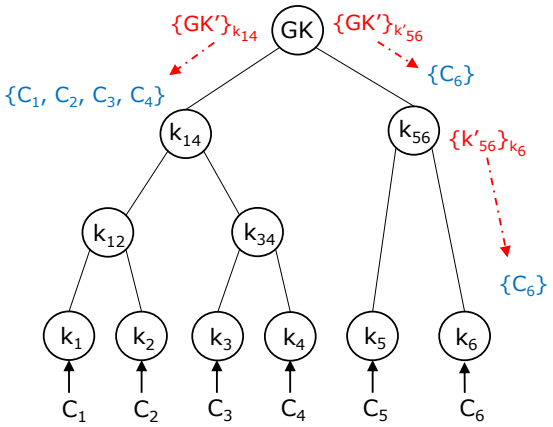


Figure 2: Logical Key Hierarchy.

at any time. However, as the customers can subscribe to multiple DR programs at the same time, an intuitive solution is to use an LKH key tree for each DR program. Hence, if a customer (e.g., C_1) subscribes to two DR programs simultaneously (e.g., DR_1 and DR_2), he needs to manage two sets of keys. As a result, directly applying LKH is costly and induces a non-negligible overhead for key storage. To reduce this cost, we propose a new key graph structure, so that a customer needs to manage only one set of keys.

The idea of our new key graph technique is to allow multiple DR programs to share a new set of keys. For instance, a customer C_1 in $DR_1 \cap DR_2$ does not need to manage two sets of keys to handle the two DR programs. Indeed, in our solution C_1 will only hold the keys on his path to the group key corresponding to his last subscription. Moreover, when a customer joins or leaves a DR program, the communication overhead for rekeying operations will be significantly low compared to the overhead induced by using separate LKH trees inside each DR program. As illustrated in Figure 3, the proposed multi-group key graph structure can be modeled as follows:

- In the *lower level*, each LKH tree represents a set of customers with the same first DR program subscription: (1) the leaf node of the tree is the customers' individual key, and (2) the root of the key tree is associated with the DR_i program group key GK_i.

- In the *upper level*, the graph represents combinations of root keys for customers subscribing to multiple DR programs at the same time.

Our key structure has the following properties:

- **Property 4.1.** *A customer only belongs to one LKH tree in the multi-group key structure corresponding to his Home DR program (initial subscription). He holds a copy of his individual secret key (leaf key) and all keys corresponding to the nodes in the path from his leaf to the root in this tree.*

Table 2: Notation Table

Notation	Description
$H(.), f(.)$	One-way hash functions
n	Total number of customers
m	Number of DR programs
C_i	The i^{th} customer
DR_i	The i^{th} DR program
m_i	Number of the i^{th} DR program subscribers
n_i	Number of customers with the same Home DR program DR_i
d	Key tree degree
h_i	Height of the i^{th} tree
GK_i	The i^{th} DR program group key
$\|K\|$	The size of keys in bit
$sub(C_i)$	Number of DR programs to which C_i subscribes
$Home\_DR(C_i)$	First DR program to which C_i subscribes
$member(i, j)$	The number of subscribers to the i^{th} DR program, having DR_j as Home DR program
$set(C_i)$	Set of DR programs to which C_i subscribes
$Child_i(GK_j)$	The i^{th} child of (GK_j) in key tree
$right(k_i)$	Right children of node k_i in the key tree
$left(k_i)$	Left children of node k_i in the key tree
$Path(C_i)$	Keys corresponding to the nodes in the path from C_i's individual key to group key
$a\|\|b$	A concatenation between a and b
$Enc(M, k)$	Message M encrypted with key k
$\oplus$	Mixing function: bitwise exclusive-or (XOR)
$A \rightarrow B : M$	A sends a message M to B

- **Property 4.2.** *A customer has all group keys of the other DR programs to which he is subscribed.*
- **Property 4.3.** *If a customer leaves his Home DR program and remains subscribed to one or more DR programs, he will shift to a new LKH tree (this will be the tree corresponding to his new Home DR program).*

In Figure 3, an example of the multi-group key graph is given. The utility provides 4 DR programs DR_1, DR_2, DR_3, DR_4. Some customers subscribe to only one DR program (e.g. C_2 subscribes only to DR_1), while other customers may subscribe to multiple DR programs simultaneously (e.g. C_1 subscribes to

DR_1, DR_2, and DR_3). In this figure, no customer subscribes to both DR_3 and DR_4 at the same time.

Next, we illustrate both member join and leave procedures executed by the MDMS when receiving a member join or leave request.

4.3.3. Leave operation

The leave operation deals with the case when a customer C_i unsubscribes from a DR program DR_j.

Let $\phi_j = \{C_l / C_l \text{ is subscribed to } DR_j\}$,

Let $\chi_{jj} = \{C_l / C_l \in \phi_j \text{ and } Home\_DR\ (C_l) = DR_j\}$,

Let $\chi_{jk} = \{C_l / C_l \in \phi_j \text{ and } Home\_DR\ (C_l) = DR_k \text{ and } DR_j \neq DR_k\ \}$,

Let $\omega_{jk} = \{C_l / C_l \in \chi_{jk} \text{ and } DR_k \in set(C_i)\}$,

Let $\delta_{jk} = \{C_l / C_l \notin \chi_{jk} \text{ and } DR_k \in set(C_i)\}$.

Two cases are possible:

- **Case 1:** When a customer C_i, who subscribed to one/multiple DR programs, leaves his Home DR program DR_j: The MDMS renews and updates keys according to Algorithm 1.

Algorithm 1 Update keys, when a customer leaves his Home DR program

procedure LEAVEHOMEPROGRAM (C_i, DR_j)
  Renew and update GK_j (GK'_j is the new group key)
  Apply the standard LKH approach in DR_j tree
  if $sub(C_i) = 1$ **then** ▷ *Customer subscribed to one DR program*
    MDMS $\rightarrow \chi_{jk}$:

$$\bigcup_{\substack{1 \leq k \leq m \\ k \neq j}} Enc(Enc(GK'_j, GK_k), GK_j)$$

  else ▷ *Customer subscribed to multiple DR programs*
    MDMS $\rightarrow \omega_{jk}$:

$$\bigcup_{\substack{1 \leq h \leq d \\ 1 \leq k \leq m \\ k \neq j}} Enc(Enc(GK'_j, Child_h(GK_k)), GK_j)$$

    MDMS $\rightarrow \delta_{jk}$:

$$\bigcup_{\substack{1 \leq k \leq m \\ k \neq j}} Enc(Enc(GK'_j, GK_k), GK_j)$$

    Shift customer C_i to LKH tree corresponding to his second subscription DR_x using the standard LKH approach (without updating GK_x that customer C_i already had)
  end if
end procedure

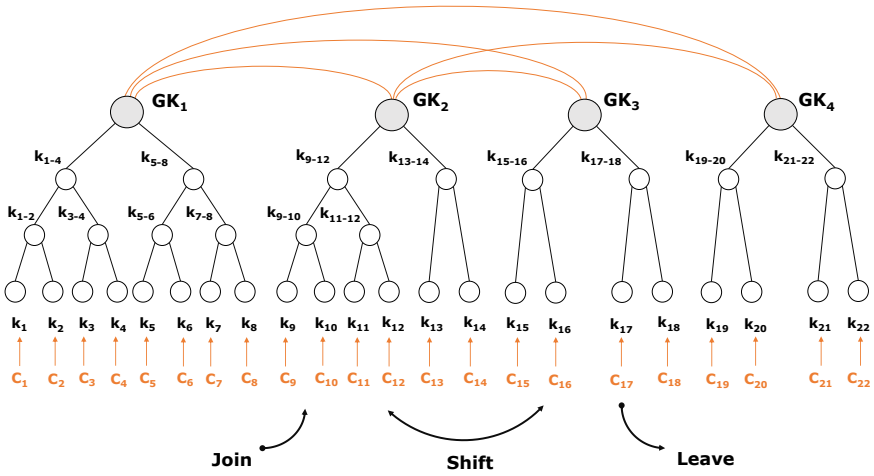


Figure 3: Proposed multi-group key graph.

We, next, illustrate with two examples how the customer leave procedure is executed by the MDMS when receiving a leave request from a Home DR program.

Example 1: We use Figure 4 to demonstrate how the MDMS handles the request of customer C_4 who subscribed only to DR_1 and requests to leave his Home DR program:

1. Standard LKH approach is used to update keys in the key tree corresponding to DR_1 (Figure 4b):

$$\text{MDMS} \rightarrow \{C_3\}: \ Enc(GK_1', k_3), \ Enc(k_{1-4}', k_3) \tag{1}$$

$$\text{MDMS} \rightarrow \{C_1, C_2\}: \ Enc(GK_1', k_{1-2}), \ Enc(k_{1-4}', k_{1-2}) \tag{2}$$

$$\text{MDMS} \rightarrow \{C_5, C_6, C_7, C_8\}: \ Enc(GK_1', k_{5-8}) \tag{3}$$

2. Update and replace the group key for customers in χ_{1k} (as shown in Figure 4c):

- First, we encrypt the new key GK_1' with keys GK_k to ensure that only customers belonging to DR_k tree can obtain the relevant key,
- Then, we encrypt this message with the old group key GK_1 to ensure that only customers subscribing to DR_1 can obtain the new group key (GK_1').

$$\text{MDMS} \rightarrow \chi_{12} : Enc(Enc(GK_1', GK_2)), GK_1) \tag{4}$$

$$\text{MDMS} \rightarrow \chi_{13} : Enc(Enc(GK_1', GK_3)), GK_1) \tag{5}$$

$$\text{MDMS} \rightarrow \chi_{14} : Enc(Enc(GK_1', GK_4)), GK_1) \tag{6}$$

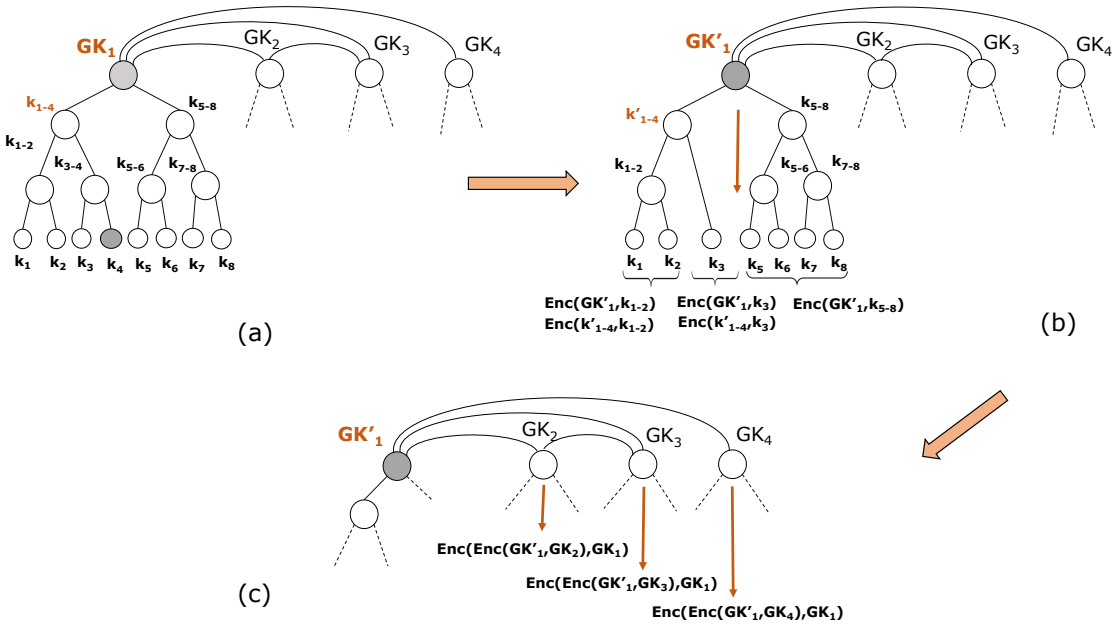


Figure 4: A leave rekeying example: when customer C_4 (subscribed only to one DR program) leaves DR_1.

Example 2: When customer C_3, who subscribed to DR_1, DR_2 and DR_3, requests to leave his Home DR program (DR_1), the MDMS renews keys as follows:

1. Standard LKH approach is used to update keys in the key tree corresponding to DR_1:

$$\text{MDMS} \rightarrow \{C_4\}: \ Enc(GK'_1, k_4), \ Enc(k'_{1-4}, k_4) \tag{7}$$

$$\text{MDMS} \rightarrow \{C_1, C_2\}: \ Enc(GK'_1, k_{1-2}), \ Enc(k'_{1-4}, k_{1-2}) \tag{8}$$

$$\text{MDMS} \rightarrow \{C_5, C_6, C_7, C_8\}: \ Enc(GK'_1, k_{5-8}) \tag{9}$$

2. Update and replace the group key for customers in ω_{1k}:

$$\text{MDMS} \rightarrow \omega_{12}: Enc(Enc(GK'_1, k_{9-12}), GK_1) \tag{10}$$

$$Enc(Enc(GK'_1, k_{13-14}), GK_1) \tag{11}$$

$$\text{MDMS} \rightarrow \omega_{13}: Enc(Enc(GK'_1, k_{15-16}), GK_1) \tag{12}$$

$$Enc(Enc(GK'_1, k_{17-18}), GK_1) \tag{13}$$

3. Update and replace the group key for customers in δ_{1k}:

$$\text{MDMS} \rightarrow \delta_{14}: Enc(Enc(GK'_1, GK_4), GK_1) \tag{14}$$

4. Shift C_3 to the LKH key tree corresponding to DR_2 which becomes his new home DR program using the standard LKH approach (without updating the group key GK_2 that customer C_3 already had).

- **Case 2:** When a customer C_i, who subscribed to multiple DR programs, leaves one DR program DR_j, which is not his Home DR program: The MDMS renews and updates keys according to Algorithm 2.

Let $DR_x = Home\_DR(C_i)$,

Let $\psi_{jk} = \{C_l / C_l \in \omega_{jk} \text{ and } DR_k \neq DR_x\}$,

Let $\pi_i = \{k_l / k_l = right(k_c) \text{ or } k_l = left(k_c),\ k_c \in Path(C_i)\}$.

Algorithm 2 Update keys, when a customer leaves a DR program ≠ his Home DR program

1: **procedure** LEAVEPROGRAM (C_i, DR_j)

2: Renew and update GK_j (GK'_j is the new group key)

3: MDMS $\rightarrow \chi_{jj}$: ▷ *Update key for customers belonging to the j^{th} DR program key tree*

$$\bigcup_{1 \leq h \leq d} Enc(GK'_j, Child_h(GK_j))$$

4: MDMS $\rightarrow \chi_{jx}$: ▷ *Update key for customers belonging to the same DR program key tree as C_i*

$$\bigcup_{k_\alpha \in \pi_i} Enc(Enc(GK'_j, k_\alpha), GK_j)$$

5: MDMS $\rightarrow \psi_{jk}$: ▷ *Update key for customers belonging to DR programs key trees to which C_i is subscribed*

$$\bigcup_{\substack{1 \leq h \leq d \\ 1 \leq k \leq m \\ k \neq j}} Enc(Enc(GK'_j, Child_h(GK_k)), GK_j)$$

6: MDMS $\rightarrow \delta_{jk}$: ▷ *Update key for customers belonging to DR programs key trees to which C_i is not subscribed*

$$\bigcup_{\substack{1 \leq k \leq m \\ k \neq j}} Enc(Enc(GK'_j, GK_k), GK_j)$$

7: **end procedure**

Example 3: We use Figure 5 to demonstrate how the MDMS handles the request of customer C_1 who subscribed to DR_1, DR_2 and DR_3, and requests to leave DR_2 which is not his Home DR program:

1. Update the group key for customers in χ_{22} (Figure 5a):

$$\text{MDMS} \rightarrow \{C_9, C_{10}, C_{11}, C_{12}\} : Enc(GK'_2, k_{9-12}) \tag{15}$$

$$\text{MDMS} \rightarrow \{C_{13}, C_{14}\} : Enc(GK'_2, k_{13-14}) \tag{16}$$

2. Update and replace the group key for customers in χ_{21} (Figure 5b) using a double encryption to ensure that only customers subscribing to DR_2 can obtain the new group key (suppose $\{C_2, C_3, C_4, C_6, C_7, C_8\}$ subscribed to DR_2):

$$\text{MDMS} \rightarrow \{C_2\} : Enc(Enc(GK'_2, k_2), GK_2) \tag{17}$$

$$\text{MDMS} \rightarrow \{C_3, C_4\} : Enc(Enc(GK'_2, k_{3-4}), GK_2) \tag{18}$$

$$\text{MDMS} \rightarrow \{C_6, C_7, C_8\} : Enc(Enc(GK'_2, k_{5-8}), GK_2) \tag{19}$$

3. Update the group key for customers in ψ_{jk} (Figure 5c):

$$\text{MDMS} \rightarrow \psi_{23} : Enc(Enc(GK'_2, k_{15-16}), GK_2) \tag{20}$$

$$\text{MDMS} \rightarrow \psi_{23} : Enc(Enc(GK'_2, k_{17-18}), GK_2) \tag{21}$$

4. Renew the group key for customers in δ_{jk} (Figure 5d):

$$\text{MDMS} \rightarrow \delta_{24} : Enc(Enc(GK'_2, GK_4), GK_2) \tag{22}$$

4.3.4. Join operation

The join operation deals with the case when a customer C_i subscribes to one new DR program DR_j. The MDMS applies the join rekeying Algorithm 3.

Algorithm 3 Update keys, when a customer joins a DR program

```
1:  procedure JoinProgram (C_i, DR_j)
2:      GK'_j = H(GK_j)
3:      if sub(C_i) ≥ 1 then                ▷ Customer subscribed to one/multiple DR programs
4:          MDMS → C_i : Enc(GK'_j, k_i)
5:          Send a notification to all customers in φ_j about the application of the one-way function
6:      else                                ▷ Customer's first subscription
7:          Send a notification to all customers in χ_jk about the application of the one-way function
8:          Apply the standard LKH approach in DR_j tree to insert customer C_i
9:      end if
10: end procedure
```

Example 4: When customer C_1, who subscribed to DR_1, DR_2 and DR_3, requests to join the DR program (DR_4), the MDMS updates and renews keys as follows:

1. Update the group key: $GK'_4 = H(GK_4)$

2. Send a notification to all customers in ϕ_4 about the application of a one-way function.

4.4. VerSAMI+: An Improved Version of VerSAMI

To deal with the presence of a very large number of smart meters with significant bandwidth constraints, we propose, thanks to the modularity of VerSAMI, an effective variant, called VerSAMI+, to overcome this limitation.

In VerSAMI+, we adopt OFT [38] (One-Way Function Trees) which is an improvement of LKH protocol (adopted in VerSAMI) that allows to reduce the number of rekeying messages.

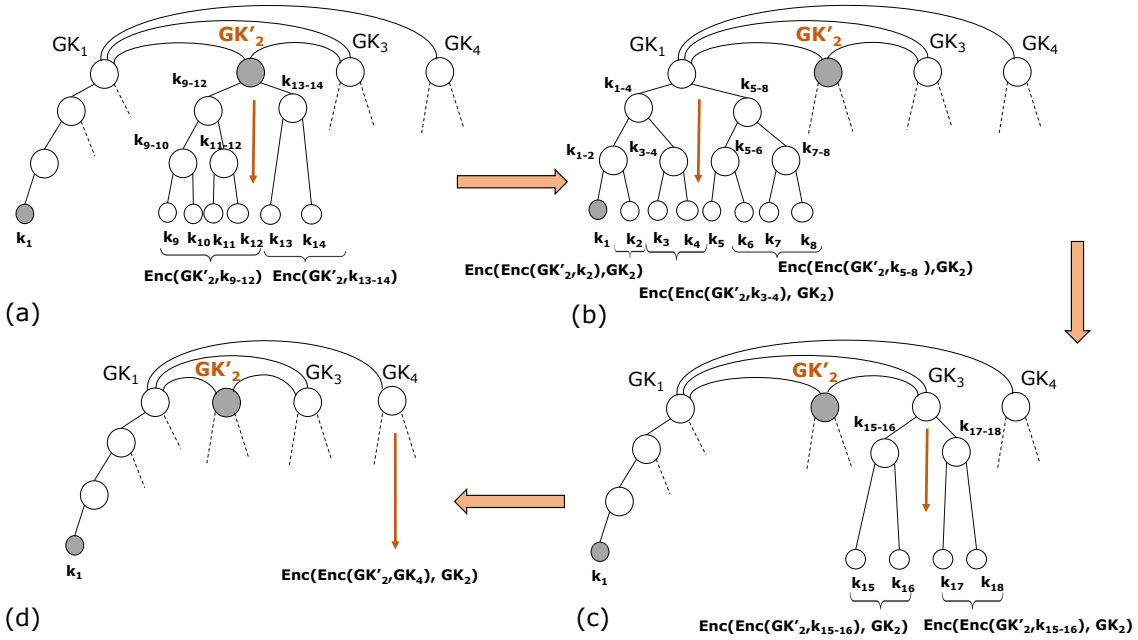


Figure 5: A leave rekeying example: when customer C_1 (subscribed to multiple DR programs) leaves DR_2.

In OFT, the MDMS and all customers individually compute the keys of interior nodes. These keys are recursively computed from the keys of their children using the formula:

$$k_i = f(k_{right(k_i)}) \oplus f(k_{left(k_i)}) \tag{23}$$

By applying the one-way function f to a key k: $f(k)$, is called the *blinded key* version of k. A customer only knows his individual key and the *blinded keys* of the *sibling nodes* of the nodes on his path to the root node, and these keys allow him to compute its *ancestors*. An example of OFT is illustrated in Figure 6.

The aforementioned join/leave operations, proposed for VerSAMI (Algorithm 1, 2, and 3), remain the same in VerSAMI+.

5. Batch Rekeying Version of VerSAMI and VerSAMI+

In VerSAMI (resp. VerSAMI+), a customer who subscribed to/unsubscribed from a DR program is accepted to/expelled from this program instantaneously. Thus, the MDMS updates keys, immediately, at every leaving and joining event, this is what we call *individual rekeying*. However, the frequent rekeying due to dynamic change of the DR programs membership can cause two problems:

- **Inefficiency problem:** When join/leave requests are frequent, many keys (auxiliary keys and group keys) may be updated and distributed to smart meters, but they are not used at all. This can incur high communication overhead.

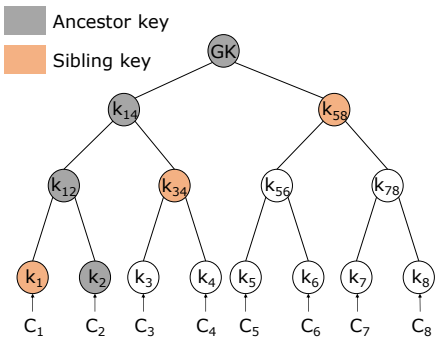


Figure 6: Example of OFT illustrating the ancestors and their corresponding sibling nodes of customer C_2.

- **Out-of-Sync problem:** It is hard to control the synchronization that will arise when rekey operations are executed after each join/leave request, because of the inter-dependencies between rekey messages as well as among rekey and other messages (metering data messages, remote load control messages, etc). For example, a SM might receive a remote load control message (or a rekey message) from the MDMS encrypted by a group key that it has not received yet. Figure 7. shows an example of this problem; at time t_1, the MDMS updates the i^{th} DR program group key GK_i for customers subscribed to this DR program; at time t_3, C_3 receives a remote load control message encrypted with GK_i' from the MDMS, but C_3 current group key for this DR program is GK. Thus, this problem may require a smart meter to buffer many messages encrypted with keys that it has not received, and/or keep many old keys.

To overcome this problem and reduce the number of rekeying operations, we propose an effective variant of VerSAMI (resp. VerSAMI+) called Batch-VerSAMI (resp. Batch-VarSAMI+). In Batch-VerSAMI (resp. Batch-VarSAMI+) membership changes are handled in batches instead of handling individually, the MDMS collects customer's join/leave requests to/from the different DR programs during a time period (known as *the batch rekeying interval*) and updates the group keys together in a batch. The main task of the MDMS is to identify which keys should be changed, deleted, or added. However, The MDMS cannot control which customer might leave a DR program, but it can control where to place the new customers in the multi-group key graph. Thus, the objectives of Batch-VerSAMI (resp. Batch-VerSAMI+) are to (1) reduce the number of rekeying messages; and (2) maintain the balance of the updated program's key trees.

At the end of each batch rekeying interval, and for each DR program DR_i; the MDMS collects: $J_h^i + L_h^i$ join/leave requests to/from the DR_i key tree (customers who subscribe/unsubscribe to/from their Home DR program), and $J_m^i + L_m^i$ join/leave requests to/from other DR programs (customers who subscribe/unsubscribe to/from a DR program which is not their Home DR program. Note that, if a customer who subscribed to multiple DR programs leaves his Home DR program DR_j and shift to DR_i, we consider

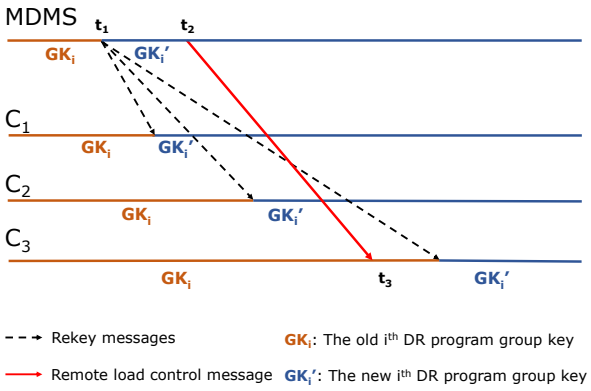


Figure 7: Example of Out-of-Sync problem

that he joins the DR_i key tree and process them (for each DR program) according to the below four possible cases to update the multi-group key graph and achieve the objectives mentioned above:

- **Case 1:** $(J_h^i = L_h^i = 0)$ **and** $(J_m^i \neq 0$ **or** $L_m^i \neq 0)$
  1. Update only the root of the i^{th} key tree (group key GK_i')
  2. Send GK_i' to customers belonging to the i^{th} key tree.
  3. Send GK_i' to customers subscribed to DR_i and belonging to other key trees.

- **Case 2:** $(J_h^i = L_h^i$ **and** $J_h^i \neq 0)$
  1. Update the root of the i^{th} key tree (group key GK_i')
  2. Update the interior nodes of the i^{th} tree by replacing leaves by joins.
  3. Mark the updated node in the key tree by $UPDATE$
  4. Send GK_i' to customers subscribed to DR_i and belonging to other key trees.

- **Case 3:** $(J_h^i < L_h^i)$
  1. Update the root of the i^{th} key tree (group key GK_i')
  2. Out of L_h^i leaves, replace J_h^i leaves with J_h^i joins.
  3. Mark the updated nodes in the key tree by $UPDATE$
  4. Mark the leaving nodes without joining replacement by $DELETE$
  5. Send GK_i' to customers subscribed to DR_i and belonging to other key trees.

- **Case 4:** $(J_h^i > L_h^i)$
  1. Update the root of the i^{th} key tree (group key GK_i')

2. Replace L_h^i leaves with $L_h(i)$ joins and mark updated nodes by $UPDATE$
3. In order to keep the tree balanced through different batches, insert individually the remaining $J_h^i - L_h^i$ new customers to the i^{th} key tree:
   (a) First, the MDMS replaces the nodes marked by $DELETE$
   (b) Then, if there are still extra J_h^i joins, insert customers individually without updating nodes marked by $UPDATE$
4. Send GK_i' to customers subscribed to DR_i and belonging to other key trees.

6. Security Analysis

In this section, we present a security analysis of our protocols and prove their security properties. As mentioned in section 3.4.1, the key management scheme should satisfy some properties. We discuss in the following, how VerSAMI, VerSAMI+, Batch-VerSAMI and Batch-VerSAMI+ satisfy these properties.

- **Property 6.1:** *VerSAMI (resp. VerSAMI+) and Batch-VerSAMI (resp. Batch-VerSAMI+) ensure strong backward secrecy.*

According to Algorithm 3, VerSAMI changes all the affected keys in the key tree when a new customer joins a DR program. This ensures that none of the old keys can be recovered by the new coming customer, which guarantees backward secrecy. Moreover, in Batch-VerSAMI (resp. Batch-VerSAMI+), the new joining/switching-in customers should wait for some time to be accepted (end of the batch rekeying interval). Thus, they do not know the keys before they join the DR program. In hence, Property 6.1 is satisfied by VerSAMI (resp. VerSAMI+) and Batch-VerSAMI (resp. Batch-VerSAMI+).

- **Property 6.2:** *VerSAMI and VerSAMI+ ensure strong forward secrecy.*

According to Algorithm 1, and 2, when a customer leaves a DR program, all keys known by the departing customer in both lower and upper level are changed and redistributed securely (because affected keys are encrypted when being broadcast), which prevents the departing customer from having access to the new keys without knowing the decryption keys. Hence, VerSAMI and VerSAMI+ satisfy Property 6.2. However, In Batch-VerSAMI (resp. Batch-VerSAMI+) an unsubscribed customer will remain in the DR program till the end of the batch rekeying period. Hence, Batch-VerSAMI and Batch-VerSAMI+ only ensure weak forward secrecy.

- **Property 6.3:** *VerSAMI and Batch-VerSAMI guarantee collusion freedom.*

According to [46], it was claimed that the OFT scheme was vulnerable to collusion freedom attacks. This means that VerSAMI+ and Batch-VerSAMI+ cannot preserve collusion freedom, whereas in VerSAMI and

Batch-VerSAMI, any set of customers that unsubscribe from a DR program cannot be able to deduce the current used group key, because all affected keys when any customer leaves a DR program will be replaced and new keys are independents (a customer does not know any key that is not in the path from his leaf to the group key corresponding to his last subscription). VerSAMI and Batch-VerSAMI thus satisfy Property 6.3.

- **Property 6.4:** *Batch-VerSAMI and Batch-VerSAMI+ alleviate the Out-of-Sync Problem.*

In [47], the authors demonstrated that, in contrast to individual rekeying, the periodic batch rekeying can alleviate the out-of-sync problem. Thus, Batch-VerSAMI and Batch-VerSAMI+ satisfy Property 6.4.

7. Performance Analysis

In this section, we analyze the performance of all the proposed protocols with respect to two metrics: storage overhead and communication overhead.

7.1. Storage Overhead

Here, we evaluate the storage overhead of VerSAMI at the MDMS side and the customer side (at smart meters), respectively. It is noted that, by fixing the degree of LKH key trees $d = 2$, storage overhead (at the MDMS and at smart meters) is the same for all the proposed variants: VerSAMI, VerSAMI+, Batch-VerSAMI, and Batch-VerSAMI+.

As used in most key management schemes, we assume that, the LKH and OFT key trees investigated in our schemes are fully loaded and maintained as balanced as possible.

7.1.1. MDMS side

Let $StorCost_{MDMS}$ denotes the storage overhead at the MDMS, and defined as the number of keys stored at the MDMS when the smart grid utility provides m DR programs to their n customers. The MDMS stores: (1) the broadcast key, (2) all individual keys of the n customers, and (3) the total number of keys on the key trees corresponding to DR programs (except the already mentioned individual keys). The storage overhead at the MDMS for VerSAMI is calculated as:

$$StorCost_{MDMS} = \Big(\sum_{i=1}^{m} \frac{dn_i - 1}{d - 1} + 1\Big)|K| \tag{24}$$

7.1.2. Customer side

Let $StorCost_{SM_i}$ denotes the storage overhead at the SM_i (customer C_i side). Each smart meter stores: (1) the broadcast key, (2) his individual secret key, (3) all keys corresponding to the nodes in the path

from his leaf to the root in his Home DR program (DR_j) key tree, and (4) all group keys of the other DR programs to which he is subscribed. Thus, the storage overhead at the smart meter is calculated as:

$$StorCost_{SM_i} = \big(h_j + sub(C_i)\big)|K| \tag{25}$$

Where, h_j is the height of j^{th} key tree, h_j is either $log_d(n_j)$ or $log_d(n_j)+1$ when the key tree has degree d.

7.2. Communication Overhead

VerSAMI, VerSAMI+, Btach-VerSAMI, and Batch-VerSAMI+ use an efficient multi-group key graph structure. The leaving/joining/shifting scenarios introduce extra rekey overhead, and even though the number of customer members and the provided DR programs are the same, the number of keys to be updated varies according to the positions of the leaving/joining/shifting customer in the multi-group key graph. In this section, we calculate the rekeying overhead for both individual rekeying versions (i.e. VerSAMI and VerSAMI+) and batch rekeying versions (i.e. Batch-VerSAMI and Batch-VerSAMI+).

7.2.1. Individual rekeying

Let $CommCost_1$ (resp. $CommCost_2$) denotes the communication overhead of VerSAMI protocol (resp. VerSAMI+). We calculate in the following the amount of rekeying messages when a customer joins/leaves a DR program:

Join operations. According to Algorithm 1 and Algorithm 2 presented in section 4.3.3, the communication overhead in the worst cases will be as follows:

- **Case 1:** When a customer C_i, who subscribed only to one DR program, leaves his Home DR program DR_j. Thus, the communication overhead of VerSAMI is:

$$commCost_1 = (dh_j + m - 1)|K| \tag{26}$$

  In VerSAMI+, as the adopted key tree (OFT) represents an improvement of LKH (adopted in VerSAMI), it allows to reduce the number of rekey messages. Thus, the communication overhead of VerSAMI+ is:

$$commCost_2 = (h_j + m - 1)|K| \tag{27}$$

- **Case 2:** When a customer C_i, who subscribed to multiple DR programs at the same time, leaves his Home DR program DR_j. Thus, the rekeying overheads of VerSAMI and VerSAMI+ are:

$$commCost_1 = (dh_j + dX + Y + dh_l)|K| \tag{28}$$

$$commCost_2 = (h_j + 2X + Y + h_l)|K| \tag{29}$$

$X = sub(C_i)$.

$Y = m - sub(C_i)$.

h_l : height of the new Home DR program's key tree.

- **Case 3:** When a customer C_i, who subscribed to multiple DR programs at the same time, leaves a DR program DR_j which is not his Home DR program DR_l. Thus, the rekeying overheads are:

$$commCost_1 = (d + dh_l + dX + Y)|K| \tag{30}$$

$$commCost_2 = (2 + 2h_l + 2X + Y)|K| \tag{31}$$

Join operations. According to Algorithm 3 presented in section 4.3.4, the communication overhead will be as follows:

- **Case 1:** When a customer C_i joins his first DR program DR_j (Home DR program). Thus, the communication overheads are:

$$commCost_1 = dh_j|K| + c \tag{32}$$

$$commCost_2 = h_j|K| + c \tag{33}$$

The " $+ c$" term is to specify the group key on which we must apply the one-way function $c = log_2 m$.

- **Case 2:** When a customer C_i joins a new DR program DR_j which is not his Home DR program DR_l. Thus, the communication overheads are:

$$commCost_1 = |K| + c \tag{34}$$

$$commCost_2 = |K| + c \tag{35}$$

7.2.2. Batch rekeying

As mentioned in Section 5, the MDMS cannot control which customers might leave a DR program, but it can control where to place the new customers in the multi-group key graph. Since Batch-VerSAMI (resp. Batch-VerSAMI+) takes different operations for the four possible cases to update the multi-group key graph, the communication overheads in the worst cases (for each DR program DR_i) are also devided into four cases. Let $CommCost_{3t}(i)$ (resp. $CommCost_{4t}(i)$) denotes the communication overhead of Batch-VerSAMI protocol (resp. Batch-VerSAMI+) for the DR program DR_i which corresponds to Case t.

- **Case 1:** $(J_h^i = L_h^i = 0)$ **and** $(J_m^i \neq 0$ **or** $L_m^i \neq 0)$

The worst case occurs when the customers subscribed to DR_i and having DR_j as Home DR program are evenly distributed across the leaf nodes in the j^{th} key tree (i.e. customers do not have shared

interior keys in the j^{th} key tree), Thus, the rekeying overheads are:

$$commCost_{31}(i) = \big(d + \sum_{\substack{j=1 \\ j \neq i}}^{m} member(i,j)\big)|K| \tag{36}$$

$$commCost_{41}(i) = \big(2 + \sum_{\substack{j=1 \\ j \neq i}}^{m} member(i,j)\big)|K| \tag{37}$$

- **Case 2:** $(J_h^i = L_h^i$ **and** $J_h^i \neq 0)$

  In this case, for simplicity we assume that $L_h^i = d^k$ for some integer k. The worst case happens when the L_h^i leaves are distributed across the leaf nodes in the i^{th} key tree, and similary to the previous case the customers subscribed to DR_i and having DR_j as Home DR program are evenly distributed across the leaf nodes in the j^{th} key tree. Thus, the communication overheads are:

$$commCost_{32}(i) = \big(L_h^i dlog_d(\frac{N}{L_h^i}) + \frac{d(L_h^i - 1)}{d-1} + \sum_{\substack{j=1 \\ j \neq i}}^{m} member(i,j) - (L_m^i - J_m^i)\big)|K| \tag{38}$$

$$commCost_{42}(i) = \big(L_h^i log_2(\frac{N}{L_h^i}) + 2(L_h^i - 1) + \sum_{\substack{j=1 \\ j \neq i}}^{m} member(i,j) - (L_m^i - J_m^i)\big)|K| \tag{39}$$

- **Case 3:** $(J_h^i < L_h^i)$

  This case is similar to the previous one, the only difference is that there are only $n_i - (L_h^i - J_h^i)$ customers left in the i^{th} key tree. Thus:

$$commCost_{33}(i) = commCost_{32}(i) - \big((L_h^i - J_h^i)|K|\big) \tag{40}$$

$$commCost_{43}(i) = commCost_{42}(i) - \big((L_h^i - J_h^i)|K|\big) \tag{41}$$

- **Case 4:** $(J_h^i > L_h^i)$

  This case is devided to two cases:

  - **Case 4.1:** $(J_h^i > L_h^i$ **and** $L_h^i = 0)$**:** In this case, there are no leaves from the i^{th} key tree. Thus, the communication overheads are:

$$commCost_{34}(i) = \big(J_h^i dlog_d(N) + \sum_{\substack{j=1 \\ j \neq i}}^{m} member(i,j) - (L_m^i - J_m^i)\big)|K| \tag{42}$$

$$commCost_{44}(i) = \big(J_h^i log_2(N) + \sum_{\substack{j=1 \\ j \neq i}}^{m} member(i,j) - (L_m^i - J_m^i)\big)|K| \tag{43}$$

  - **Case 4.2:** $(J_h^i > L_h^i$ **and** $L_h^i > 0)$**:** In this case, there are more joins than leaves to the i^{th} key tree. Thus, the communication overheads are a combination of case 2 and case 4.1:

$$commCost_{34}(i) = commCost_{32}(i) + (J_h^i - L_h^i)dlog_d(N)|K| \tag{44}$$

$$commCost_{44}(i) = commCost_{42}(i) + (J_h^i - L_h^i)log_2(N)|K| \tag{45}$$

8. Simulations and Performance Comparison

In this section, we use simulation to examine the impact of customer behaviors and prove the efficiency of our proposed schemes. We compare our four schemes with the two most recent proposed schemes, i.e. Liu's *et al.* scheme [22] and SKM+ [23], to illustrate how *the multi-group key structure* improve the performance of key management.

8.1. Network and Dynamic Membership Models

Before performing simulations, the AMI network architecture and the dynamic behavior of customers must be modeled. For this, we consider a smart grid with the following parameters:

8.1.1. Network Model

- As shown in Fig 8, network model for AMI is divided into a number of hierarchical networks comprising the MDMS, concentrators, and SMs.
- Communications between SMs and concentrators are established through Wi-Fi technology. However, since the distances between concentrators and the MDMS are far away, the communications between concentrators and the MDMS are established through wired links (the SDH network based on optical fibers is always used as the main communication channel between concentrators and the MDMS; the transmission rate is 155 Mb/s, 622 Mb/s, and more).
- The simulation scenario corresponds to a city (100 x 100 Km area) with about 2 million population, with the assumption that each family (average size is 4 persons/family) owns an IEEE 802.11 wireless-based SM.
- The MDMS is placed at the center of the service area, and 5000 concentrators are uniformly distributed in the area. Thus, the number of SMs covered by each concentrator is 100 on average.
- In our simulations, for communications through optical fibers, the transmission rate is set to 155 Mb/s. Whereas, for communications through Wi-Fi, a fixed antenna model is used with the two-ray propagation channel model, the transmission power is set to be fixed at 0.03652 W, providing a range of 150 m, which is consistent with practical values for Wi-Fi, and the channel data rate is set to 11 Mb/s.

8.1.2. Dynamic Membership Model

- The utility provides m DR programs to customers (e.g., Real Time Pricing program, Time Of Use Pricing program, Direct Load Control program, Emergency Demand Reduction program, etc).

- Customers arrival to the Home DR program is modeled as a Poisson process with parameter λ_1 (subscribers/month), i.e., the number of customers subscribing to the first Home DR program per month.
- Customers arrival to other DR programs is modeled as a Poisson process with parameter λ_2 (subscribers/month), i.e., the number of customers subscribed to one/multiple DR programs and subscribing to a new DR program per month.
- Given that, to the best of our knowledge, there are no statistical studies of DR programs membership behavior for the moment, we assume that membership duration in the Home DR program follows an Exponential law with parameter μ_1, and the membership duration in the other DR programs follows an Exponential law with parameter μ_2 (with $\mu_1 > \mu_2$).
- A typical customer session starts by a join event to a DR program, which can be followed by one or more join/leave events to/from other DR programs. The customer leaves this DR program at the end of his membership.
- We assume that $p1$ (percent) of the customers only subscribe to one of the DR programs (*single subscription*), and $p2$ (percent) subscribe to multiple DR programs (*multiple subscription*).
- A 256b long symmetric keys are used.
- Balanced binary LKH/OFT trees ($d = 2$) are used.

We consider a session of *36 months*. We vary separately the inter-arrival average λ_1 and λ_2. The average membership duration in a DR program is *6 months* for the first subscription (Home DR program), and *3 months* for other subscriptions. Moreover, we assume that the batch rekeying runs *twice a week*. Storage and Communication overhead of Liu's *et al.* scheme and SKM+ scheme are readily obtained from [22] and [23].

8.2. Simulation Results

8.2.1. Storage Overhead

MDMS Side. We evaluate the storage overhead of the key management schemes at the MDMS side by fixing the inter-arrival rates $\lambda_1 = \lambda 2 = 5000$ *subscribers/month* and varying separately the number of DR programs provided by the utility, and the number of customers subscribed to each DR program. We assume that $p1 = p2 = 50\%$.

Figure 8 shows a comparison of storage overhead at the MDMS between the four schemes (note that, by the use of balanced binary LKH/OFT key trees, the storage overhead is the same for all the proposed schemes, i.e., VerSAMI, Batch-VerSAMI, VerSAMI+, and Batch-VerSAMI+). In Figure 8a, we consider

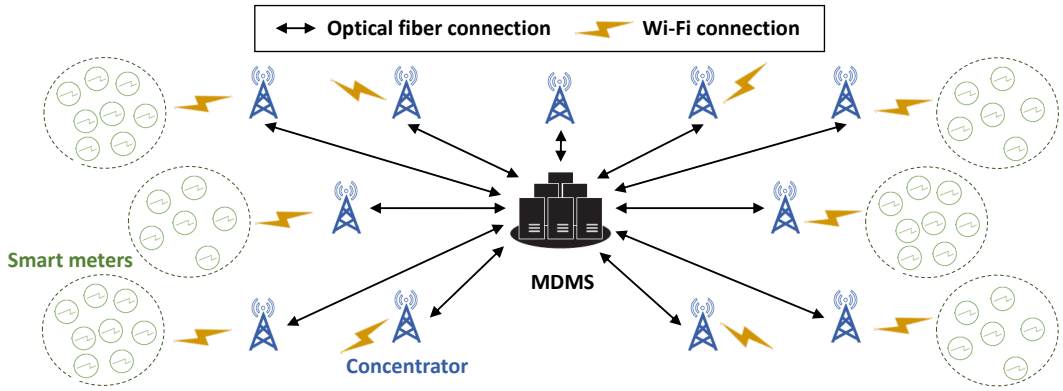


Figure 8: Network model for AMI

DR programs of average 20000 subscribers, and 200000 subscribers and we vary the number of provided DR programs. Figure 8b shows a comparison of storage overhead at the MDMS with respect to DR programs size while fixing the number of provided DR programs to 5 DR programs and 15 DR programs.

Liu's *et al.* scheme does not adopt a key graph technique, as a consequence, the MDMS stores fewer keys than that in SKM+ and VerSAMI, in which the number of keys stored at the MDMS increases proportionally with the number of subscribed DR programs (resp. with the DR programs size), whereas, in VerSAMI we notice that the number of provided DR programs (resp. the DR programs size) does not significantly affect the storage overhead at the MDMS due to the proposed efficient multi-group key graph structure.

It is noted that for MDMS, the storage overhead is not a problem; we can use special key servers as storage. However, the storage overhead in the smart meters cannot be neglected due to the fact that smart meters storage ability is limited [12–15].

Customer Side. Two test cases are generated for evaluation based on major subscriptions to the provided DR programs. In Case 1, 70 percent of the customers subscribe to multiple DR programs simultaneously, and the other 30 percent only subscribe to one of the DR programs (i.e. $p1 = 30\%$ and $p2 = 70\%$). In Case 2, 30 percent of the customers subscribe to multiple DR programs, and the other 70 percent only subscribe to one of the DR programs (i.e. $p1 = 30\%$ and $p2 = 70\%$). We evaluate the storage overhead at smart meters by fixing the inter-arrival rates $\lambda_1 = \lambda 2 = 5000$ $subscribers/month$ and varying separately the number of DR programs provided by the utility, and the DR programs size. As for the MDMS, the storage overhead at smart meters is the same for all the proposed schemes, i.e., VerSAMI, Batch-VerSAMI, VerSAMI+, and Batch-VerSAMI+.

Figure 10 shows a comparison of average storage overhead at smart meters with respect to the number of provided DR programs. We considered DR programs of average 80000 and 200000 subscribers. In both Case 1 and Case 2, we can notice that in VerSAMI, a smart meter stores little more keys than that in Liu's *et al.*

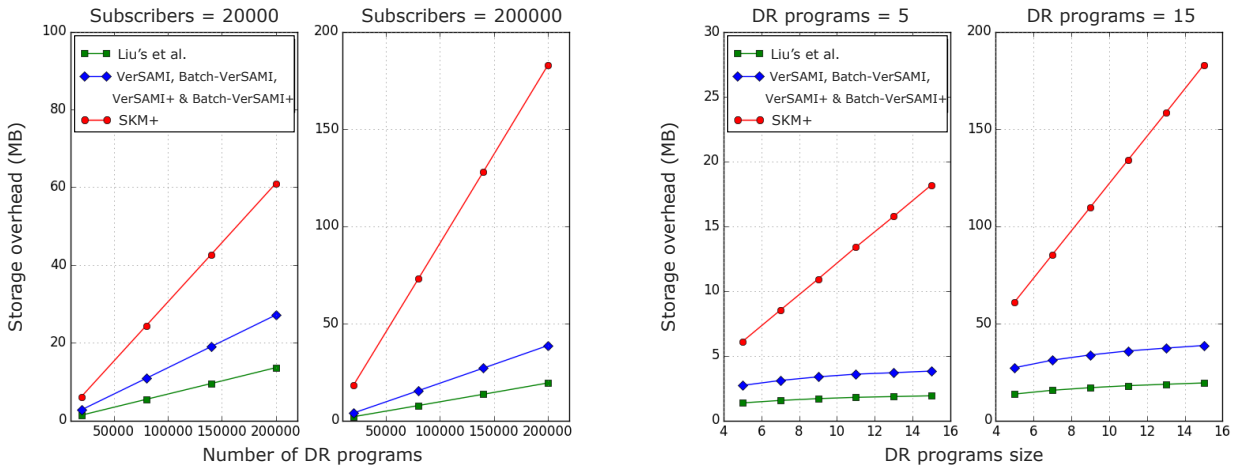
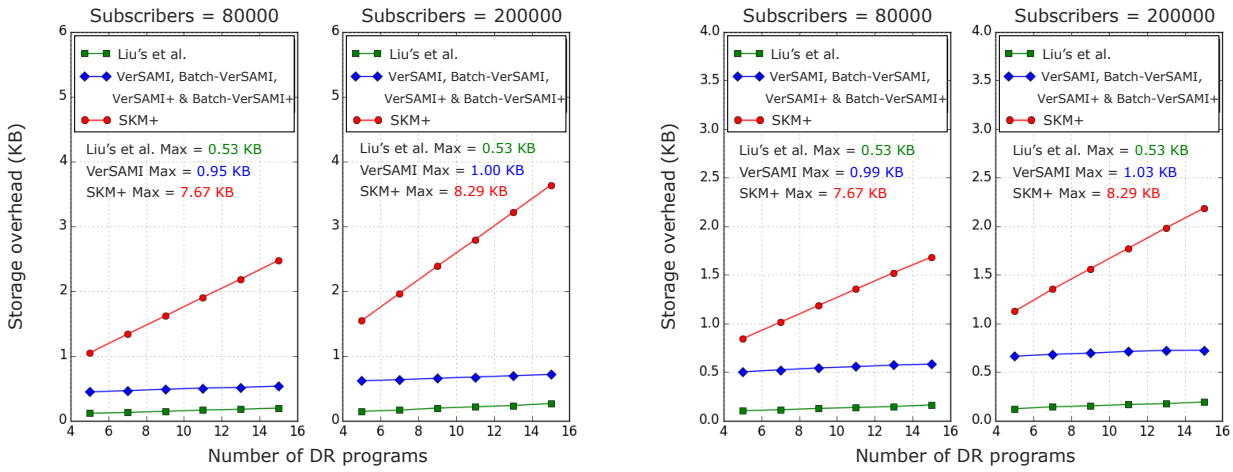


Figure 9: Storage overhead at the MDMS with respect to (a) number of DR programs (b) DR programs size.

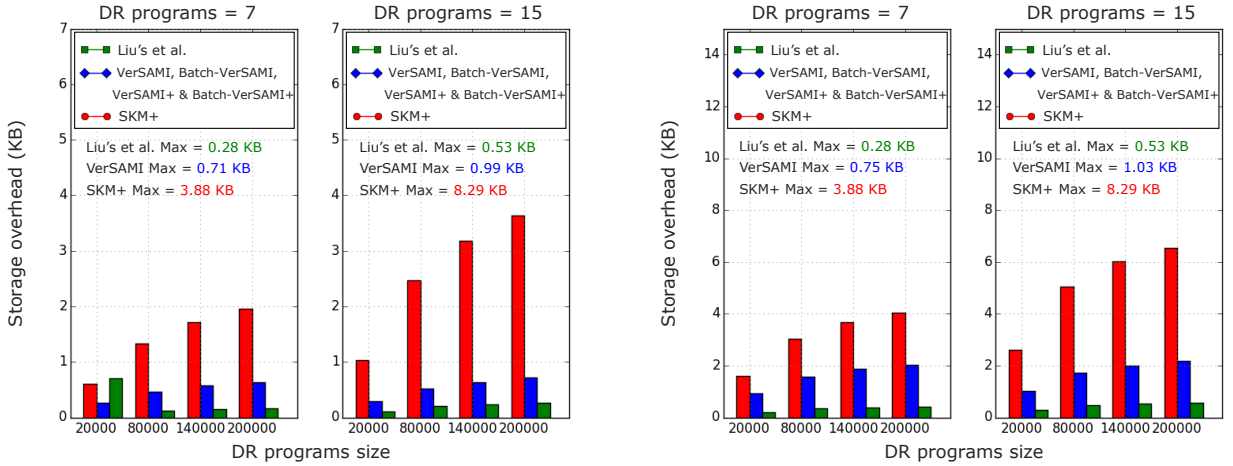
scheme but much fewer keys than that in SKM+. In Case 1, reduction reaches more than 80% in average storage cost and more than 87% in maximum storage cost when the number of subscribers is about 200000 and a customer can subscribe to 15 DR programs at the same time. Whereas, in Case 2, reduction reaches more than 66% in average storage cost and more than 87% in maximum storage cost. This can be explained as follows: in Liu's *et al.* scheme, a smart meter stores one key for each subscribed DR program, because Liu *et al.* does not adopt a key graph technique. In SKM+, authors used an OFT for each DR program, the number of keys stored will increase significantly when a customer subscribes to new DR programs. Whereas, in VerSAMI we can see that the number of subscribed DR programs does not significantly affect the storage overhead.

Figure 11 shows the average storage overhead at smart meters with respect to DR programs size. We consider, respectively, 7 and 15 DR programs. In Liu's *et al.* scheme, smart meters store only the group keys, as a consequence, the DR program size does not affect significantly the storage overhead. Whereas, in SKM+ and VerSAMI the DR programs size affects the storage overhead, as the number of customers increases, the storage overhead increases due to the rise of the height of the used LKH/OFT key trees, but smart meters store much fewer keys in VerSAMI with respect to SKM+. Figure 11 also show that VerSAMI, smart meters store little more keys in Case 2 (in which the major subscriptions are *single subscriptions*) than in Case 1, due to the increase of the customers belonging to their Home DR program key tree. But, this variation does not affect significantly the storage overhead at smart meters. Note that, the maximum storage overhead in a smart meter reaches 0.53 KB for Liu's *et al.* scheme, 1.03 KB for both VerSAMI and VerSAMI+, and 8.29 KB for SKM+ which is a high overhead taking into account that smart meters own limited storage resources, such as 8 KB RAM and 120 KB Flash memory set in [14].



(a) Case 1 (b) Case 2

Figure 10: Average storage overhead at SMs with respect to number of DR programs. (a) Case 1. (b) Case 2.



(a) (b)

Figure 11: Average storage overhead at SMs with respect to DR programs size. (a) Case 1. (b) Case 2.

8.2.2. Communication Overhead

Performance with different number of DR programs. We first evaluate and compare the communication overheads of the six key management schemes by considering DR programs of average 200000 subscribers, and fixing inter-arrival rates $\lambda_1 = \lambda 2 = 5000$ *subscribers/month* and varying the number of DR programs provided by the utility. Two test cases are generated for evaluation based on major subscriptions to DR programs: in Case 1, major subscriptions are *multiple* (i.e. $p1 = 30\%$ and $p2 = 70\%$), and in Case 2, major subscriptions are *single* (i.e. $p1 = 70\%$ and $p2 = 30\%$).

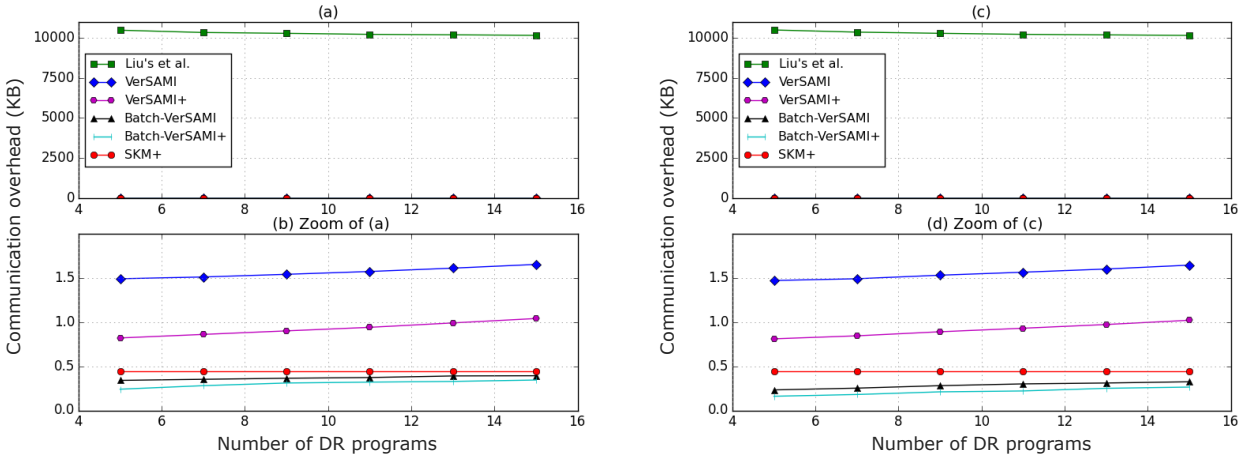


Figure 12: Average communication overhead per event with respect to number of DR programs. (a,b) Case 1. (c,d) Case 2.

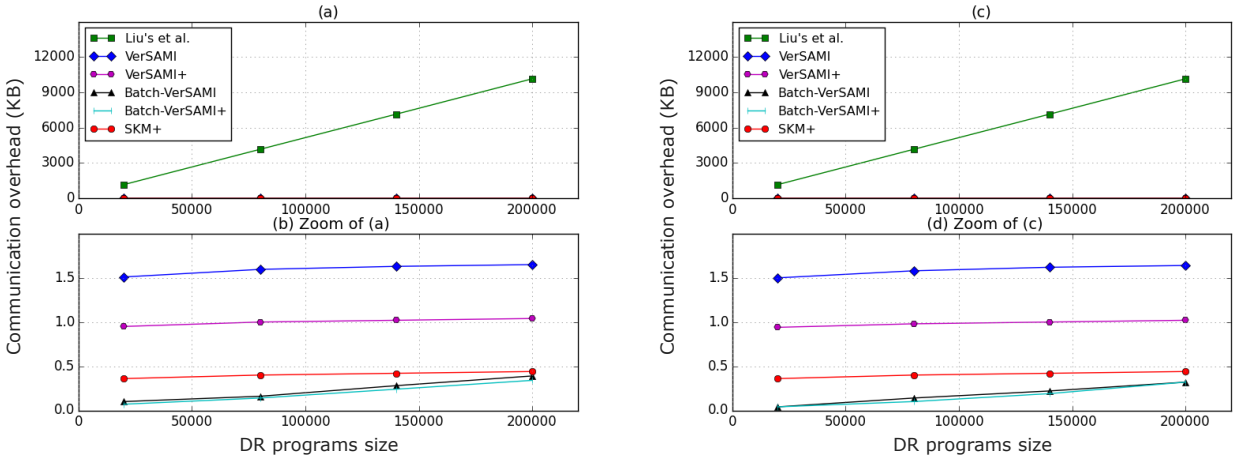


Figure 13: Average communication overhead per event with respect to DR programs size. (a,b) Case 1. (c,d) Case 2.

Figure 12 shows a comparison of average communication overhead per event with respect to the number of DR programs. In both Case 1 and Case 2, the communication overhead due for the Liu's *et al.* scheme is remarkably higher than that of the five other schemes because of the inefficient multicast key management. In all of our proposed schemes, communication overhead reduction reaches more than 99% with respect to Liu's *et al.* scheme. It is noted that in SKM+, authors use a separate OFT key tree for each DR program, which induces less communication overhead than VerSAMI and VerSAMI+ that adopt a multi-group key graph. However, we can see that Batch-VerSAMI and Batch-VerSAMI+ provide the smallest communication overheads due to the efficient periodic rekeying, which reduces the number of rekeying messages.

Performance with different DR programs size. We use the same previous settings, and we fixe the number of DR programs to 15 and vary the DR programs size.

Figure 13 shows a comparison of average communication overhead per event for the six schemes with respect to the DR programs size while fixing the number of DR programs to 15 DR programs. In Liu's *et al.* scheme, when a join/leave occurs the MDMS sends, individually, the new group key for all customers subscribed to this DR program. Thus, the communication overhead increases linearly with the number of DR program subscribers. Whereas, the overhead remains much lower in SKM+, VerSAMI, and VerSAMI+ as shown in Figure 12b and 12d (the communication overhead of SKM+, VerSAMI and VerSAMI+ is too little to be seen in Figure 12a and Figure 12c). Certainly, by using a multi-group key graph, VerSAMI and VerSAMI+ introduces extra communication cost compared to SKM+, but this overhead is minor regarding the overall advantages of the proposed schemes, mainly when considering the storage overhead as shown above. Moreover, we can see that batch rekeying (both Batch-VerSAMI and Batch-VerSAMI+) has a substantial benefit over all the other schemes based on individual rekeying.

Performance with different subscriptions and inter-arrival rates. By fixing the number of provided DR programs to 15, and the average number of subsribers in each DR programs to 200000, four test cases are generated for evaluation based on major subscriptions and major events (as summarized in Table 3). In Case 1 and Case 3, $p1 = 70\%$ and $p2 = 30\%$. Whereas, in Case 2 and Case 4, $p1 = 30\%$ and $p2 = 70\%$. Furthermore, we vary the rates for major events while keeping the other rates at 1000 $customers/month$.

Figure 14 shows that the communication overhead of Liu's *et al.* scheme is remarkably higher than the other schemes due, as mentioned above, to the innefficient multicast key management. In SKM+, authors do not adopt a multi-group key graph; i.e. when a customer joins (leaves) a Home DR program or another DR program, he only needs to join (leave) one key tree. Thus, the overhead remains almost flat, regardless of the inter-arrival rates of customers events. In VerSAMI and VerSAMI+, the communication cost grows when inter-arrival rates of customers events increase, because of extra overhead that is induced in managing keys for customers subscribing to multiple DR programs, and shift between key trees (in Case 1 and Case 2). We can see, also, that batch rekeying has a substantial benefit over all the other schemes based on individual rekeying. Moreover, the overhead of Batch-VerSAMI and Batch-VerSAMI+ is little more in Case 1 and Case 2, than in Cases 3 and Case 4, due to the extra overhead of joining and leaving Home DR programs key trees.

8.2.3. Time of Key Distribution

We define the time of key distribution as the required delay to distribute the needed keys to their recipients per event.

We evaluate the average time the key distribution when a join/leave event occurs for the six key management schemes (Liu's *et al.* scheme, SKM+, VerSAMI, VerSAMI+, Batch-VerSAMI, and Batch-VerSAMI+)

Table 3: Cases for evaluation based on major subscriptions and major events

Case	Major subscriptions	Major events
Case 1	Single	Join / Leave Home DR programs
Case 2	Multiple	Join / Leave Home DR programs
Case 3	Single	Join / Leave other DR programs
Case 4	Multiple	Join / Leave other DR programs

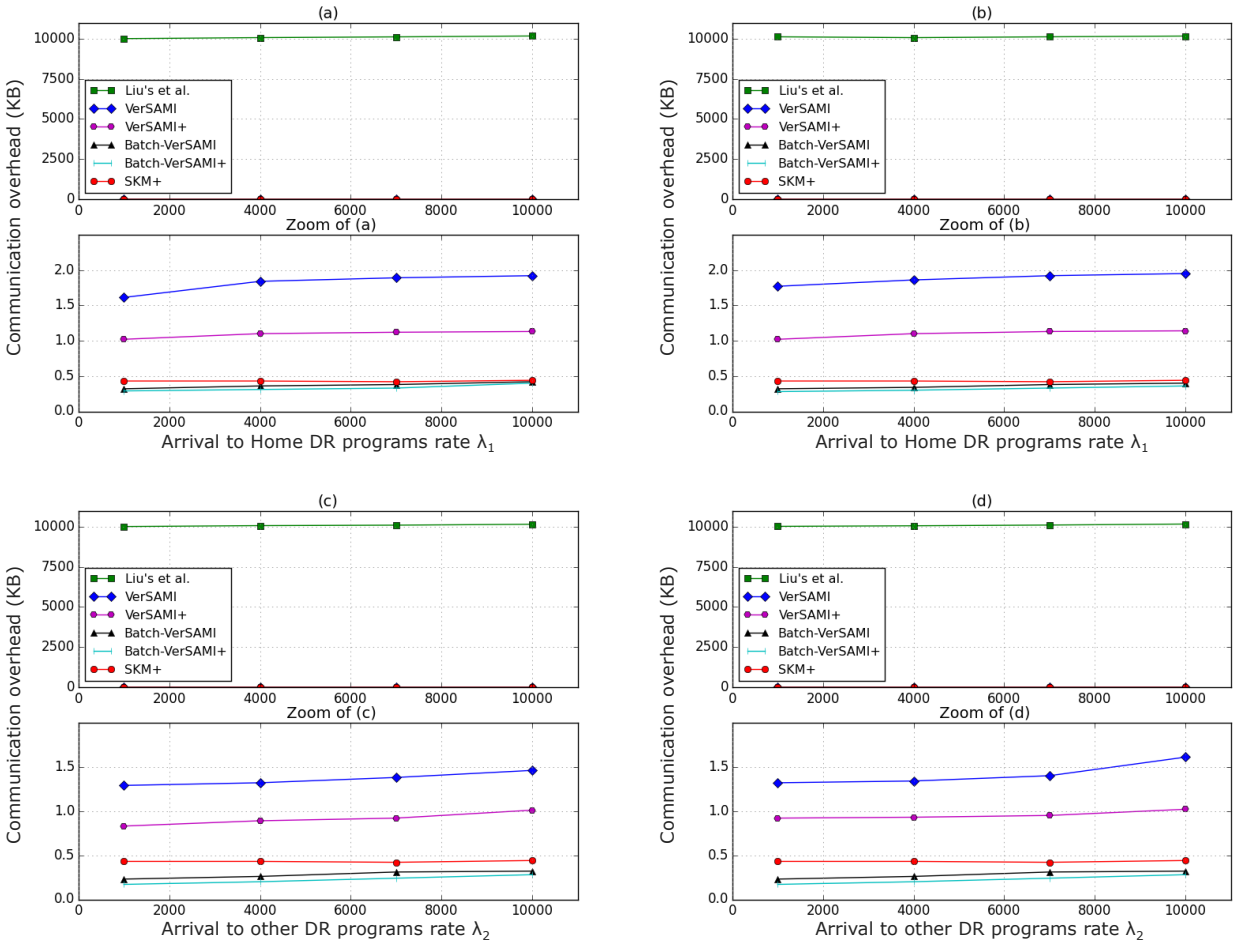


Figure 14: Average communication overhead per event. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4.

by fixing $p1 = p2 = 50\%$ and $\lambda_1 = \lambda 2 = 5000$ *subscribers/month*. We vary, respectively, the size of DR programs and the number of DR programs provided by the utility as shown in Figure 15 and Figure 16.

In Liu's *et al.* scheme, when a join/leave event occurs the group key is sent, individually, to all customers subscribed to this DR program. Thus, it can be observed in Figure 15 that the time of key distribution

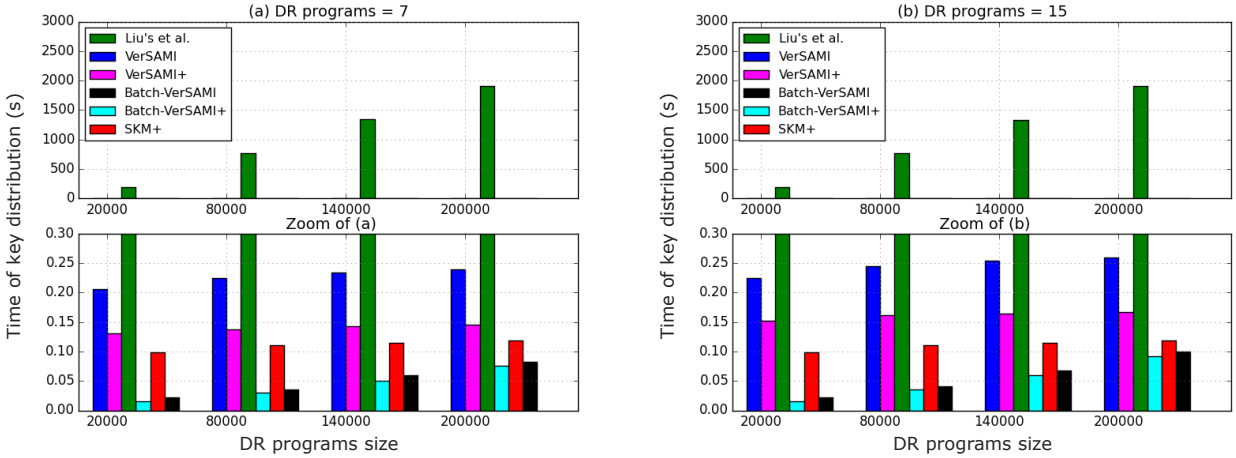


Figure 15: Average time cost of key distribution per event with respect to DR programs size.

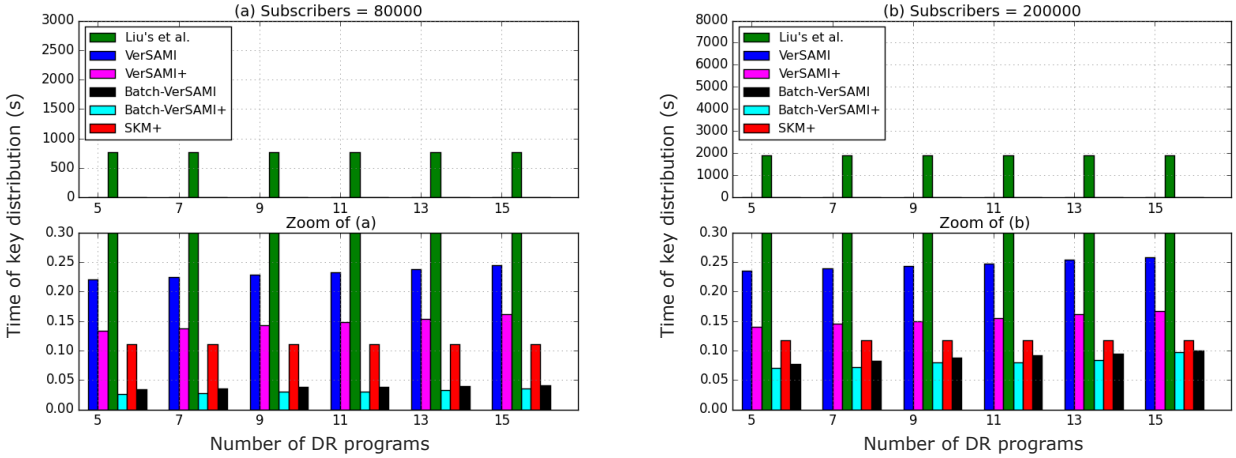


Figure 16: Average time cost of key distribution per event with respect to number of DR programs.

increases linearly with the number of DR program subscribers. Moreover, the time of key distribution is remarkably higher than the other schemes and reaches about 33 *minutes*; this can affect the refreshing of keys and the distribution of the network traffic in AMI systems. The time of key distribution remains much lower in VerSAMI, VerSAMI+, and SKM+, as shown in Figure 15b and Figure 16b. It varies between $206 - 258$ ms for VerSAMI, $131 - 167$ ms for VerSAMI+ and $98 - 118$ ms for SKM+. Thus, the time of key distribution will not affect the refreshing of keys and the distribution of the network traffic in AMI systems. Moreover, it can be observed that batch rekeying has a substantial benefit in term of time cost of keys distribution per join/leave event over all the other schemes based on individual rekeying.

Table 4: Comparison of Key Management Schemes

		Liu's *et al.* [22]	**SKM+ [23]**	**VerSAMI**	**VerSAMI+**	**Batch-VerSAMI**	**Batch-VerSAMI+**
Key graph approach		Yes	Yes	Yes	Yes	Yes	Yes
Versatility		Yes	Yes	Yes	Yes	Yes	Yes
Backward secrecy		Strong	Strong	Strong	Strong	Strong	Strong
Forward secrecy		Strong	Strong	Strong	Strong	Weak	Weak
Collusion freedom		Yes	No	Yes	No	Yes	No
Prevent Out-of-Sync Problem		No	No	No	No	Yes	Yes
Overhead	Storage	Low	High	Low	low	Low	Low
	Communication	Very high	Very low	Low	Low	Very low	Very low

8.3. Discussion

In this section, we compare the six schemes using both security and performance metrics. As shown in Table 4, all schemes ensure: versatility. Moreover, the backward secrecy property is ensured by all the schemes. On the other hand, in batch rekeying based schemes (i.e. Batch-VerSAMI and Batch-VerSAMI+) an unsubscribed customer will remain in the DR program till the end of the batch rekeying period. Hence, Batch-VerSAMI and Batch-VerSAMI+ ensure weak forward secrecy compared to the other individual rekeying based schemes. Furthermore, batch rekeying based schemes alleviate the Out-of-Sync problem.

We, also, remark in Table 4, that the key management schemes based on OFT key trees (SKM+, VerSAMI+, and Batch-VerSAMI+) are vulnerable to collusion freedom attacks as claimed in [46].

Regarding performance comparison given in Table 4, Liu's *et al.* scheme do not adopt an efficient key graph technique. Thus, their scheme induces low storage overhead and very high communication overhead for such a large-scale system, due to inefficient multicast key management. Certainly, SKM+ incurs a very low communication cost, but in this scheme a smart meter has to store a large amount of keys when customer subscribe to multiple DR programs (storage overhead reaches more than 8 KB as shown in simulations), which is a high overhead taking into account the fact that the smart meters own limited storage resources.

By using a multi-group key graph, VerSAMI and VerSAMI+ induce low storage overhead and can support more customers; the advantage of our multi-group key graph is larger when more customers subscribe to multiple DR programs simultaneously. Certainly, VerSAMI and VerSAMI+ produce little more communication overhead compared to SKM+, but this overhead is minor regarding the overall advantages of the proposed schemes, mainly when considering the storage overhead.

We can notice that both Batch-VerSAMI and Batch-VerSAMI+ have a substantial benefit, in term of communication overhead, over all the other schemes based on individual rekeying.

We can also notice that individual rekeying schemes ensure strong backward/forward secrecy but can suffer from an Out-of-Sync problem. On the other hand, batch rekeying schemes alleviate this problem and

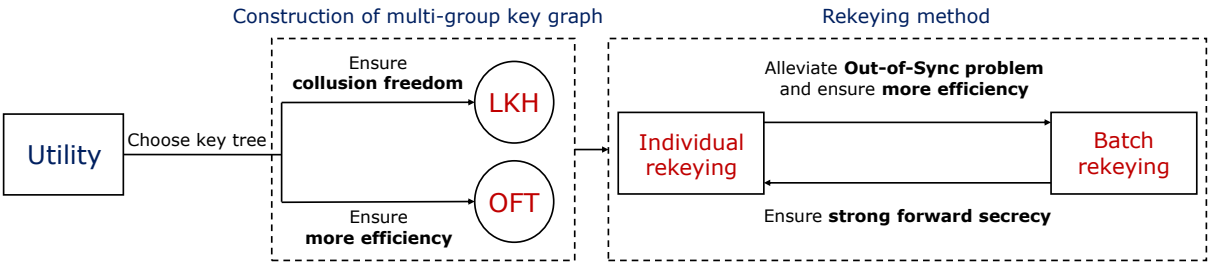


Figure 17: Combination of the proposed key management schemes.

achieve more efficiency in term of communication overhead, but do not ensure strong forward secrecy. Thus, we can propose to combine our proposed key management schemes as shown in Figure 17. First, the utility chooses a key tree to construct the multi-group key graph. Then, it can choose either using individual or batch rekeying scheme taking to the advantages and disadvantages of each rekeying method.

9. Conclusion

In this paper, we proposed new versatile, efficient, flexible and scalable key management schemes for advanced metering infrastructure in smart grid, to secure unicast, multicast, and broadcast communications. We designed a novel multi-group key graph structure, as well as algorithms for rekey operations, which allow the multiple Demand Response programs subscription for each customer while maintaining the group secrecy, backward/forward secrecy, and collusion freedom. It is shown in our analysis and simulations that, in comparison to two existing key management schemes for advanced metering infrastructure (Liu's *et al.* scheme and SKM+), our schemes induce low storage overhead without increasing the communication overhead and the time of key distribution. Thus, the proposed solutions meet the different needs of the power utility.

References

[1] E.A. Lee, "Cyber Physical Systems: Design Challenges," in *11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pp.363-369, May 2008.

[2] H. Farhangi, "The path of the smart grid," in *IEEE Power & Energy Mag.*, vol. 8, no. 1, pp.18-28, Jan. 2010.

[3] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid - The new and improved power grid: A survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944?980, Fourth Quarter 2012.

[4] Ye Yan, Yi Qian, H. Sharif, and D. Tipper, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges," in *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 5-20, First Quarter 2013.

[5] Z.M. Fadlullah *et al.*, "Toward Intelligent Machine-to-Machine Communications in Smart Grid," in *IEEE Communications Magazine*, vol. 49, no. 4, pp. 60-65, Apr. 2011.

[6] R.R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on Advanced Metering Infrastructure," in *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 473-484, Dec. 2014.

[7] A. Anzalchi, and A. Sarwat, "A survey on security assessment of metering infrastructure in Smart Grid systems," in *Proceedings of the IEEE SoutheastCon 2015*, pp. 1-4, Apr. 2015.

[8] Federal Energy Regulatory Commission. "Assessment of Demand Response and Advanced Metering," Staff Report, Dec. 2014, [Online]. Available: http://www.ferc.gov/legal/staff-reports/2014/demand-response.pdf. [Accessed: Dec 21, 2016].

[9] R. Deng, Z. Yang; M.Y. Chow, and J. Chen, "A Survey on Demand Response in Smart Grids: Mathematical Models and Approaches," in *IEEE Transactions on Industrial Informatics*, vol.11, no.3, pp.570-582, Jun. 2015

[10] F.M. Cleveland, "Cyber Security Issues for Advanced Metering Infrasttructure (AMI)," in *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1-5, Jul. 2008.

[11] E.L. Quinn, "Smart metering & privacy : existing law and competing policies," A report for the Colorado Public Utilities Commission, University Colorado Law School - CEES, May 2009.

[12] W. Wang, and Z. Lu, "Cyber security in the Smart Grid: Survey and challenges," in *Comp. Networks*, vol. 57, no. 5, pp. 1344-1371, Apr. 2013.

[13] S. Das, Y. Ohba, M. Kanda, D. Famolari, and S.K. Das, "A key management framework for AMI networks in smart grid," in *IEEE Communications Magazine*, vol. 50, no. 8, pp. 30-37, Aug. 2013.

[14] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An Efficient Merkle-Tree-Based Authentication Scheme for Smart Grid," in *IEEE Systems Journal*, vol.8, no.2, pp.655-663, Jun. 2014.

[15] Y. Liu, C. Cheng, T. Gu, T. Jiang, and X. Li, "A Lightweight Authenticated Communication Scheme for Smart Grid," in *IEEE Sensors Journal*, vol.16, no.3, pp.836-842, Feb. 2016.

[16] J. Kamto, L. Qian, J. Fuller, and J. Attia, "Light-weight key distribution and management for Advanced Metering Infrastructure", in *IEEE GLOBECOM Workshops (GC Wkshps)*, pp. 1216-1220, Dec. 2011.

[17] Q. Li, and G. Cao, "Multicast authentication in the smart grid with one time signature," in *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 686-696, Dec. 2011.

[18] H. Nicanfar, P. Jokar, and V.C.M. Leung, "Smart grid authentication and key management for unicast and multicast communications," in *IEEE PES Innovative Smart Grid Technologies Asia (ISGT)*, pp. 1-8, Nov. 2011.

[19] D. Wu, and C. Zhou, "Fault-tolerant and scalable key management for smart grid," in *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 375-381, Jun. 2011.

[20] J. Xia, and Y. Wang, "Secure Key Distribution for the Smart Grid," in *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1437-1443, Sep. 2012.

[21] Ye Yan, R.Q. Hu, S.K. Das, H. Sharif, and Yi Qian, "An efficient security protocol for advanced metering infrastructure in smart grid," in *IEEE Network*, vol. 27, no. 4, pp. 46-71, Jul. 2013.

[22] N. Liu, J. Chen, L. Zhu, J. Zhang, and Y. He, "A Key Management Scheme for Secure Communications of Advanced Metering Infrastructure in Smart Grid," in *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, pp. 4746-4756, Oct. 2013.

[23] Z. Wan, G. Wang, Y. Yang, and S. Shi, "SKM: Scalable Key Management for Advanced Metering Infrastructure in Smart Grids," in *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 7055-7066, Dec. 2014.

[24] M. Nabeel, X. Ding, S.H. Seo, and E. Bertino, "Scalable end-to-end security for advanced metering infrastructures," in *Information Systems*, vol.53, pp. 213-223, Oct. 2015.

[25] A. Mohammadali, M. Sayad Haghighi, M.H. Tadayon, and A. Mohammadi Nodooshan, "A Novel Identity-Based Key Establishment Method for Advanced Metering Infrastructure in Smart Grid," in *IEEE Transactions on Smart Grid*, vol.PP, no.99, pp.1-1, Oct. 2016.

[26] M. Mahmoud, J. Misic, and X. Shen, "Efficient public-key certificate revocation schemes for smart grid," in *Proc. of IEEE Global Communication Conference, Atlanta, GA, USA*, pp. 778-783, Dec. 2013.

[27] M. Mahmoud, J. Misicc, K. Akkaya, and X. Shen, "Investigating public-key certificate revocation in smart grid," in *IEEE Internet Things journal*, vol. 2, no. 6, pp. 490-503, Dec. 2015.

[28] M. Mahmoud, K. Akkaya, K. Rabieh, and S. Tonyali, "An efficient certificate revocation scheme for large-scale AMI networks," in *Proc. IEEE Int. Perform. Comput. Commun. Conf. (IPCCC), Austin, TX, USA*, pp. 18, Dec. 2014.

[29] K. Rabieh, M. Mahmoud, K. Akkaya and S. Tonyali, "Scalable Certificate Revocation Schemes for Smart Grid AMI Networks Using Bloom Filters," in *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 420-432, Jul-Aug 2017.

[30] K. Akkaya, K. Rabieh, M. Mahmoud and S. Tonyali, "Customized Certificate Revocation Lists for IEEE 802.11s-Based Smart Grid AMI Networks," in *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2366-2374, Sep. 2015.

[31] NIST, "Guidelines for Smart Grid Cybersecurity," NISTIR 7628 Revision 1, Sep. 2014, [Online]. Available: http://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf. [Accessed: Dec 29, 2016].

[32] W. Diffie, and M.E. Hellman, "New directions in cryptography, " in *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, Nov. 1976

[33] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science*, vol 196, pp. 47-53, 1984.

[34] R. M. Needham, and M. D. Schroeder, "Using encryption for authentication in large networks of computers," in *Communications of the ACM*, vol. 21, no. 12, pp. 993-999, Dec. 1978.

[35] V.S. Miller, "Use of Elliptic Curves in Cryptography," in *Advances in Cryptology : Proceedings of CRYPTO 85, Lecture Notes in Computer Science*, vol. 218, pp. 417-426, 1986.

[36] C. K. Wong, M. Gouda, and S. Lam, "Secure group communication using key graphs," in *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16-30, Feb. 2000.

[37] L. Chen, and C. Kudla, "Identity based authenticated key agreement protocols from pairings," in *Proc. IEEE CSFW, Pacific Grove, CA, USA*, pp. 219-233, Jun. 2003.

[38] D. A. McGrew, and A. T. Sherman, "Key establishment in large dynamic groups: Using one-way function trees," in *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444-458, May 2003.

[39] T. Sauter, and M. Lobashov, "End-to-End Communication Architecture for Smart Grids," in *IEEE Transactions on Industrial Electronics*, vol. 58, no. 4, pp. 1218-1228, Apr. 2011.

[40] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in HAN, NAN and WAN," in *Computer Networks*, vol. 67, pp. 74-88, Jul. 2014.

[41] T. Ericson, "Direct load control of residential water heaters," in *Energy Policy*, vol. 37, no. 9 , pp. 3502-3512, Sep. 2009.

[42] R. Tyagi, J.W. Black, "Emergency demand response for distribution system contingencies," in *2010 IEEE PES Transmission and Distribution Conference and Exposition*, pp. 1-4, Apr. 2010.

[43] K. Herter, "Residential implementation of critical-peak pricing of electricity," in *Energy Policy*, vol. 35, no. 4 , pp. 2121-2130, Apr. 2007.

[44] R. Tan, "Integrity Attacks on Real-Time Pricing in Electric Power Grids," in *ACM Transactions on Information and System Security*, vol. 18, no. 2 , pp. 5:1-5:33, Dec. 2015.

[45] NIST, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography," NIST Special Publication 800-56A (Revision 2), May 2013, [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf. [Accessed: Dec 27, 2016].

[46] J. Liu, and B. Yang, "Collusion-Resistant Multicast Key Distribution Based on Homomorphic One-Way Function Trees," in *IEEE Transactions on Information Forensics and Security*, vol.6, no.3, pp.980-991, Sep. 2011.

[47] X.S. Li, Y.R. Yang, M.G. Gouda, and S.S. Lam, "Batch Rekeying for Secure Group Communications," in *Proceedings of the 10th International Conference on World Wide Web WWW'01*, pp. 525-534, 2001.