applied sciences

MDPI

Article

Security Management for an Advanced Metering Infrastructure (AMI) System of Smart Electrical Grids

Ahmed A. Abdullah [1], B. M. El-den [1], Khaled M. Abo-Al-Ez [2] and Tarek M. Hassan [1,*]

[1] Faculty of Engineering, Delta University for Science and Technology, Gamasa 35712, Egypt; ahmed.abdelaleem@deltauniv.edu.eg (A.A.A.); basant.ibrahim@deltauniv.edu.eg (B.M.E.-d.)
[2] Centre for Power Systems Research (CPSR), Faculty of Engineering and the Built Environment, Cape Peninsula University of Technology, Cape Town 7535, South Africa; aboalezk@cput.ac.za
* Correspondence: tarek.hassan@deltauniv.edu.eg

Abstract: Advanced Metering Infrastructure (AMI) plays a crucial role in enabling the efficient functioning of Smart Electrical Grids, but its successful implementation hinges on robust cybersecurity measures. To uphold data confidentiality and integrity, the deployment of an effective key management scheme (KMS) for multiple Smart Meters (SMs) and devices is imperative. The AMI exhibits unique characteristics, including storage and computation constraints in SMs, hybrid message transmission techniques, and varying participation levels in Demand Response (DR) projects, necessitating a tailored approach to security compared to other systems. In this research, we propose a KMS that is designed to address the specific security concerns of the AMI. The scheme comprises three key management procedures catering to the unicast, broadcast, and multicast modes of hybrid transmission. Given the resource limitations of SMs, we adopted simple cryptographic techniques for key creation and refreshing policies, ensuring efficiency without compromising on security. Furthermore, considering the variability of participants in DR projects, we established key refreshing policies that adapted to changing involvement. The effectiveness and security of the proposed KMS were rigorously evaluated, demonstrating its practical applicability and ability to safeguard the AMI ecosystem. The results of the evaluation indicate that our approach provides a viable and robust solution to the security challenges faced by AMI systems. By employing the proposed KMS, stakeholders can confidently deploy and manage AMI, ensuring the protection of sensitive data and maintaining the integrity of the Smart Electrical Grid.

Keywords: Advanced Metering Infrastructure (AMI); data transmission mode; smart grid; Key Management Scheme (KMS)

Citation: Abdullah, A.A.; El-den, B.M.; Abo-Al-Ez, K.M.; Hassan, T.M. Security Management for an Advanced Metering Infrastructure (AMI) System of Smart Electrical Grids. *Appl. Sci.* **2023**, *13*, 8990. https://doi.org/10.3390/app13158990

Academic Editor: Gerard Ghibaudo

Received: 1 July 2023
Revised: 27 July 2023
Accepted: 1 August 2023
Published: 5 August 2023

1. Introduction

The Advanced Metering Infrastructure (AMI) plays a pivotal role in modernizing traditional power distribution systems by enabling bidirectional communication and data exchange between utility providers and end-users. This transformation is a critical element of the Smart Electrical Grid, enhancing energy efficiency and grid reliability and enabling the integration of renewable energy sources. However, with an increasing reliance on interconnected digital technologies, the AMI system also becomes vulnerable to various cyber threats, making robust security management an indispensable aspect of its successful implementation [1].

The fundamental objective of an AMI system is to collect real-time data from Smart Meters (SMs) installed at consumer premises and communicate this information back to utility companies for billing, load balancing, and demand response purposes. The data transmitted through the AMI system include sensitive information about energy consumption patterns, user behavior, and potentially private details of consumers. Consequently, the confidentiality and integrity of these data are paramount to protecting the privacy and security of consumers and maintaining the overall stability of the electrical grid [1,2].

To safeguard the AMI system against potential cyber-attacks, it is essential to establish a comprehensive and robust Security Management framework. The Security Management approach for AMI should encompass various aspects, including but not limited to authentication, access control, encryption, and key management. In this paper, we focus on discussing the modeling of Security Management for the Advanced Metering Infrastructure (AMI) system of Smart Electrical Grids with specific emphasis on the Key Management Scheme (KMS) that is designed to address the unique security challenges of AMI [3].

Security considerations for AMI differ significantly from conventional IT systems due to several distinct features. First, Smart Meters (SMs) often have limited storage and computational capabilities, necessitating the development of lightweight cryptographic techniques to ensure secure key generation, distribution, and management. Second, the AMI system employs a hybrid message transmission approach that combines unicast, broadcast, and multicast modes, introducing complexities when managing encryption keys effectively. Finally, the participation of consumers in Demand Response (DR) projects can vary over time, demanding flexible key refreshing policies to accommodate changes in the AMI ecosystem [4].

SMs are a major component of modern smart grids (SG). Since SMs have limited computation power and are positioned at an exceedingly long distance from the utility, Advanced Metering Infrastructure (AMI) is an essential component in the SG that represents the structure of a complex communication network. It integrates SMs, Monitoring Systems (MS), sensing devices, and the Meter Data Management System (MDMS) [1]. Through AMI, utilities can congregate their fundamental targets for revenue protection and load management. AMI gathers instantaneous information about individual and aggregated demand, puts caps on consumption, and performs various revenue models to control their costs. The application of AMI in the SG enables greater control and operation with smaller margins. This scheme requires a sort of communication and coordination between the different components of the power grid. For years, data exchanged in traditional power line communications were only electrical system measurements, such as voltage measurements, and these data were not considered to be as valuable as financial information [2,3]. However, with the changing paradigms in modern power system operations, financial information is currently exchanged in power system communications for the purpose of deregulated energy markets. In these energy markets, a small error in the power information (both electrical and financial) can lead to a disruption in the bids of highly competitive electricity markets. This raised the likelihood of cyber security issues and necessitated the integration of security protocols within SG for comprehensive security functionality [4,5]. SG's evolution to a cyber-physical system by deploying AMI makes it prone to cyber threats aside from SCADA vulnerabilities. AMI cyber security, from an oriented threat perspective, impacts data confidentiality, integrity, availability, and accountability [6]. The most crucial security issues that must be resolved before AMI can be deployed are message confidentiality for user privacy and behavior, message authentication for meter readings (DR), and load control communications. Using encryption and authentication methods that rely on the safety of cryptographic keys can resolve confidentiality and integrity difficulties. In AMI systems, key management for several devices is essential for security. The key management scheme consists of a key organizational structure, key generation, updating, distribution, storage laws, etc. (KMS). Recently, a number of studies on the key management of AMI systems were released. An example of a complete system for secure AMI communications, including secrecy and authentication, is presented in [7]. Using mutual authentications, this system can provide trust services, data privacy, and integrity. A survey on key management and authentication approaches in smart metering systems is presented in [8], including security measures like the Advanced Encryption Standard (AES) [8], encryption, and public-key infrastructure authentication.

The focus of this paper is on-device authentication, data secrecy, and message authentication mechanisms for Advanced Metering Infrastructure (AMI) applications. The outcomes have effects on key management. There is not a complete solution yet for man-

aging keys in AMI systems for various smart devices. If the AMI network is constructed on a wireless sensor network, the authors in [9] propose a key establishment and security technique based on public-key cryptography. Depending on the stage of smart grid development in various nations, different AMI network types exist [10,11]. The significance of key management for a large number of devices in AMI is important for the security preservation of cryptographic keys [12]. Moreover, a few KMSs have been set up to enable secure communication between SCADA systems and wide-area protection systems: two examples of power control systems [13,14]. However, because of the differences in these systems' architectures, message characteristics, and needs, these KMSs, along with those utilized in general IT systems, have difficulty when immediately adapted to an AMI system.

We may infer from the findings of the available study that the suggested KMSs are for a certain AMI system. When application functions, communications, and information technology change, the KMS can no longer be appropriate. However, the majority of AMI systems have still been in the experimental stage until recently; therefore, the future is still unknown. Additionally, there are regional and national differences in the design and development of an AMI system. The functionalities that must be deployed are different from the standpoint of the application's needs. The applications are quite straightforward, like metering, measuring, and monitoring; the other applications are overly complicated and focus on DR and load management applications. According to the needs of the program and user desire, there are several communication options as well.

Due to these aforementioned issues, we are attempting to recommend a more widespread KMS for AMI systems. To summarize AMI's characteristics and development, we believe that its structure and constituent parts are stable. In other words, key management's primary, ongoing goal is securing SMs. In order to manage the keys of several SMs, a key management framework based on a key graph is suggested. Although the AMI's functions are not set in stone, we can create a KMS to hold all of the potential functions. In practice, users can select a portion of the KMS for particular applications. Although the properties of messages sent through communication channels can be determined by the function needed, communications are also not fixed. To address this issue of managing keys for multiple smart devices in AMI systems, this research proposes three different key management methods for broadcast, unicast, and multicast communications, which are designed based on functional requirements and message types. Each of these methods includes specific procedures for key regeneration and refreshing, considering factors such as the computation and storage limitations of SMs, the time requirements of functions, and other relevant factors.

The remaining sections of this paper are arranged as follows: Section 2 examines the framework and also the message flow for an AMI system. KMS design difficulties with the AMI system are introduced in Section 3. Section 4 represents the KMS design difficulties with an AMI system. The key management and key refreshing policy for unicast, broadcast and multicast communication is discussed in Sections 5–7, respectively. Section 8 introduces the security examination. In Section 9, the performance analysis in terms of the cost of storage, the time cost for computation, and the time cost for distributions are discussed. In Section 10, a conclusion is introduced.

2. Advanced Metering Infrastructure (AMI) Features

This section presents the AMI's main features, including the framework and the interconnection messages.

2.1. Framework of AMI System

As shown in Figure 1, the AMI system consists of various technologies and applications that combine to function as a single entity. These technologies are SMs, user gateways (UGs), wide-area communication infrastructure, and Meter Data Management Systems (MDMSs). Brief descriptions of these technologies are mentioned below:

- **SMs:** SM is a programmable solid-state device that can perform a variety of functions, including Bi-directional metering, including the real-time monitoring of power consumption, peak demand, voltage, current, frequency, and power factor. They can support net metering, transmit notifications of power outages or restoration, monitor power quality, permit remote turn-on or turn-off operations, and assist with time-based pricing. They can also give consumption statistics for both consumers and utilities. Additionally, they can help with (DR) goals [15], which enable greater energy efficiency because information feedback has been demonstrated to decrease customer consumption [16].
- **UGs:** The Universal Gateway (UG) is responsible for performing protocol switching and facilitating communication between two different networks, such as the wide area network and the in-home network. This function is typically conducted by other devices, like SMs or personal computers (Pcs).
- **HANs:** A Home Area Network (HAN) is a specific type of local area network that connects to Distributed Energy Resources (DERs), local control devices, SMs, and the (UG) [16,17].
- **Wide-area communications infrastructure:** They support continuous contact between the utility, customer, and the controlled electrical load. It has stringent privacy restrictions while using open, two-way communication protocols. One popular architectural [18] feature is the use of local aggregators to compile data from collections of meters SMs and send them to a central server. Radiofrequency, optical fiber, the power line carrier, the Internet. . .etc., may all be considered for use in providing any or this entire infrastructure.
- **Meter Data Management System (MDMS):** MDMS is a database of meter data that includes analytical capabilities and is connected to the user side through an AMI head end. It communicates with other electric power utility systems, such as customer information systems, outage management systems, distribution management systems, and so on, through enterprise buses.

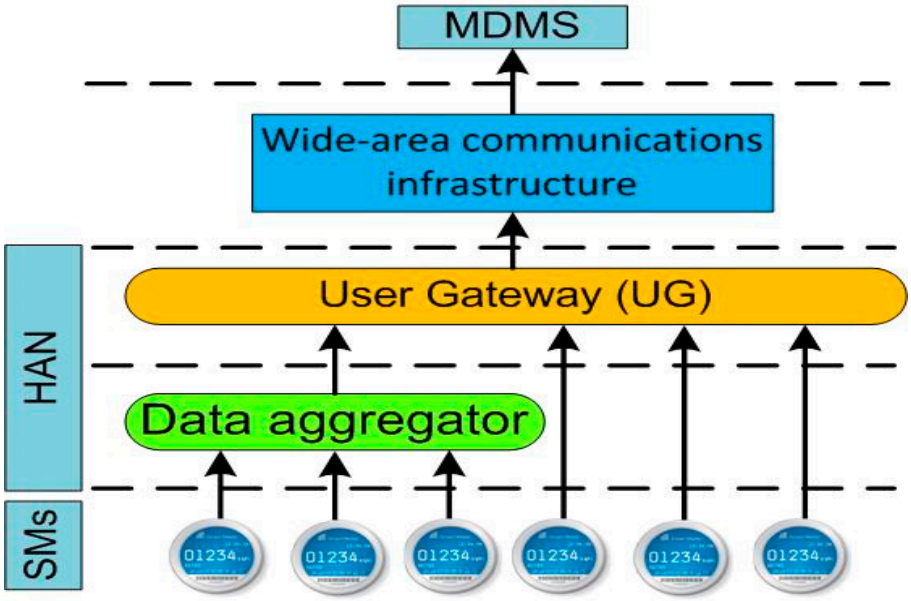

Figure 1. AMI framework.

2.2. Interconnection Messages in AMI System

Over AMI communication networks, interactive messages are sent and received between smart grid devices. These messages include topics such as electricity pricing, remote load management, meter data, notifications for power outages and returns, publishing DR projects, subscribing to or canceling DR projects, and more. There are two components involved in the sending and receiving of these communications. The first one is user-level

devices (SMs, UGs, etc.) which are the sending components. To keep things short, we refer to these devices as NXs. The other component is the data management system (DMS) which aids in the transmission and reception of AMI data.

Messages may be categorized as either unicast, broadcast, or multicast. The unicast message is sent from DMS to NX, but broadcast messages are sent to all NXs, whereas multicast messages are sent to DMS and a subset of NXs participating in the same DR project. Table 1 details the types of messages, along with their respective senders, recipients, and modes of transmission [19]. This table also includes the time requirements for various message types based on the requirements of the State Grid Corporation of China. Even though these time estimates are not identical in every country, their variances are negligible when considering the global smart grid's overall development goal.

Table 1. AMI messages with transmission mode and time requirements [19].

Message Type	Sender	Receiver	Transmission Mode	Time Requirements
Meter data	NX DMS	DMS NXs	Unicast Broadcast	<15 s <15 s
Joining or leaving the DR	NX	DMS	Unicast	<15 s
Pricing	DMS	NXs/DR	Broadcast/multicast	<15 s
Remote control	DMS	NXs/DR	Broadcast/multicast	<5 s
Notifications	NX	DMS	Unicast	<5 s
Publishing DR	DMS	NXs	Broadcast	<5 s

3. A KMS Design Challenge with AMI System

The standard components of any KMS may include key management frameworks, key generation and refreshment rules, key distribution policies, and key storage techniques. Conclusions about the challenges of creating the KMS may be drawn from the AMI's structure and messaging as follows:

- Hybrid transmission techniques are utilized in bidirectional communications, such as broadcast, unicast, and multicast. It is expected that the KMS will be able to accommodate all of these transmission modalities. In order to find a solution to this issue, the KMS must be adaptable enough to accommodate all three distinct forms of transmission. The protocols governing the production, updating, and distribution of keys are each crafted specifically for each mode.
- The NXs are usually implemented using embedded systems. When compared to a typical computer or server, they are less powerful both computationally and in terms of storage. Additionally, the messages call for limited-time broadcasts. The key generation and refreshing algorithms are most significantly impacted by NXs' low computational capabilities. Primarily, quite simple cryptographic methods like hashing might be utilized. Additionally, the frequency of key distribution should be kept to a minimum because of the restricted time for message delivery. Similarly, it is best to reduce the number of keys and other relevant data that must be saved in NXs.
- Projects using DR users do not have fixed users. Depending on their preferences and needs, they may choose projects at various periods. Users who participate in the same DR project may band together to establish groups; these users are known as group members.
- Due to the dynamic nature of group participation in DR projects, ensuring both forward and backward security in the multicast mode becomes critical. New users should be assigned to groups when they join the project, and each group member should receive new keys and additional data to ensure secure communication. While users who leave the project no longer receive updates, shared keys, and additional data should be refreshed frequently to preserve security.

4. Key Management Framework (KMF)

In this section, the definitions, common abbreviations, and KMS for the AMI system are explained. Abbreviations specifies a set of abbreviations and their meanings for use in the KMS.

Based on the key graph concept [20], an AMI system's key management architecture is created. This is conducted in order to manage all three distinct forms of message transmission. As shown in Figure 2, KMF can be defined as:

$$\text{KMF} = (\text{U,K,R}) \tag{1}$$

where:

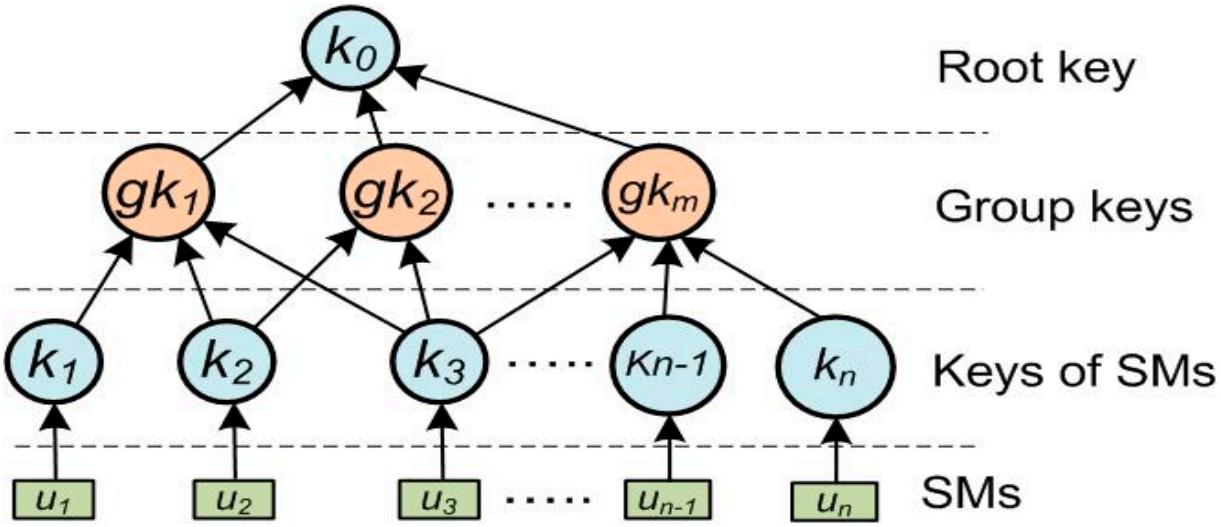

Figure 2. The U, K, and R for KMF.

$U = \{u_1, u_2, \ldots, u_n\}$ This is a set that is finite and non-empty and represents NXs within the AMI system.

$K = \{k_0, k_1, \ldots, k_n, kg_1, kg_2, kg_m\}$. This is a set that is finite and non-empty, where $\{k_1, k_2, \ldots, k_n\}$ represents the keys of NXs. {gk1, gk2, …, gkm}. "k" represents the group keys of demand response projects, and "k_0" is the root key of the key hierarchy.

R: The user–key relation is a binary relation between U and K, denoted as $Rk \subset U \times K$. It indicates that user u knows key k if and only if (u, k) is present in R.

In addition to this, a function is connected to the set, and its definition is as follows in (2):

$$userset(k) = \{u|(u,k) \in R\} \tag{2}$$

For example, of the AMI, the KMF shown in Figure 3 might be expressed in (3), (4), and (5), for instance:

$$U = \{u_1, u_2, u_3\} \tag{3}$$

$$K = \{k_0, k_1, k_2, k_3\} \cup \{gk_1, gk_2\} \tag{4}$$

$$R = \left\{ \begin{array}{c} (u_1, k_0),\ (u_1, k_1), (u_1, gk_1), \\ (u_2, k_0),\ (u_2, k_2), (u_2, gk_1), (u_2, gk_2), \\ (u_3, k_0), (u_3, k_3), (u_3, gk_2) \end{array} \right\} \tag{5}$$

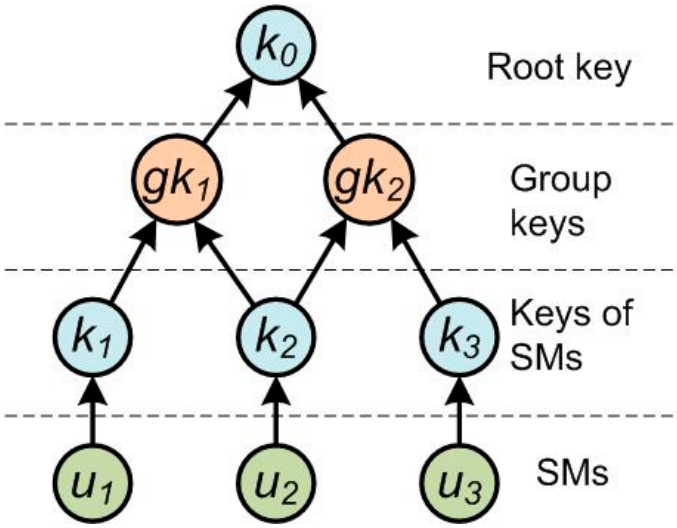

Figure 3. An instance of the key management framework.

The set user set identifies the users involved in DR project $1(\mathrm{gk}_1) = \{\mathrm{u}_1, \mathrm{u}_2\}$, while the users participating in DR project 2 are denoted by the set user set $((\mathrm{gk}_2) = \{\mathrm{u}_2, \mathrm{u}_3\}$. For the broadcast mode, the KMF uses the group keys, and the root key k_0. $[\mathrm{gk}_1, \mathrm{gk}_2, \ldots, \mathrm{gk}_\mathrm{m}]$ are the keys for the multicast mode in several DR project groups $[\mathrm{k}_1, \mathrm{k}_2, \ldots, \mathrm{k}_\mathrm{n}\ ]$ are for each MS and each SX in unicast mode. The core of key management is these keys, so it is important to outline the procedures for their creation, distribution, and periodic refreshment. In addition, in order to authenticate and verify messages, as well as encrypt and decrypt them, session keys are required throughout the communication process. As a result, the procedures used to create, distribute, and renew session keys must be carefully designed while considering the amount of traffic on the network and the computing power of SXs.

In order to provide a concise introduction to the KMS, the MS is denoted by the symbol u_0. First, the keys of KMF and the extra value for creating session keys are created by u_0 via the use of certain key servers. After that, the keys are disseminated to SXs. The initialization procedure is shown in Figure 4, which is then followed by explanations in further depth.

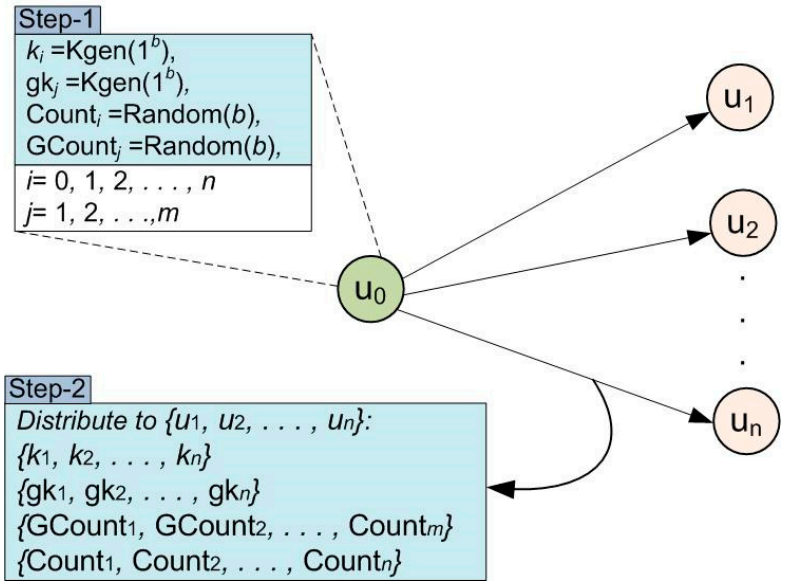

Figure 4. Initial keys and their additional value to distribute to SXs.

Step 1: The first b-bit keys for each SX project group and each DR project group are formed after the initial construction of the key u_0.

$$k_i = Kgen\left(1^b\right),\ i = 0,1,2,\ldots,n$$
$$gk_j = Kgen\left(1^b\right),\ j = 0,1,2,\ldots,m$$
$$Count_i = Random(b),\ i = 0,1,2,\ldots,n$$
$$GCount_j = Random(b), j = 0,1,2,\ldots,m$$

Step 2: The initial keys are distributed, as well as the extra value.

$$u_0 = \{u_1,\ u_2,\ldots,\ u_n\}:\ k_i,\ gk_j,\ Count_i,\ GCount_j, i = 0,1,2,\ldots,n,\ \ j = 1,2,\ldots,m$$

The user keys k_1, k_2, …, k_n as well as the relevant extra numbers $Count_1$, $Count_2$, …, $Count_n$ are transmitted to "u_1, u_2, …, u_n", respectively, using safe ways. Additionally, the root key k_0 and the extra value $Count_0$ are dispersed among "u_1, u_2, …, u_n." The keys "$gk_1, gk_2, \ldots, gk_m$" and the accompanying supplementary values "$GCount_1, GCount_2, \ldots, GCount_m$" are distributed among SXs in accordance with the users' involvement preferences in DR projects. These preferences are reflected in the GCount values.

5. Unicast Communication

This section presents the key management and key refreshing policy for unicast communication.

5.1. Key Management for Unicast Communication

Messages in AMI systems were analyzed, and the results revealed that the unicast transmission method comprises three distinct kinds of messages: the remote load control, meter data, and subscription to or cancellation of DR projects. The messages may proceed either way, from the MS to the SX or from the SX to the MS. It is essential that both the secrecy and the integrity of the communications are maintained throughout the process of communication. In order to accomplish this goal, the session key has to be reset at the beginning of each session.

There are three distinct groups into which the key management steps for unicast communication can be divided. The phases for the unicasting communication method are shown in Figure 5.

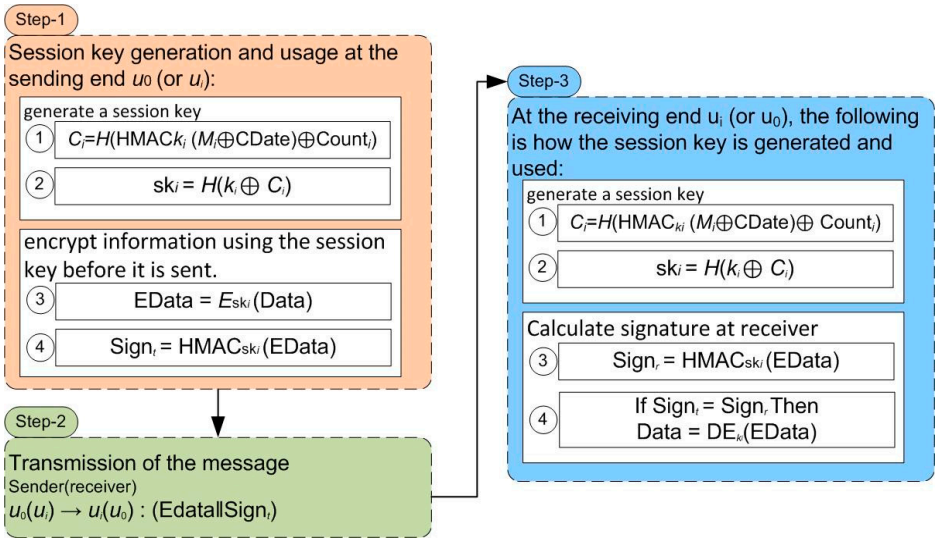

Figure 5. Steps for unicasting communication process.

An initial session key generation is required. The session key ski is built in Steps 1.1 and 1.2 using the metering data Mi, the metering date CDATE, the value Counti, and the user key ki. In Step 1.3 and Step 1.4, upcoming information for sharing is encrypted and signed using this session key before it is sent. In step 2, the data that have been encrypted together with the signature value are sent to the recipient. In step 3, the receiver is able to obtain the components that are required to produce the session key. Step 3.1 and Step 3.2 are responsible for the generation of the session key ski. It is possible to decode the data and verify the signature using the session key provided. Step 3.3 and Step 3.4 are responsible for verifying the signature between the transmitter and receiver before extracting the data using the decryption function (DE). Figure 6 shows the flowchart for the unicasting communication process.

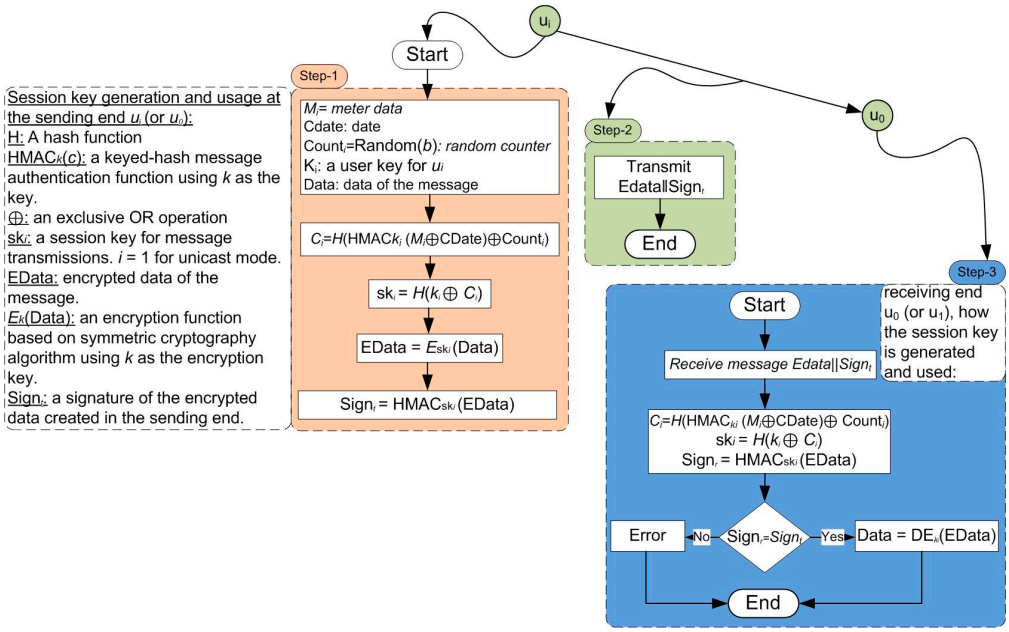

Figure 6. Flowchart for the unicasting communication process.

5.2. Key Refreshing Policy for Unicast Communication

It is recommended that the user key k_i should be renewed at set intervals, such as once every day, once per week, and so on, to ensure that the key is always in its most up-to-date state. If you decide to utilize a HASH function as the key refreshing technique, in addition to lowering the price of key distribution in the network, you can guarantee independence between the new and old keys. Before each session, the session keys sk_i ought to be regenerated so that the confidentiality of the information can be maintained. Because there are so many SXs, the process of refreshing the session keys in the MS and then distributing them might result in a significant increase in the cost of the network traffic. On the other hand, this problem can be resolved if they are appropriately updated in both MS and SX in accordance with the chosen method.

If they were able to be appropriately refreshed in both MS and SX in accordance with a predetermined plan, this issue would not arise. The fact that part of the data that are utilized for key refreshing may be readily acquired by either side is the most significant aspect of the agreed-upon technique. As a result of our research, we discovered that the SX and MS were able to obtain the daily metering data of an electrical customer. This is due to the fact that metering is an essential feature of AMI systems. This has led to the creation of a unique session key refreshing approach that is based on metering data. The renewing rules for keys and variables in unicast communication are listed in Table 2.

Table 2. Updating keys and variable policies for unicast communication.

Variable	Refreshing Period	Refreshing Algorithm
K_i	Periodically	$K_i = H(K_i)$
$Count_i$	Every session	$Count_i = Count_i + 1$
C_i	Every session	$C_i = H(HMAC_{ki}(M_i \oplus CDate) \oplus Count_i)$
sk_i	Every session	$sk_i = H(K_i \oplus C_i)$

6. Broadcast Communication

This section presents the key management and key refreshing policy for broadcast communication.

6.1. Key Management for Broadcast Communication

The publishing of DR projects and information on energy prices are both messages that may be delivered using the broadcasting technique in order to ensure the message's secrecy and authenticity; the session keys must be updated before each broadcast session. The three stages of detailed key management for broadcast communication are depicted in Figure 7.

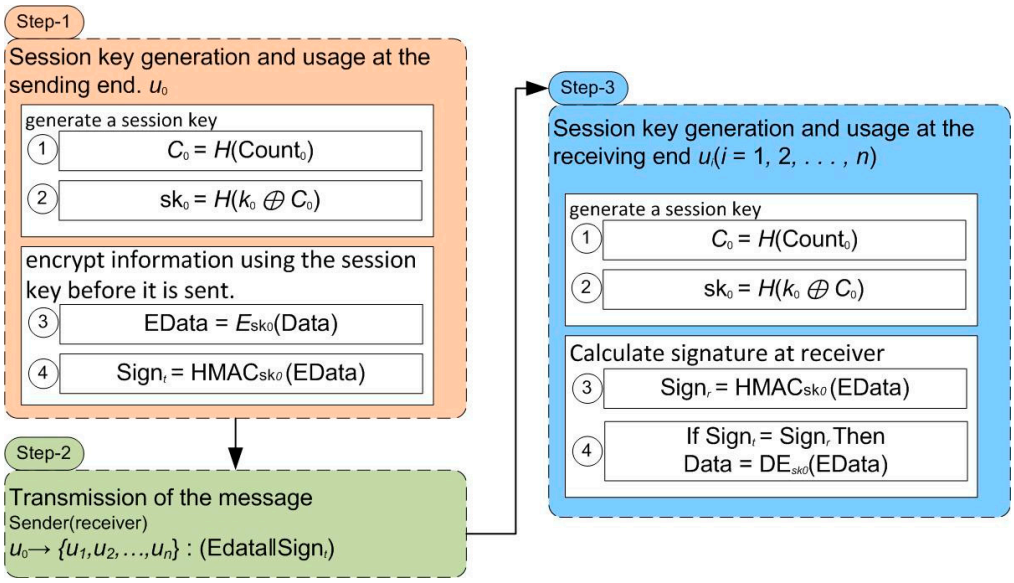

Figure 7. Steps for broadcast communication process.

The variables $Count_0$ and the root key k_0 are used in Step 1.1 and Step 1.2 to produce the session key sk_0. The information for transfer is encrypted and signed using this session key before being sent in steps 1.3 and 1.4. In step 2, the data that have been encrypted together with the signature value are then sent to all the recipients of SXs. Step 3.1 and Step 3.2 generate the session key at the recipients then Step 3.3 calculates the receiver signature. In Step 3.4, the recipients compare the transmitter signature and recipient signature, if there are equal, and the information is decrypted.

6.2. Key Refreshing Policy for Broadcast Communication

The key refreshing policy for unicast communication and broadcast communication are almost identical. This is because both types of communication use the same key. It is recommended that k_0 be updated at regular intervals, such as once per day or once per month, to maintain its validity. If a HASH function is employed, the key independence between new and old keys would perform admirably. Both new and old keys would be affected by this. Before each session, the broadcast communication session key sk_0 has to

have its cache cleared. In addition, the implementation of the HASH function and routine k_0 and C_0 resets would be an excellent way to secure the independence of session keys as well as their unpredictability. The renewing of rules for keys and variables in broadcast communications is listed in Table 3.

Table 3. Refreshing policies for keys and variables in broadcast communication.

Variable	Refreshing Period	Refreshing Algorithm
K_0	Periodically	$K_0 = H(K_0)$
$Count_0$	Every session	$Count_0 = Count_0 + 1$
C_0	Every session	$C_0 = H(Count_0)$
sk_0	Every session	$sk_0 = H(K_0 \oplus C_0)$

7. Multicast Communication

This section presents the key management and key refreshing policy for multicast communication.

7.1. Key Management for Multicast Communication

The multicast mode is an option for the transmission of electrical price information as well as remote load control in all different kinds of AMI messages The users who subscribe to a DR project are not permanently set; hence, the group's members that are designated to receive multicast messages should be updated at regular intervals (for example, once every day or once every week) in accordance with the current state of the electric power utilities. As a result, the key management for multicast communication has been split up into two distinct pieces. The first is comparable to a broadcast communication's key. Additionally, each new session must begin with the production of the multicast communication session key. The group key and extra value should be generated and updated with the help of unicast communication for the second key, taking into mind that users might participate in or withdraw from a DR project.

The generation and use of session keys within a DR project group are quite similar to broadcast communication. The main factor that distinguishes the receivers is their range. The process for generating a session key is shown in Figure 8.

It is necessary to generate a session key initially. In steps 1.1 and step 1.2, the value GCountj is used to determine the group key gkj, which is used to generate the session key gskj. The information for transfer is encrypted and signed using this session key before it is sent in Steps 1.3 and 1.4. The receipts of DR project j receive data that have been encrypted together with the signature value. In step 3.1 and step 3.2, the receipts in the DR project regenerate the session key gskj. It is possible to decode the data and verify the signature using the session key provided in Step 3.3 and step 3.4.

When a user makes the decision to participate in or withdraw from a DR project, the user is responsible for sending a message to u0. After receiving the acknowledgment, the group key is updated. Figure 8 is an illustration of the procedure as a whole. The implementation of this sub-process makes use of unicast messages. The information includes a total of seven steps, as shown in Figure 9.

Step 1 is responsible for encrypting and signing the requested information. The request message is sent to u0 in step 2. After receiving the request from the user in step 3, the u0 must first decrypt and validate the request before examining it to decide whether it is a request to subscribe to project j or to exit the project. When the user requests to leave the project, DR projects could be incompatible, and the user must abandon the one they are currently working on in order to subscribe to the new one.

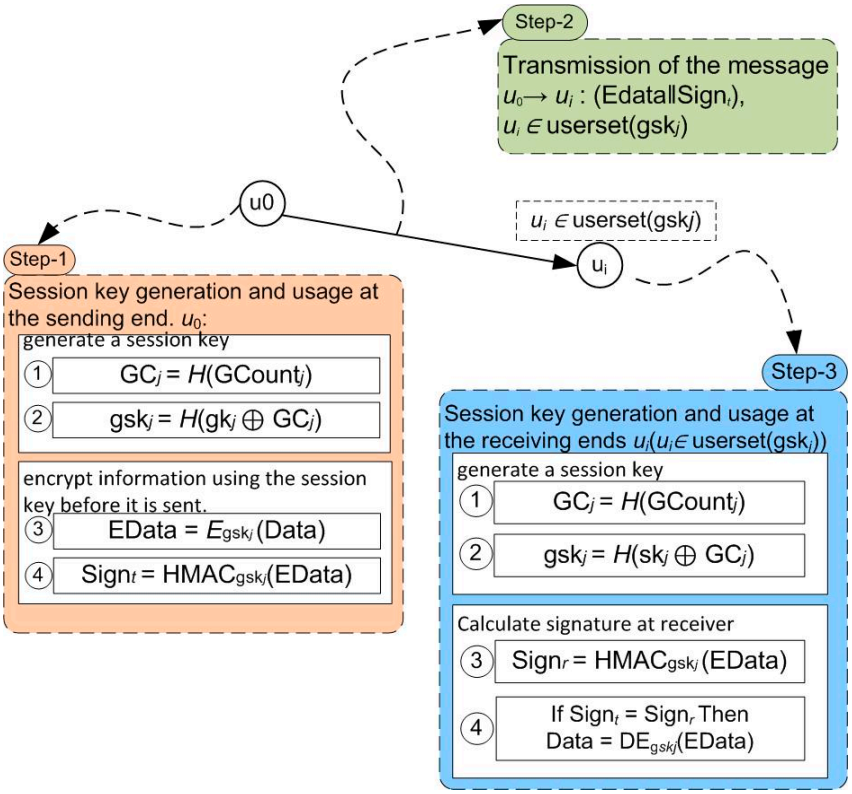

Figure 8. Steps for generation and usage of a session key in the DR project.

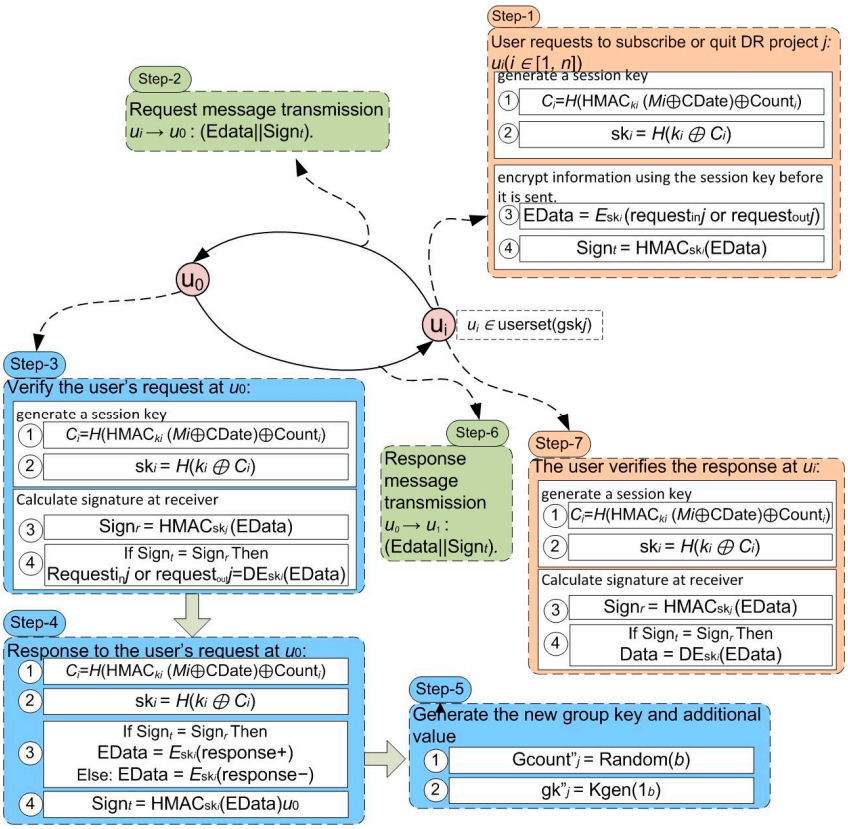

Figure 9. Seven steps for regeneration and refreshment of keys in the event that users request to leave or join a DR project.

In step 4, the u_0 response to the user's request and creates a new key for the group, as well as an additional value in step 5. In step 6, the u_0 transmit response sends a message to the user. Step 7 is responsible for notifying the user that the request was sent to the DR project and is responsible for determining whether or not it was successful.

In the case that certain people join or leave the DR project, it is important to regenerate the group key in addition to the extra value before the update time. Figure 10 depicts steps 8 through 10 for regenerating and dispersing the new group key with an additional value. Step 8 prepares to distribute the modified group key and the added value to each DR project j member. The new group key and additional value are distributed in Step 9 to every single person in the DR project j community. In step 10, the user utilizes the new session key.

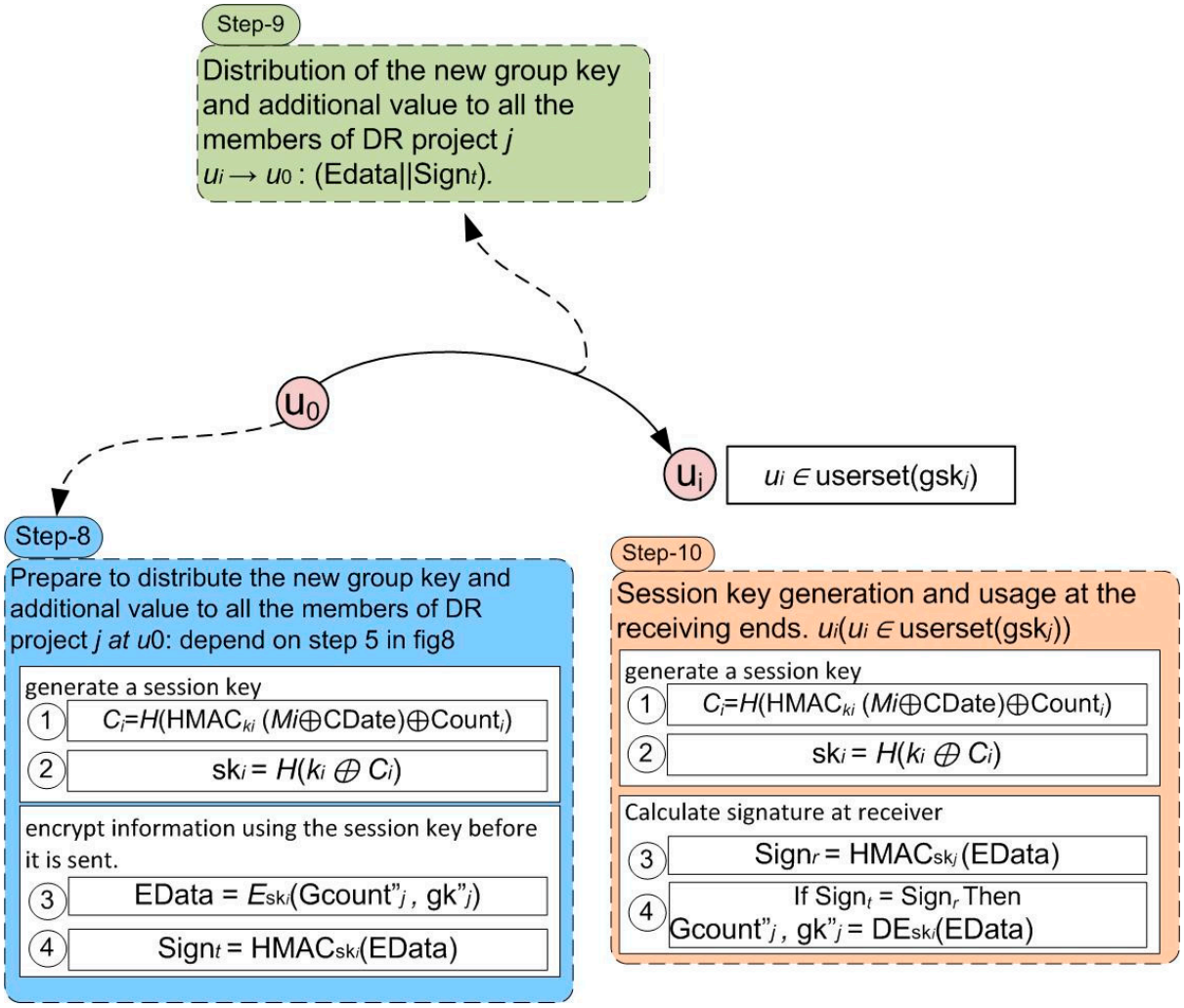

Figure 10. Steps 8 through 10 are used to regenerate, distribute, and add value to the new group key.

7.2. Key Refreshing Policy for Multi Communication

The key refreshing policy for unicast communication and broadcast communication are almost identical. Whether or not the users of the DR project change is determined by how the key refreshing policy is executed. If the users do not alter, the gkj, GCountj GCj, and gskjupdating procedures are comparable to the broadcast communication procedure from which the data are provided in Table 4. However, when users join or leave a DR project, u0 needs to regenerate gkj and GCountj and distribute them to all users in the project. The data for this process are listed in Table 5.

Table 4. The fixed DR project group's primary refreshing policy for multicast communications.

Variable	Refreshing Period	Refreshing Algorithm
gk_j	Periodically	$gk'_j = H(gk_j)$
$GCount_j$	Every session	$GCount'_j = GCount_j + 1$
GC_j	Every session	$GC_j = H(GCount_j)$
gsk_j	Every session	$gsk_j = H(sk_j \oplus j)$

Table 5. The DR project group's key refreshing policy when new users are introduced via multicast communication.

Variable	Refreshing Period	Refreshing Algorithm
gk_j	Periodically	$gk'_j = Kgen(1^b)$
$GCount_j$	Periodically	$GCount'_j = Random(b)$

8. Security Examination

A secure random key generation process using b bits is used to create both a user key and a group key. A user key or a group key is used as a starting point for the generation of a session key, which is then combined with another value and hashed. The user key provides a safe environment. The extra value, which is generated by feeding a random integer into a HASH function, is completely arbitrary and unrelated to anything else. Since this is the case, the session key is likewise safe.

The user key may be automatically updated at certain intervals inside the KMS. The number of users who are leaving or joining the DR project determines whether or not the group keys need to be refreshed. In the case that a user wants to join or leave the DR project at the time of the update, the group key must be reset.

When it comes to refreshing techniques, the HASH function is used for both the user key and the session key. The session key differs from other keys in that it can be updated using either a random additional value or metering information together with the metering date. As a direct consequence of this, the new keys function independently of the old ones.

Only the two endpoints of the communication channel have access to the session keys. The receiver first uses a secure session key to validate the digital signature associated with the encrypted material. The communication is only decrypted by the recipient if it is successful in passing authentication. In such a case, the message is deleted without being read. After this, the authenticity of the information transfer as well as its integrity is guaranteed.

Considering that people have the option to either join or leave the group that is working on the DR project, it is important to think about the group's forward and backward security. The group keys and other values are all generated and refreshed in accordance with our strategy if any user decides to join or leave the group before being sent to the new group's members. This process only takes place if there are users who make either of these decisions.

9. Performance Analysis and Results

It is necessary to store the relevant data in order to make use of the KMS. These data should include a variety of keys, counters, and other values. The information that needs to be kept in MS and SXs is outlined in Table 6. In Table 7, the calculation techniques for the storage cost of the communication endpoints are provided. These approaches are based on the information in Table 6.

Table 6. The information that needs to be saved in MS and SXs.

		MS (u_0)	**SX (u_i,$i \neq 0$)**
User keys		$k_0, k_1, k_2, \ldots, k_n$	k_0, k_i
Group keys		$gk_1, gk_2, \ldots, gk_m$	$gk_1, gk_2, \ldots, gk_m$
counters	Unicast	$Count_1, Count_2, \ldots Count_n$	$Count_i$
	Broadcast	$Count_0$	$Count_0$
	Multicast	$GCount_1, GCount_2, \ldots GCount_m$	$GCount_x, GCount_y, \ldots (x, y \in [1, m])$
Additional values	Unicast	$C_1, C_2, \ldots, C_n$	C_i
	Broadcast	C_0	C_0
	Multicast	$GC_1, GC_2, \ldots, GC_m$	$GC_x, GC_y, \ldots (x, y \in [1, m])$
Session key	Unicast	$sk_1, sk_2, \ldots, sk_n$	sk_i
	Broadcast	sk_0	sk_0
	Multicast	$gsk_1, gsk_2, \ldots, gsk_m$	$gsk_x, gsk_y, \ldots (x, y \in [1, m])$

Table 7. The information that needs to be saved in MS and SXs.

	Communication Ends	**MS (u_0)**	**SX ($u_i, i \neq 0$)**
counters	Number of keys	$< 2n + 2m + 2$	$4 + m$, $4 + 2m$
	Number of Counters	$n + m + 1$	$2, 2 + m$
	Number of Additional values	$n + m + 1$	$2, 2 + m$

The length of the key used in symmetric cryptography methods typically ranges between 128 and 256 bytes in real-world applications (such as AES and IDEA). In this particular piece of research, we chose a key that was 128 bits long, with a counter that was the same length and an extra value. For MS, the storage of keys and the data associated with them can be managed by specialized key management servers. There is no issue with the expense of storage.

In contrast, SXs have a restricted capacity for storing data. As a result, it is necessary to determine the storage cost that is the absolute maximum and achievable for each SX in accordance with the number of SX and the number of DR projects. Table 8, which is derived from Table 6, summarizes the storage cost according to the number of DR projects and SXs.

Table 8. Storage cost according to number of DR project and SXs (1000–10,000).

		Storage Cost (Kbytes) (SXs = 1000 to 10,000)
Number of DR project	$m = 5$	0.448
	$m = 10$	0.768
	$m = 15$	1.088

Figure 11 shows the relation between the number of DR projects and the storage cost in Kbytes. From this figure, it is shown that when the number of DR projects increases, the storage cost also increases.

As a result of this, we have determined that the storage cost incurred by each SX inside the AMI system would not rise as the number of SXs increased. Only an increase in the total number of DR projects would result in an increase in the storage cost for each SX. We assumed that the number of DR projects did not exceed fifteen and that each SX's maximum related storage cost was 1.088 KB when everything operated as it should. The outcome might be considered satisfactory.

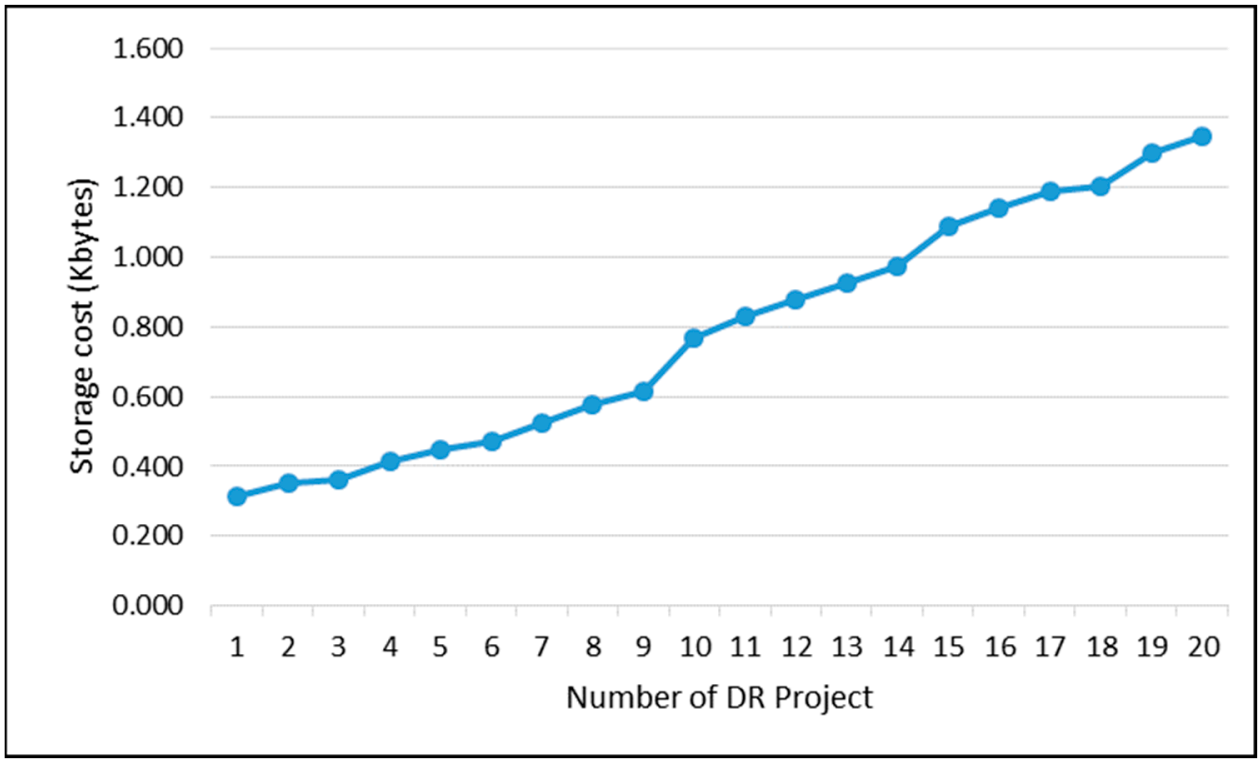

Figure 11. Storage cost according to number of DR projects and SXs (1000–10,000).

9.1. The Time Required for Conducting Computations

Because the transmission of messages has a time constraint, it is necessary to investigate the amount of time required to complete the maximum number of calculation jobs within a certain period of time. In accordance with the procedures of the key management, the following is an overview of the calculation technique for computing time costs for each of the three ways of transmission, which can be found in Table 9.

Table 9. Computation time cost for three transmission ways.

Transmission Way	MS (u_0)	SX ($u_i, i \neq 0$)
Unicast	$\leq (C_{HMAC} + 2C_H + 3C_{XOR}).n$	$\leq (C_{HMAC} + 2C_H + 3C_{XOR})$
Broadcast	$(2C_H + C_{XOR})$	$(2C_H + C_{XOR})$
Multicast	$\leq (C_R + C_{Kgen}).N_P + (C_{HMAC} + 2C_H + 3C_{XOR}).N_G + (2C_H + C_{XOR}).m$	$\leq (C_{HMAC} + 2C_H + 3C_{XOR}) + (2C_H + C_{XOR})$

(1) C_{HMAC}: Time cost of executing an HMAC operation.
(2) C_H: Time cost of executing a HASH operation.
(3) C_{XOH}: Time cost of an exclusive OR operation.
(4) C_R: Time cost of a random-number generation.
(5) C_{Kgen}: Time cost of b-bit key generation algorithm.
(6) *n: number of SXs.*
(7) *m: number of DR projects*
(8) NP: represents the count of DR projects that experience user joining or leaving, with a maximum limit of m
(9) NG: represents the total number of users in DR projects that undergo user joining or leaving, with a maximum limit of n.

Embedded systems are always used to conduct the implementation of the SX. To perform computations involving cryptography, embedded cipher chips are often used. Hash functions, symmetric cryptographic methods, and HMAC all have an approximate operating rate in the range of 10–50 Mb/s. An XOR operation has an extremely low throughput, making it impossible to take it into account.

The time investment required for computing in each SX can be tallied. Table 10 presents an overview of the results. According to the results, the amount of time consumed on the calculation in each SX was extremely limited for SXs that did not have an effect on the transmission of various messages.

Table 10. Time consumed in each SX for three transmission way.

	Unicast	**Broadcast**	**Multicast**
Time cost of computation in each SX (μs)	7.68–38.4	5.12–25.6	12.8–38.4

The PCI cryptographic coprocessor could be used to help with the calculation in MS. Symmetric cryptographic techniques, hash functions, and HMAC operate at a pace of between 50 Mb/s and 1 Gb/s, while the rate of random number generation was 1 Gb/s. The speed of an XOR operation need not be taken into consideration. Table 11 contains the results of calculations that were performed to calculate the computation time cost for broadcast and unicast modes. The results indicate that the time cost was incredibly low and practically had no effect on the numerous messages being transmitted because it was so minutely high.

Table 11. Time consumed in MS for unicast and broadcast.

	Unicast	**Broadcast**
Time cost of computation in MS (μs)	0.384–7.68	0.256–5.12

The value of NP and NG should be considered while determining the time cost for the multicast mode. The results of this calculation can be seen in Table 12. The amount of time consumed increases according to the value of the NG. However, the time cost does not have any effect on the transmission of messages, even if NG is set to a value of ten thousand. The value of Np changed from 5 to 15, and the time consumed did not have any effect.

Table 12. Time consumed in MS for multicast with NG and NP consideration.

N_G		**1000**	**2000**	**3000**	**5000**	**10,000**
Time cost of computation in MS (ms)	NP = 5	0.38–6.98	0.69–14.33	1.12–22.95	1.89–37.87	3.75–75.74
	N_P = 10	0.38–6.98	0.69–14.33	1.12–22.95	1.89–37.87	3.75–75.74
	N_P = 15	0.38–6.98	0.69–14.33	1.12–22.95	1.89–37.87	3.75–75.74

Figure 12 shows the relation between the total number of users in DR projects and the time cost of the computation. It is shown that when the number of users increased, the time cost of computation increased. It is also shown that when the number of projects changed from 5, 10, and 15, however, the time cost of computation did not change. Thus, the time cost of computations depended on the number of users, not the number of DR projects.

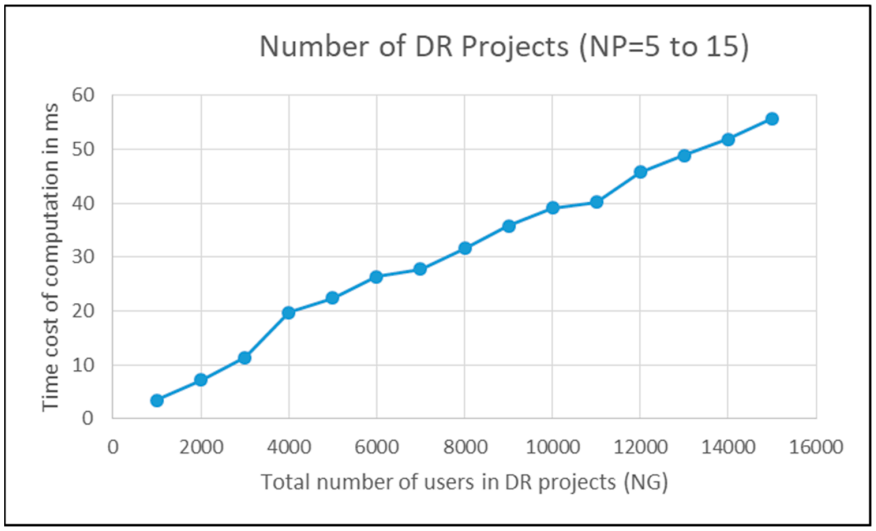

Figure 12. Time consumed in MS for multicast with NG and NP consideration.

9.2. Time Cost for Distribution

The time needed to distribute keys during a refreshing period is calculated based on the key management study by multiplying NG (the total number of users in DR projects with users joining or exiting) by CT (the time needed to distribute a package containing the key and related data). The distribution package size for the key and data typically did not exceed 384 bytes. Typically, the Synchronous Digital Hierarchy (SDH) network based on optical fibers was used to transmit data at a rate of 155 Mb/s, 622 Mb/s, or higher between the MS and AMI systems. Table 13 can be used to calculate the cost of this distribution. The results suggest that the distribution time does not affect the key refreshment process or network traffic distribution in AMI systems.

Table 13. Time cost for key and associated data distribution.

N_G	1000	2000	3000	5000	10,000
Distribution cost (ms)	2.48	4.96	7.44	12.39	24.77

Figure 13 shows the relationship between the total number of users in DR projects and the time cost of key distribution and associated data. It is shown that when the number of users increased, the time cost of the key distribution increased.

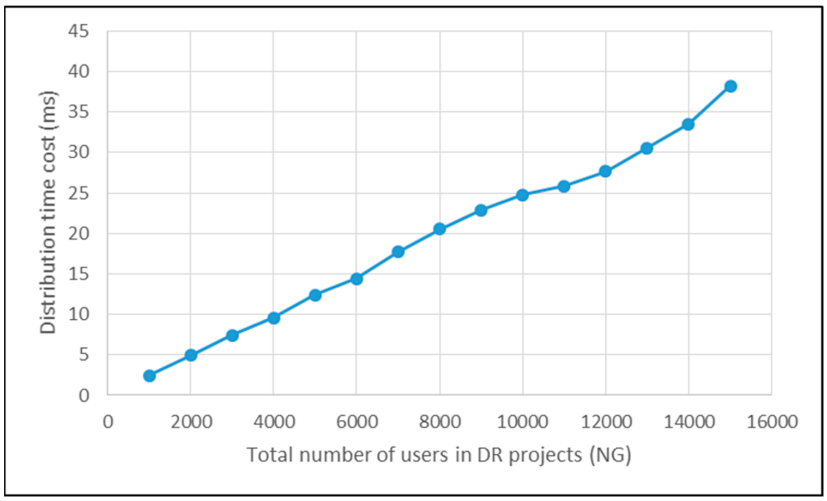

Figure 13. Distribution time consumed for the key and associated data with an NG consideration.

10. Conclusions

The AMI is one of the most important elements in the smart grid, and therefore, it was necessary to take care of the confidentiality of the transfer of information from it to monitoring and control centers. A new KMS that addresses common security concerns has been developed in order to overcome the primary problems with controlling and monitoring AMI systems. Three essential management techniques are supported by the KMS design for unicast, broadcast, and multicast modes of hybrid transmission.

The storing and computation of keys and associated data can be simply implemented in SMs or UGs based on our system's performance and security evaluations. Additionally, in an AMI system, the distribution of keys and the data they are connected with do not obstruct normal network activity.

After implementing the proposed strategy in our paper, the results were improved and indicated the stability of the AMI system and the security of the data. The storage cost was calculated according to the number of DR projects and SXs. The time required for conducting computations and the time cost for key distribution were calculated. The storage cost ranged from 0.448 to 1.4 Kbytes as the number of DR projects ranged from 5 to 20, which indicated that the storage cost increased when the DR increased. These values of storage cost are insignificant when compared with their improvement in storage devices.

The time required for conducting computations was calculated for three transmission ways and for each SX and MS. The proposed key management algorithm consumed about 5 to 38 microseconds at each SX and consumed 0.3 to 5 microseconds at MS with unicast and broadcast transmission ways. The time consumed in MS for multicast should take NG and NP into consideration.

The time cost for key distribution depends on the number of users in the DR project. Therefore, with 1000 users, the distribution time would be about 2.48 ms, and for 10,000 users, the distribution time would be about 24.77 ms.

In future work, the energy-efficiency could be studied, and research could be conducted on energy-efficient cryptographic algorithms to minimize their impact on performance and battery life. Additionally, Real-World Deployment and Testing could be applied to evaluate its effectiveness in actual AMI systems. This could help identify practical challenges and fine-tune the scheme accordingly.

Author Contributions: Conceptualization, A.A.A. and B.M.E.-d.; methodology, K.M.A.-A.-E., software, A.A.A. and B.M.E.-d., validation, T.M.H.; formal analysis, K.M.A.-A.-E.; investigation, A.A.A.; resources, T.M.H.; data curation, B.M.E.-d.; writing—original draft preparation, A.A.A., and K.M.A.-A.-E.; writing—review and editing, K.M.A.-A.-E., and T.M.H.; visualization, B.M.E.-d.; supervision, K.M.A.-A.-E.; project administration, A.A.A.; funding acquisition, K.M.A.-A.-E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Centre for Power Systems Research (CPSR), the Faculty of Engineering and the Built Environment, Cape Peninsula University of Technology, Cape Town, South Africa, and The APC was funded by Centre for Power Systems Research (CPSR), Faculty of Engineering and the Built Environment, Cape Peninsula University of Technology, Cape Town, South Africa.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Abbreviation	Meaning
n	The number of smart meters SX
m	The number of DR projects
u_i	The ith user in AMI. u_0 refers to MS; ot hers refer to SXs
k_i	A user key for u_i
gk_i	A group key for the ith DR project
sk_i	A session key is used for message transmissions, with $i = 0$ and the specifically for broadcast mode and other values for the unicast mode.
gsk_i	a session key for multicast mode of the ith DRproject
C_i	an additional value to help generate sk_i
GC_i	an additional value to help generate gsk_i
$Count_i$	a counter for u_i to help generate C_i
$GCount_i$	a counter for u_i to help generate GC_i
$Data$	Data of the message
$EData$	Encrypted data of the message.
M_i	Metered data of u_i in the previous day, which is in binary mode with a fixed length.
$CDate$	The measure date of M_i.
$Sign_t$	A signature of the encrypted data created in the sending end.
$Sign_r$	A signature of the encrypted data created in the receiving end.
$Project_j$	The jth DR project
$request_{in}j$	request information of joining in the jth DR
$request_{out}j$	request information of quitting in the jth DR
$response+$	Indicating that the distribution by MS succeeded.
$response-$	Indicating that the distribution by MS failed.
$Kgen(1^b)$	A secure $b-$ bit key generation algorithm.
$Random\ (b)$	A function to generate a $b-$ bit random number.
$E_k(Data)$	An encryption function that utilizes a symmetric cryptography algorithm and utilizes the encryption key "k"
$DE_k(EData)$	A decryption function that employs a symmetric cryptography algorithm and uses "k" as the decryption key.
$k \oplus c$	an exclusive OR operation between k and c.
$k\|\|c$	A concatenation between k and c.
$H(k)$	A hash function.
$HMAC_K(c)$	A keyed–hash message authentication function that employs as "k" the key.

References

1. Kumar, V.; Kumar, R.; Pandey, S.K. LKM-AMI: A lightweight key management scheme for secure two way communications between smart meters and HAN devices of AMI system in smart grid. *Peer Peer Netw. Appl.* **2021**, *14*, 82–100. [CrossRef]
2. Dileep, G. A survey on smart grid technologies and applications. *Renew. Energy* **2020**, *146*, 2589–2625. [CrossRef]
3. Hussain, S.M.S.; Ustun, T.S.; Kalam, A. A review of IEC 62351 security mechanisms for IEC 61850 message exchanges. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5643–5654. [CrossRef]
4. Panda, D.K.; Das, S. Smart grid architecture model for control, optimization and data analytics of future power networks with more renewable energy. *J. Clean. Prod.* **2021**, *301*, 126877. [CrossRef]

5. Hasan, M.K.; Alkhalifah, A.; Islam, S.; Babiker, N.B.; Habib, A.A.; Aman, A.H.M.; Hossain, M.A. Blockchain technology on smart grid, energy trading, and big data: Security issues, challenges, and recommendations. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9065768. [CrossRef]
6. Yaacoub, J.P.A.; Salman, O.; Noura, H.N.; Kaaniche, N.; Chehab, A.; Malli, M. Cyber-physical systems security: Limitations, issues and future trends. *Microprocess. Microsyst.* **2020**, *77*, 103201. [CrossRef] [PubMed]
7. Islam, N.; Rahman, M.S.; Mahmud, I.; Sifat, M.N.A.; Cho, Y.Z. A Blockchain-Enabled Distributed Advanced Metering Infrastructure Secure Communication (BC-AMI). *Appl. Sci.* **2022**, *12*, 7274. [CrossRef]
8. Abdalzaher, M.S.; Fouda, M.M.; Emran, A.; Fadlullah, Z.M.; Ibrahem, M.I. A Survey on Key Management and Authentication Approaches in Smart Metering Systems. *Energies* **2023**, *16*, 2355. [CrossRef]
9. Kim, J.; Ahn, S.; Kim, Y.; Lee, K.; Kim, S. Sensor network-based AMI network security. In Proceedings of the IEEE PES Transmission and Distribution Conference and Exposition: Smart Solutions Changing World, New Orleans, LA, USA, 19–22 April 2010; pp. 1–5.
10. Kabalci, Y. A survey on smart metering and smart grid communication. *Renew. Sustain. Energy Rev.* **2016**, *57*, 302–318. [CrossRef]
11. Avancini, D.B.; Rodrigues, J.J.; Martins, S.G.; Rabêlo, R.A.; Al-Muhtadi, J.; Solic, P. Energy meters evolution in smart grids: A review. *J. Clean. Prod.* **2019**, *217*, 702–715. [CrossRef]
12. Ghosal, A.; Conti, M. Key management systems for smart grid advanced metering infrastructure: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2831–2848. [CrossRef]
13. Ghosh, S.; Sampalli, S. A survey of security in SCADA networks: Current issues and future challenges. *IEEE Access* **2019**, *7*, 135812–135831. [CrossRef]
14. Liu, Y.; Gao, H.; Gao, W.; Peng, F. Development of a substation-area backup protective relay for smart substation. *IEEE Trans. Smart Grid* **2016**, *8*, 2544–2553. [CrossRef]
15. Latif, A.; Paul, M.; Das, D.C.; Hussain, S.S.; Ustun, T.S. Price based demand response for optimal frequency stabilization in ORC solar thermal based isolated hybrid microgrid under salp swarm technique. *Electronics* **2020**, *9*, 2209. [CrossRef]
16. Chakraborty, S.; Das, S.; Sidhu, T.; Siva, A.K. Smart meters for enhancing protection and monitoring functions in emerging distribution systems. *Int. J. Electr. Power Energy Syst.* **2021**, *127*, 106626. [CrossRef]
17. Wang, X.; Mao, X.; Khodaei, H. A multi-objective home energy management system based on internet of things and optimization algorithms. *J. Build. Eng.* **2021**, *33*, 101603. [CrossRef]
18. Rafique, Z.; Khalid, H.M.; Muyeen, S.M. Communication systems in distributed generation: A bibliographical review and frameworks. *IEEE Access* **2020**, *8*, 207226–207239. [CrossRef]
19. Liu, N.; Chen, J.; Zhu, L.; Zhang, J.; He, Y. A Key Management Scheme for Secure Communications of Advanced Metering Infrastructure in Smart Grid. *IEEE Trans. Ind. Electron.* **2013**, *60*, 4746–4756. [CrossRef]
20. Benmalek, M.; Challal, Y.; Derhab, A. An improved key graph based key management scheme for smart grid AMI systems. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; pp. 1–6.