Chapter 4
A Novel AckIBE-Based Secure Cloud Data Management Framework

Dharavath Ramesh and Syam Kumar Pasupuleti

Abstract A smart grid of cloud includes various operations and other measures like smart meters, smart appliances, and renewable energy efficiency resources. The primary issues of this grid are how to manage various kinds of front-end devices such as smart meters and power assets efficiently and also, to efficiently process an enormous amount of data of participating devices. Since the cloud environment possesses various properties like scalability, cost saving, energy saving, and flexibility, it can serve as an efficient entrant to face these issues and challenges. This chapter introduces a more secure smart cloud computing framework-based AckIBE for data management, which we term as "*Smart-Model.*" The aim is to construct a hierarchical structure of homogeneous and heterogeneous cloud centers that delivers various types of computing services to support big data analysis and information management. Furthermore, we introduce a security-related solution based on acknowledgment identity-based encryption (*AckIBE*), signature and proxy re-encryption to face critical security issues of the proposed framework. Additionally, we introduce acknowledgments sent by the end-user to the provider to ensure that the data have been received by the end-user and not lost in the environment of cloud communication.

Keywords Cloud data · Secure management · Smart model · Identity-based encryption

D. Ramesh (✉)
Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, Jharkhand, India
e-mail: ramesh.d.in@ieee.org

S. K. Pasupuleti
Department of Computer Science and Engineering, Institute for Development and Research in Banking Technology, Hyderabad, India
e-mail: psyamkumar@idrbt.ac.in

B. B. Gupta et al. (eds.), *Handbook of Computer Networks and Cyber Security*,
https://doi.org/10.1007/978-3-030-22277-2_4

1 Introduction

While comparing traditional power grids with smart grids, it is found that smart grid models ensure improvement in terms of reliability, substantiality, and efficiency of computing services [1]. Although smart grids provide various advantages to electrical-related grids, their inclusion and accuracy is limited to smaller locations. Various challenges and issues recommend that smart grids to be deployed in larger capacities. Information management is concerned with the management of gathering, storing, and processing the information [2–4]. At the same time, there is needed to handle and manage large quantities of data that contain the selection, deployment, and inclusion of the data, monitoring the data, and analysis of the data of smart cloud models. Big data in the smart cloud models are created from several sources. These sources can be utilization activities of the users; phase-wise data used for storage and retrievals; data on energy consumption used by various smart location meters; management, maintenance, and control over the data. Other parameters also include network-related data acquired by operational devices such as servers and virtual machines, not directly obtained through the measurement but widely used in decision making.

The measurement of big data in terms of power utilities is increasing exponentially. By the year 2020, it is estimated that the number of smart operational meters of various cloud models of various continents will reach 650 million, whereas China is predicted to install about 450 million smart operational meters by that date. Smart grids usually require real-time processing, any delay in which may lead to a serious consequence in the whole system.

1.1 Support of Cloud Computing

Cloud computing has various advantages such as energy efficiency, scalability, flexibility, agility, and cost saving [5]. This has made it a significant model of computing in the near future. The cloud in the form of smart models addresses the issue of large-scale information and also responsible for a high energy and cost saving platform. This is due to (1) high scalability in order to deal with the amount of information being processed and (2) efficient utilization of resources in the corresponding data centers. Cloud environment also yields faster computation, efficient storage, and distributed computing facility to manage the big data. To process the potential of big data, there is a need to acquire new data analysis algorithms and approaches to manage the growth of enormous unexpected data. With the help of under managed cloud infrastructure, a service provider can provide better, cheaper, and more reliable cloud services to the consumers and end-users. Some of the related properties of operational grid and some cloud models in the form of smart-cloud models are analyzed to validate the relationship between them [6]. Motivated by the work presented in [27], in this paper, we propose a secure

Smart Cloud Framework (i.e., Smart Model) based AckIBE in the management of big data in homogeneous and heterogeneous cloud data centers. This chapter has threefold contributions:

- Introduction of smart-model: A framework based on cloud computing to perform information management of big data in the form of smart models that gives reasonable scalability as well as security.
- An identity-based encryption-based security solution is introduced for the proposed smart-model of IBE and identity-based proxy re-encryption to provide secure communication.
- We further introduce acknowledgments as AckIBE and show how messages along with signatures and the acknowledgments are sent in a hierarchical cloud environment from one level to another.

The rest of this chapter is organized as follows. Section 2 reviews the related work. Sections 3 and 4 present the proposed Smart-Model with possible security solutions. In a particular manner, Sect. 3 emphasizes the proposed methodology of Smart-Model architecture while Sect. 4 emphasizes related solutions based on AckIBE. Section 5 illustrates the security solutions and Sect. 6 presents the related schemes for secure framework. Finally, Sect. 7 demonstrates the security analysis followed by conclusions and future scope in Sect. 8.

2 Related Literature

2.1 Security Approaches of Smart Model

Smart model management of information generally consists of three main tasks, namely gathering, processing, and storing of information. For gathering of information, since smart models accumulate huge information from different kinds of devices located at different locations; several solutions have been introduced to address this challenge [2, 4, 10]. To manage the challenge of interoperability, a proposal to standardize the data structures is used in smart grids is proposed [11].

Since the deployment in smart grid is large, it suffers from several security vulnerabilities [24]. Authors of [12–15] introduced different methodologies to acquire the security challenges with respect to the processing of information of smart meters. Wei et al. [17] respectively proposed to protect smart model against cyber-attacks. Zhang et al. [16] proposed frameworks of security which are used in controlling the consistency of the security requirements of all the components of smart model. An authentication approach using digital signatures and time stamps is proposed by Rogers et al. [18]. As discussed in [8, 19, 20], identity-based cryptography is considered as a good candidate for secure cloud computing.

Authors of [21, 22, 28–30] introduced various security architectures for efficient cloud data storage. A methodology proposed in [25] discusses identity-based

signature (IB-S) schemes in the non-hierarchical-based environment of cloud. The work proposed in [26] constructs an agreement protocol named IB-key in the environment of general grid computation whereas the proposed work provides security based on IB-encryption/signature and IB proxy re-encryption schemes to the proposed model.

3 Basic IB Schemes

There are two different blocks in cryptography for the security of the Smart-Model, namely identity-based encryption (IB-E) and identity-based signature schemes (IB-S) which are available. Li et al. [25] proposed identity-based cryptography to remove the need to check whether the certificates are valid in the traditional public_key scenario. In the scheme of IB-E, the generator of the key named private_key (PKG) with a reliable party firstly produces secret key called master_key (mk) and a related parameter known as params.

The private keys are distributed in the form of digital certificates which are issued in normal public key schemes. The PKG authenticates users and then sends them the private keys with respect to their identities. Any sender who possesses IDrec enciphers an original plain-text PT(M) into a ciphertext C by executing the Encrypt algorithm. When ciphertext C is obtained, the receiver deciphers C by executing the decryption algorithm taking input as the KIDrec, the private key received from the party PKG.

Similarly, the description of an identity-based signature scheme [8] is proposed as follows. As soon as the signer provides user identity IDsig, the party computes the private_key as KIDsig with respect to the IDsig by executing the extraction algorithm taking input as the secret master key mk. By executing the sign algorithm, the signer signs with PT(M) to obtain a corresponding signature using KIDsig. Both the IB-E and IB-S does not use digital certificates, but provide certification for the each user. The user, who registered his/her identity and received his/her private key can only decrypt using the decryption procedure or create a valid signature. The signature scheme IB-S had already been proposed by Shamir [8], but the practical realization of IB-E was achieved in [7]. Hierarchical identity-based cryptography is the extension of identity-based cryptography [23] in such a manner that the root PKG delegates private key generation and identity authentication to other users that act as lower-level PKGs.

3.1 Other IB Schemes

The process of proxy re-encryption makes a proxy to change the ciphertext created using the public key of Alice in such a way that the changed ciphertext can be deciphered using the private key of another party Bob. Ateniese et al. [20] introduced

the first fully functioning proxy re-encryption scheme. After Ateniese et al.'s work, numerous proxy re-encryption schemes with different functionalities have been introduced. Ramesh et al. proposed an e-Stream-based secure dynamic updation policy for secure cloud storage. In this, the authors examined a stream cipher called ChaCha20 to provide the security for efficient data storage dynamically [21]. Xiaming Hu et al. proposed Secure and Efficient Identity-Based Proxy Signature Scheme in the Standard Model Based on Computational Diffie–Hellman Problem on proxy signature scheme [22].

4 Secure Smart Model

In this section, we illustrate the system construction with its architecture, component views, and flow of information management. Smart-Model denotes a framework that provides scalable, flexible, and secured transformation of data designed for smart-models and uses cloud computing technology. Here, we have adopted an idea to construct the model in three different layers of hierarchy as: Top-Cloud, Regional-Cloud, and End-user levels. The first and second level consists of cloud computing centers whereas the last level consists of end-user intelligent devices. The cloud at the topmost level takes the charge of managing and handling the participated devices and collection of data at various regional cloud centers. On the other side, the regional cloud computing devices handle lower hierarchical level located front-end intelligent devices, which are at a level lower than the computing entities (centers) of regional cloud (i.e., Homogeneous region) with the data transmitted from participating devices. Since smart grids are sensitive and needed strict protection, information leakage of any kind should be prevented in smart grids else it may lead to fatal consequences. In this framework, we further introduce a security solution in the form of IB-Encryption, Signature, and IB proxy re-encryption schemes [7–9]. The advantage of using identity-based encryption over traditional public key encryption scheme is that the former uses identities instead of digital certificate which depend upon public key infrastructure.

This saves the resource utilization for performing computation and resolve scalability problems. Also, in order to ensure that data reach the destined receiver and not get lost in the large cloud environment, we introduce acknowledgment to be sent by receivers to the senders. These acknowledgments are also sent in an encrypted form on receiving the acknowledgment the senders decrypt it and gets the concerned information. The architecture used is drawn in Fig. 4.1.

4.1 Smart-Model: System Architecture

In this section, we brief about the proposed architecture. The overview of the proposed model is shown in Fig. 4.2. This model includes a constructed grid, which

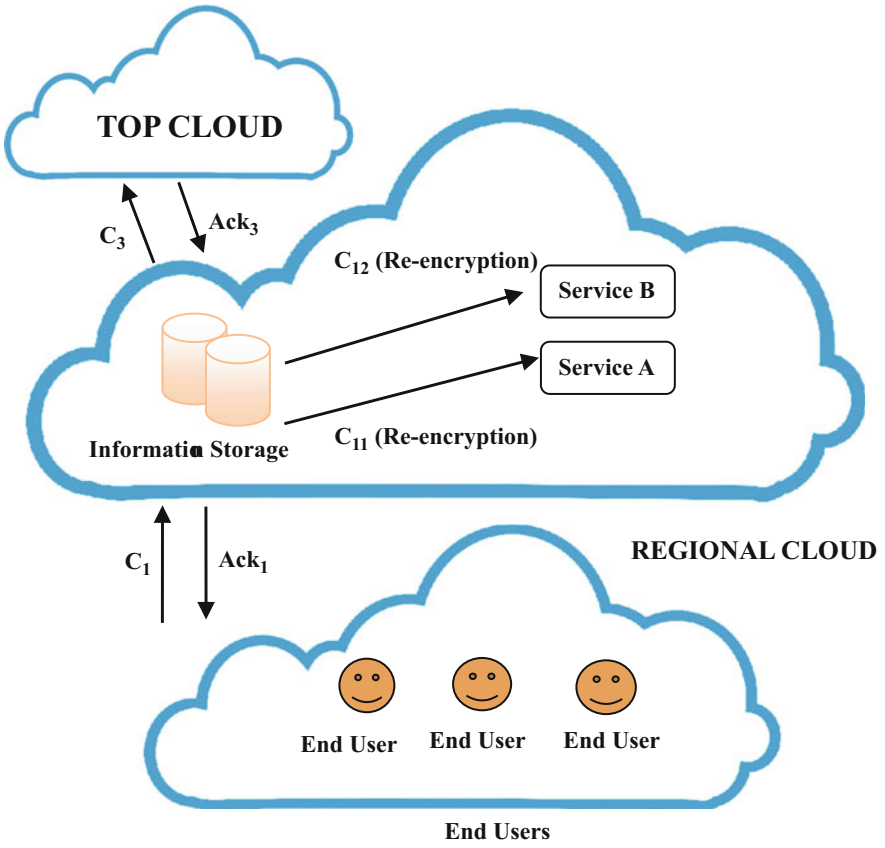

Fig. 4.1 Architecture used in the system

is partitioned into segmented regions. A cloud computing center handles these regions. The computing center is arranged and managed either by public or private cloud offerings. The basic functionality of any regional center is to handle end-user devices which are situated in the same locality (i.e., corresponding region) and also to give a primary level of processing of data that comes from the participated active devices. The main computing center at the top level is responsible to manage and process the suitable information data for the participated grids. And also, the center is responsible for the deployment of the following services that fall under cloud computing.

Infrastructure-as-a-Service (IaaS) This service is provided with on-demand basis, which makes resources available to all the applications and services deployed. The basic functionalities of management in the proposed model such as collection of data, processing, and storage are managed under this service.

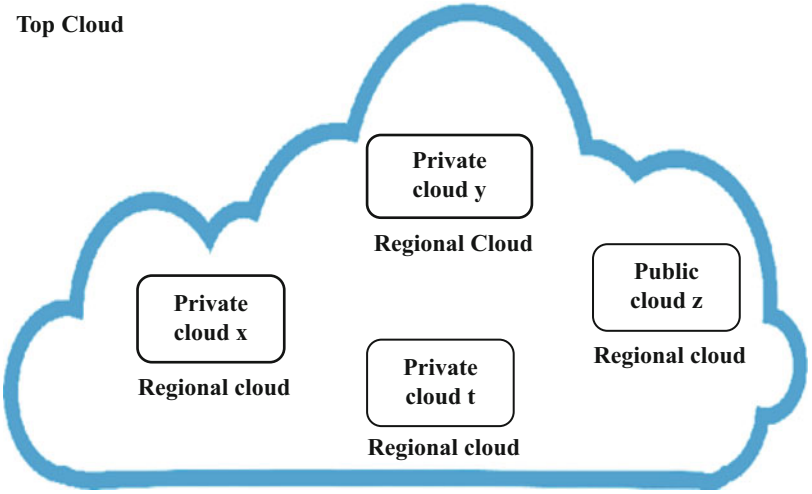

Fig. 4.2 Architecture of the smart frame

Software-as-a-Service (SaaS) This service deploys the required services of a smart model at the top of the system. For example, required services that enable customers to save and optimize their usage of energy [20], e.g., GPM.

Platform-as-a-Service (PaaS) This service offers different tools and library functions responsible for the development of cloud computing services and applications. Since there are numerous applications which are required to support various security offerings to permit legal interceptions, it is convenient to have platform-as-a-service that has these inbuilt requirements for the implementation of the applications.

Data-as-a-Service (DaaS) For providing relevant information for statistics purpose, DaaS can be deployed. Smart grid data are usually enormously large in amount. It serves beneficial to provide such statistics services for service users.

4.2 Component Views

In this framework, we propose four basic functional clusters as follows. These types of services are illustrated in Fig. 4.3.

Information Storage All the information on smart grid collected from front-end intelligent devices like smart meters, etc., are kept in main storage, which are developed to get information from various modes of transportation with the help of wired channel as well as wireless channel. The related statistics exist in the corresponding cluster.

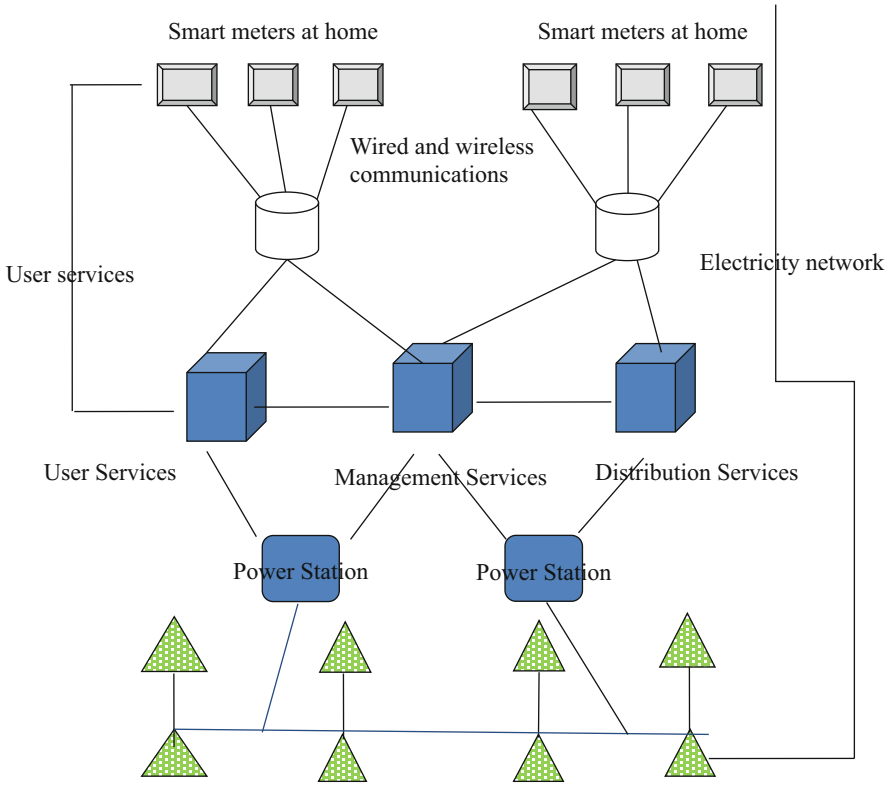

Fig. 4.3 Functionality of cloud service clusters

User Services All the services that an electricity consumer uses fall under this service. The examples include monitoring, controlling, and optimizing the use of their electric utilization. This sort of service includes most of the SaaS and also PaaS that provides libraries for user services.

Control and Management Services All services with respect to system management like governance service, monitor, task scheduling, and security fall under this category.

Electricity Distribution Services The services related to electricity distribution fall under this category. Examples include optimization service, measuring quality of service measurement, services pertaining to distribution.

4.3 Flow of Information Management

As smart grids are supposed to handle the enormous amount of data, it is challenging to efficiently manage the information flows in the system. In our proposed Smart-Model, a centralized service is suggested to manage the flow of information. The required inputs are taken from the clusters which are in service and other statistics such as size of the data and the time at which the data are entered into the cluster. Taking these inputs, the service creates a basic schedule of information flow. The schedule gives the description of the beginning and end of information flows and also how their processing is done (i.e., type of operations used on the flows with their locations). For execution, participated centers with their corresponding clusters need to go through the schedules.

It is important to notice that since the amount of information and related requests in the model may vary time by time, every flow has got an elapsed time. Once this elapsed time expires, a new schedule has to be inclined and sent to the participating centers again. A related smart model flow is shown in Fig. 4.4.

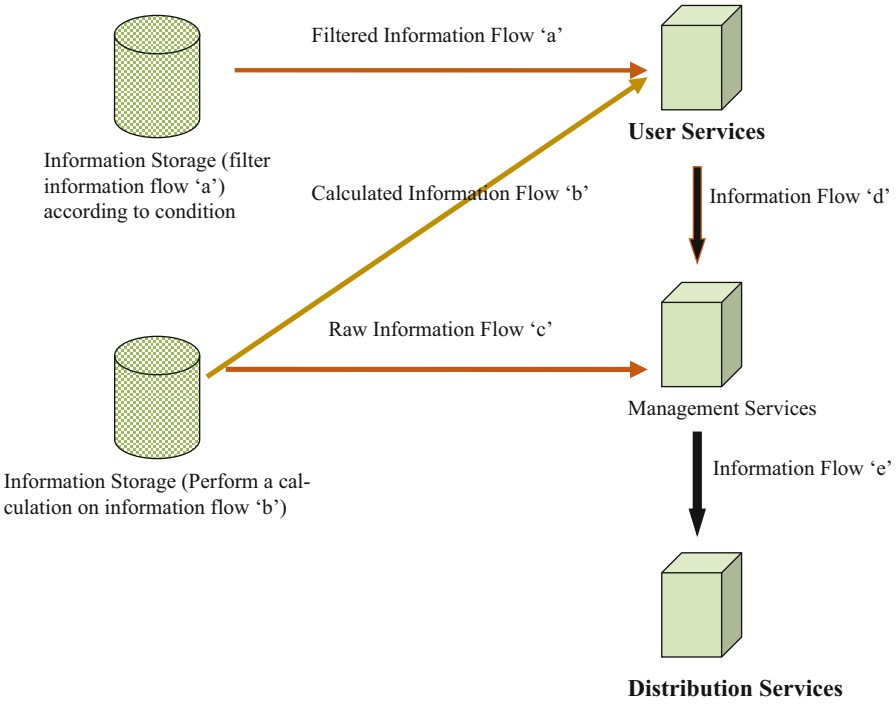

Fig. 4.4 Flow of information schedule

5 Security Solutions for Smart Model

5.1 Model Description

In this section, we assume the following parameters while realizing the security framework. The working of this framework is illustrated in Fig. 4.5 in the form of proxy re-encryption methodology.

- There exists a generator of private PKG that issues private keys for the entities participating in the hierarchy whenever they register. It is assumed that the party PKG is responsible and possesses the capacity for maintaining the Smart-Model generally at different levels with reliable credentials.
- Unique strings are used as IDs to identify the existing cloud at the top level and end-users assigned. These are used as either to encrypt the original message or to verify the signature.
- Every participating entity receives its related private key based on the identity which can decipher the ciphertext that includes the confidential data.
- Every participating entity sends an encrypted data to the entity that is participating its peer level. So, the end-user can send the data to the regional cloud entities. Similarly, the entities present in the regional cloud are able to send encrypted data to the cloud existing at a higher level.
- Every participating entity authenticates shared data through its private key received from PKG.
- Every level, which receives the data, can send the acknowledgment to the sender.

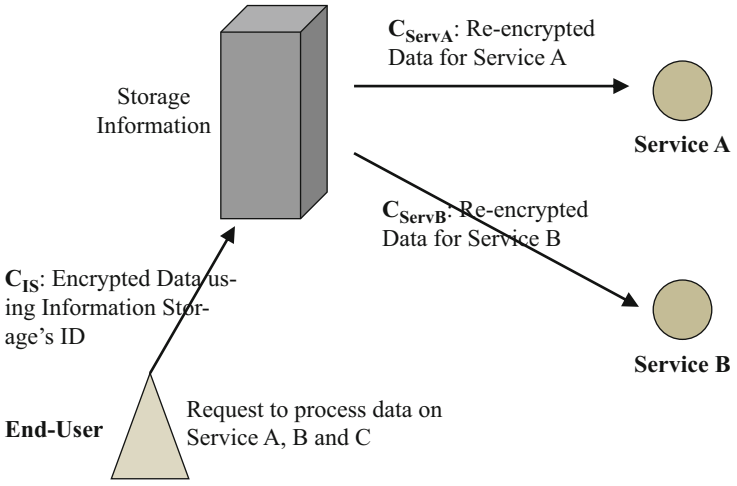

Fig. 4.5 Proxy re-encryption

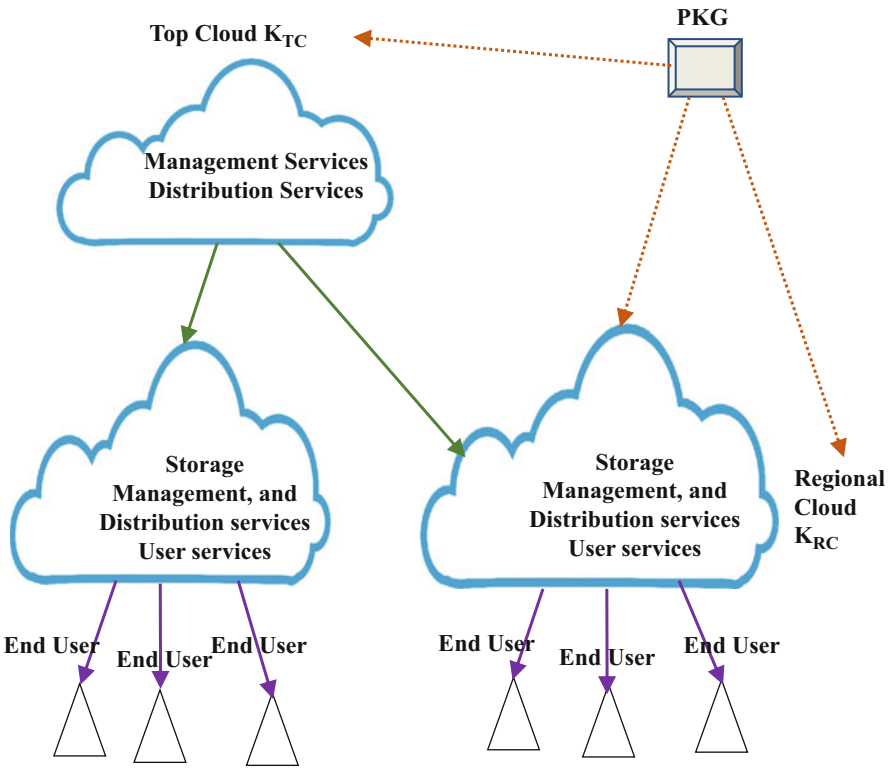

Fig. 4.6 Hierarchical architecture

Based on the above assumptions, we construct the architecture as depicted in Fig. 4.6. The hierarchy of the top cloud contains distribution services, management services, and power stations. The top cloud manages the regional clouds. These regional clouds contain basic user services and storages of information. Below regional clouds, there exists a lower hierarchy of smart (intelligent) end-user devices.

5.2 Key Generation

Setup With the help of a parameter γ, the *PKG* produces a secret key, also known as master key *mkey* and a parameters' set *params*. This *params* is distributed to end-users and all the clouds.

Extract_TCKey: After getting the identity TC of top cloud, the party PKG produces a private key K_{TC}, in correspondence with the identity TC by executing *Extract()*, the extraction algorithm of the private key in which TC is taken as input. It is represented as:

$$K_{\mathrm{TC}} \leftarrow \text{Extract\_TCKey}\,(params, mkey, TC)$$

Extract_ISKey: After getting a IS as Information Storage's identity, the PKG produces a private key K_{IS}, with the identity of IS by executing the extraction algorithm of private key *Extract()* in which IS is taken as input. It is represented as:

$$K_{\mathrm{IS}} \leftarrow \text{Extract\_ISKey}\,(params, mkey, IS)\,.$$

Extract_ServiceKey: After getting Service A's identity *ServA* in the regional cloud, the PKG produces a key K_{ServA} as private key in correspondence with the identity *ServA* by executing *Extract()*, the extraction algorithm of the private key in which *ServA* is taken as input. It is represented as:

$$K_{ServA} \leftarrow \text{Extract\_ServiceKey}\,(params, mkey, ServA)\,.$$

Extract_EUKey: After getting the identity EU of top cloud, the party PKG produces a private key K_{EU} in correspondence with the identity EU by executing *Extract()*, the extraction algorithm of private key in which EU is taken as input. It is represented as:

$$K_{\mathrm{EU}} \leftarrow \text{Extract\_EUKey}\,(params, mkey, EU)\,.$$

5.2.1 Encryption to Top Cloud

a) *Encrypt_to_TC*: Any information storage can encipher M, an original message into a ciphertext $\mathrm{C_{TC}}$ by executing *Encrypt(),* the IBE encryption algorithm taking input as Information Storage's identity TC and params. We represent the encryption as follows:

$$\mathrm{C_{IS}} \leftarrow \text{Encrypt\_to\_TC}\ (params, TC, M)\,.$$

b) *Decrypt_TC*: The top cloud deciphers the obtained *C* (Ciphertext) to deciphered message *M* by executing *Decrypt IBE decryption algorithm* with the key K_{TC} generated in correspondence with the information storage's identity *TC*. The decryption is presented as follows:

$$\mathrm{M} \leftarrow \text{Decrypt\_TC}\,(\text{params},\ K_{\mathrm{TC}},\ \mathrm{C_{TC}})\,.$$

5.2.2 Encryption to IS

(a) *Encrypt_to_IS*: Any end-user can encipher *M* into a ciphertext C_{IS} by executing *Encrypt()*, the *IBE encryption algorithm* taking input as identity *IS* of Information Storage and params. We represent the encryption as follows:

$$\mathrm{C_{IS}} \leftarrow \text{Encrypt\_to\_IS } (params, IS, M)$$

(b) *Decrypt_IS*: Any regional cloud can decipher the obtained ciphertext *C* to *M* by executing IBE decryption algorithm with its private key $\mathcal{K}_{IS}$ in correspondence with its identity *IS*. This represents the decryption as follows:

$$\mathrm{M} \leftarrow \text{Decrypt\_IS}\left(params, \mathcal{K}_{IS,}\, C_{IS}\right)$$

5.2.3 Proxy Re-encryption

(a) *RencKGen*: The storage of regional cloud produces a re-encryption key $\mathrm{RencK_{IS}} \rightarrow_{\mathrm{ServA}}$ by taking input as $\mathcal{K}_{\mathrm{IS}}$, the self-private key, IS, ServA. This is represented as:

$$\mathrm{RencK_{IS}} \rightarrow {}_{\mathrm{ServA}} \leftarrow \text{RencKGen}\left(\mathcal{K}_{\mathrm{IS,}}\ \mathrm{IS, ServA}\right)$$

(b) *Re_encrypt*: The ciphertext $\mathrm{C_{IS}}$ is re-encrypted with the help of the re-encryption key $\mathrm{RencK_{IS}} \rightarrow_{\mathrm{ServA}}$ and receives a ciphertext $\mathrm{C_{ServA}}$. This process is represented by CServA ← Re_encrypt ($\mathrm{RenK_{IS\text{->}ServA}}$, $\mathrm{C_{IS}}$).

(c) *Decrypt_Service*: The service A deciphered ciphertext $\mathrm{C_{ServA}}$ with the help of its private key $\mathcal{K}_{\mathrm{ServA}}$. It is represented as M ← Decrypt_Service($\mathcal{K}_{\mathrm{ServA,}}$ $\mathrm{C_{ServA}}$).

5.2.4 Signature Generation by Top Cloud

(a) *Sign_TC*: Any user at end level is able to produce a signature δ for the original message (M) with the help of the private key $\mathcal{K}_{\mathrm{IS}}$ with respect to its identity TC. This is represented as follows: $\delta \leftarrow$ Sign_TC(params, $\mathcal{K}_{\mathrm{IS,}}$ M).

(b) *Verify_TC*: Verification of the signature δ of message *M* with the help of identity of the end-user and parameter *params*. This is represented by w ←Verify_TC(params, IS, δ,M). The result *w* denotes "*acceptance*" or "*rejection.*" Verification of the signatures produced by a service in a regional cloud is done in a similar manner.

5.2.5 Signature Generation in Regional Cloud

(a) *Sign_IS*: End-user produces a signature δ for the original message M with the help of the key generated as $\mathcal{K}_{IS}$ with respect to its identity IS. This is represented as follows:

$$\delta \leftarrow \text{Sign\_IS}\left(params, \mathcal{K}_{IS,}\, M\right).$$

(b) *Verify_IS*: Verification of the signature δ of message M with the help of identity of the end-user and parameter params. This is represented by d ←Verify_IS(params, IS, δ,M). Verification of the signatures produced by a service is done in a similar manner.

5.2.6 Signature Generation by End-Users

(a) *Sign_EU*: End-user produces a signature δ for the original message M with the help of the key $\mathcal{K}_{ServA}$ with respect to its identity EU. This is represented as follows:

$$\delta \leftarrow \text{Sign\_EU}\left(\text{params}, \mathcal{K}_{\text{EU,}}\, \text{M}\right).$$

(b) *Verify_EU*: Verification of the signature δ of message M with the help of identity of the end-user and parameter *params*. This is represented by d ←Verify_EU(params, EU, δ,M). The result d is either "accept" or "reject."

Acknowledgment by the Regional Cloud Any level, whether topmost cloud, regional cloud or the end-user can send the acknowledgment to any sender level. It is also sent in an encrypted form so that the recipient can decrypt it. The same encryption procedure is used.

6 Schemes for Secure Framework

The framework discussed below uses an IBE scheme [6] and identity-based proxy re-encryption scheme [1]. Both the schemes use a bilinear pairing *e: G X G -> G_T*. Here, the groups G and G_T are of prime order, which has the following properties:

- Bilinear: ∀r, s ∈ Z_p^*, e(g^r,h^s) = $e(g,h)^{rs}$.
- Non-degeneracy: It follows: $e(g,h) \neq 1$.
- Practically, e must be computable.

6.1 Confidentiality

The below-mentioned parameters propagate the knowledge of generating keys.

- *Key_Setup*: The group *PKG* produces G and G_T of order p as a prime and an admissible pairing $e: G \times G \rightarrow G_T$, a generator $g \in G$ and a hash function $H_1: \{0,1\}_* \rightarrow G$ and $H_2: G_T \rightarrow \{0,1\}^n$ for a positive integer n where n is the size of the plaintext. We then take random a where $a \in g^u$. The top cloud sets secret master key $mkey = u$ and a set of public parameters $params = (G, G_T, e, g, a, H_1, H_2)$. The parameter *params* is distributed to top, regional, and end-users by *PKG*.
- *Extract_TC_Key*: After getting the identity of top cloud TC, the PKG calculates $H1(TC)^u \in G$ and returns the private key $K_{TC} = H1(TC)^u$.
- *Extract_IS_Key*: After getting the top cloud's identity IS, the PKG calculates $H1(IS)^u \in G$ and returns the private key $K_{IS} = H1(TC)^u$.
- *Extract_Service_Key*: After getting the top cloud's identity ServA, the PKG calculates $H1(ServA)^u \in G$ and returns the private key $K_{ServA} = H1(ServA)^u$.
- *Extract_User_Key*: After getting a user's identity EU, the PKG calculates $H1(EU)^u \in G$ and returns the private key $K_{EU} = H1(EU)^u$.

6.1.1 Encryption to Top Cloud

- *Extract_to_TC*: A regional cloud entity can encipher an original message M with the help of *params* parameter and the identity TC of top cloud using following calculations. Take random value v where, $v \in Z_p$. Calculate $C1 = g^v$ and $C2 = M.\ e(a, H1(TC))^u$. Later, we get output ciphertext as $C_{TC} = (C1, C2)$.
- *Decrypt_TC*: With the help of private key $K_{TC} = H1(TC)^v$, the top cloud can decrypt a received ciphertext $C_{TC} = (C1, C2)$ into M, where $M = C2/(e(C1, K_{TC}))$.

6.1.2 Encryption to Information Storage

- *Extract_to_IS*: Any regional cloud entity can encipher an original message M with the help of the top cloud's identity TC and parameter params and using following calculations. Take random value v where, $v \in Z_p$. Calculate $C1 = g^v$ and $C2 = M.\ e(a, H1(TC))^v$. Later, we get output ciphertext as $C_{IS} = (C1, C2)$.
- *Decrypt_IS*: With the private key $K_{IS} = H1(IS)^v$, the top cloud can decrypt a received ciphertext $C_{IS} = (C1, C2)$ into M, where $M = C2/(e(C1, K_{IS}))$.

6.1.3 Proxy Re-encryption to Information Storage

- *RenKGen*: A Re-encryption key is received by an information storage possessing identity as IS by calculating $RenK_{IS->ServA} = (RK_1, RK_2, RK_3)$. Here we

compute $RK_1 = g^x$ and $RK_2 = L.e(a, H_1(ServA))^x$ and $RK_3 = K_{IS}^{-1}.H_2(T)$. We take random $x \in Z_p$. and $L \in G_T$.

- *Re_encrypt*: We have the Re-encryption key $RenK_{IS->ServA} = (RK_1, RK_2, RK_3)$. The ciphertext $C_{IS} = (C1,C2)$ is re-encrypted by service A and a new ciphertext is calculated as $C_{ServA} = (C1,C2,e(C1,RK3),RK1,RK2)$.
- *Decrypt_Service*: Let $C_{servA} = (C_1', C_2', RK_1', RK_2') = (C1, C2.e(C1,RK_3), RK_1, RK_2)$.

Since we have $K_{servA} = H1(ServA)^u$, the service A calculates $L = RK_2' / e(K_{servA}, RK_1')$. Later, we calculate $M = C_2' / e(C_1', H_2(L))$.

6.2 Authentication Service

Following is the description of the IBS scheme that makes use of IBS scheme Gentry and Silverberg has drawn from bilinear pairings.

Key Generation Another hash function represented as $H_2 : \{0,1\}^* \rightarrow G$ will be used in the signature generation. We have a master key of PKG as u_0 and a public parameter's set params = (G, GT, e, g0, b, H1, H2). Here, we take $b = g_0^{u0}$ as random. We have similar computations of extraction of key to regional cloud and top cloud as that of scheme of IBE.

6.2.1 Signature Generation by End-User Cloud

Sign_EU Every regional cloud computes a signature ∂ for the *M* with the help of its private key K_{TC} ($= \mathcal{K} = g_1^{u0}$). First, calculate $g_1 = H_1(EU) \in G$ and $g_M = H_1(EU,M) \in G$. Then choose w randomly as $w \in Z_p$, and calculate $\partial_1 = \mathcal{K}.\, g_M^w$ and $\partial_2 = g_0^w$. Later, we get signature $\partial = (\partial 1, \partial 2)$ as the output.

Verify_EU Any participating entity can perform verification of the signature ∂ for the message M with the help of the params parameters and EU, the identity of the top cloud. For verification a verifier checks whether $e(g_0, \partial_1) = e(b, g_1)\, e(\partial_2, g_M)$.

7 Security Analysis

The IBE scheme's correctness can be proven easily. The proof of the scheme is as follows. Let $C_{ServA} = (C_1', C_2', RK_1', RK_2') = (C_1, C_2, e(C_1,RK_3), RK_1, RK_2)$.

$$\begin{aligned} RK_2{}^{'}/e\left(K_{servA}, RK_1{}^{'}\right) &= RK_2{}^{'}/e\left(H_1(ServA)^u, RK_1\right) \\ &= L.e(u, H_1\,(ServA))^x/e\left(H_1(ServA)^u, RK_1\right) \\ &= L.e(g^u, H_1\,(ServA))^x/e\left(H_1(ServA)^u, RK_1\right) \\ &= L.e\left(H_1(ServA)^u, g^x\right)/e\left(H_1(ServA)^u, RK_1\right) \\ &= L. \end{aligned}$$

$$\begin{aligned} C_2{}^{'}/e\left(C_1^{'}, H_2\,(L)\right) &= C_2.e\,(C_1, RK_3)/e\,(C_1, H_2\,(L)) \\ &= C_2.e\left(C_1, K_{IS}{}^{-1}.H_2\,(L)\right)/e\,(C_1, H_2\,(L)) \\ &= C_2.e\left(C_1, K_{IS}{}^{-1}\right) \\ &= M.e(a, H_1\,(IS))^v.e\left(C_1, K_{IS}{}^{-1}\right) \\ &= M.e\left(g^v, H_1(IS)^u\right).e\left(C_1, K_{IS}{}^{-1}\right) \\ &= M. \end{aligned}$$

The validity of the verification algorithm of the signature scheme can be proved as:

$$\begin{aligned} e\;(g_0, \partial_1) &= e\;\left(g_0, \boldsymbol{K}.g_M^w\right) \\ &= e\,((g_0, K\;)\cdot e\left(g_0, g_M^w\right) \\ &= e\,(g_0, g_1{}^{u_0})\,e\left(g_0^w, g_M\right) \\ &= e\left(g_0^{u_0}, g_1\right)e\left(g_0^w, g_M\right) \\ &= e\,(b, g_1)\;\;e\,(\partial_2, g_M) \end{aligned}$$

7.1 Customized Platform

We provide a particularstate of the transition through the usage of the platform. We have participated entities as Top cloud, entities in the Regional Cloud and end-user. The scenario shows private key generation of the entities, Signature Generation and Encryption, Decryption and Signature Verification, Acknowledgment sent by a sender and received by the receiver. Let the confidential message be *"SM8||75KW||Kolkata."* The scenario is represented in Fig. 4.7, in the below manner.

8 Conclusions and Future Scope

This chapter introduces a secure framework (i.e., smart model) which is a general framework used for managing big data information in smart grids. The proposed framework is based on cloud computing technology and is formulated at three levels of hierarchy, i.e., top, regional, and end-user levels. The top cloud manages the

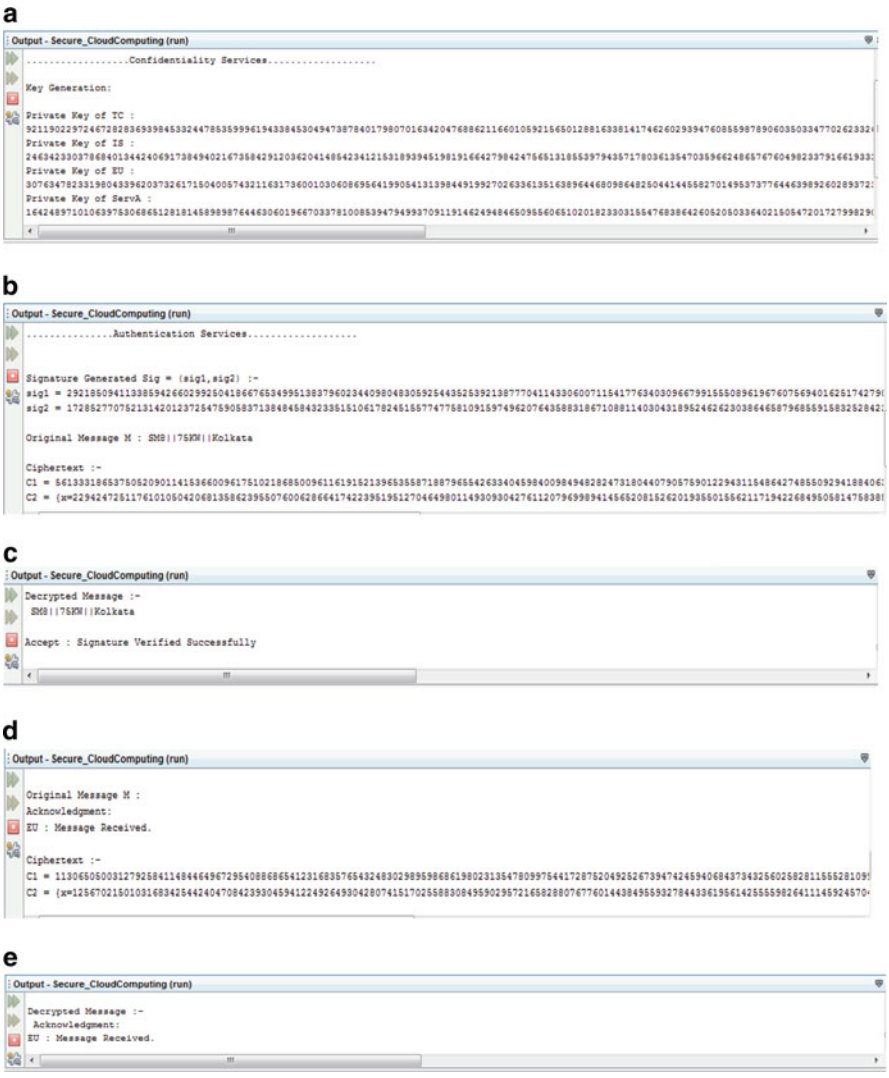

Fig. 4.7 Basic Operations of the model. (**a**) Registration of entities in Regional Cloud, Top Cloud, End-user, Service A. (**b**) Signature Generation and Encryption by the sender. (**c**) Decryption and Signature Verification by the receiver. (**d**) Acknowledgment Sent by the receiver to the sender. (**e**) Acknowledgment received by the sender. (**a**) *First step*: The entities of two clouds and end-user are registered and their private keys are generated. (**b**) *Second step*: In the second step: The meter of the smart_model uses the regional center identity to encipher its confidential message with respect to the daily consumption of electricity. Along with this, a signature is also generated based on IBS scheme and both the encrypted message (ciphertext) and the signature are sent to the Regional center (server). (**c**) *Third step*: The received message is decrypted by the regional center using its generated private key and also verified for authentication using a verification process of IBS scheme. (**d**) *Fourth step*: The regional center sends an encrypted acknowledgment to the sender (here smart meter). The encryption and decryption process is done by using the same IBE scheme. (**e**) *Fifth step:* The smart meter receives the acknowledgment by decrypting the received data

regional cloud whereas every regional cloud handles data got from various front-end intelligent devices. Since the cloud environment needs a security solution, two strategies named identity-based cryptography and identity-based proxy re-encryption have been provided. Thus, the proposed security framework is scalable, flexible, and secure. Additionally, we applied acknowledgment scheme so that the sender receives the feedback from the destined receiver to ensure that the data is not lost and has been delivered successfully. We have also described the architecture showing that how entities in regional cloud, top cloud, and end-user interact and transfer confidential data, signature, and acknowledgment within the system.

The efficiency of this framework can be further extended by using Identity-Based proxy signature scheme in the standard model based on the Computational Diffie Hellman Problem. This provides tight security reduction and more complete security, including resisting the delegator attack. It has more efficient performance and less computational cost than other similar existing schemes. Also, apart from this scheme, Identity-based Conditional Proxy Re-encryption can be used. This scheme is secure against the chosen ciphertext and identity attack in the random oracle model.

Acknowledgment This work is supported by the Indian Institute of Technology (ISM), Dhanbad, Govt. of India. The authors wish to express their gratitude and heartiest thanks to the Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, India for providing their research support.

References

1. Farhangi, H. (2010). The path of the smart grid. *IEEE Power and Energy Magazine, 8*(1), 18.
2. Bojkovic, Z., & Bakmaz, B. (2012). Smart grid communications architecture: a survey and challenges. In *Proceedings of the 11th International Conference on Applied Computer and Applied Computational Science* (*ACACOS*) (pp. 83–89).
3. McDaniel, P., & McLaughlin, S. (2009). Security and privacy challenges in the smart grid. *IEEE Security and Privacy, 7*(3), 75–77.
4. Fan, Z., Kulkarni, P., Gormus, S., Efthymiou, C., Kalogridis, G., Sooriyabandara, M., et al. (2013). Smart grid communications: overview of research challenges, solutions, and standardization activities. *IEEE Communications Surveys and Tutorials, 15*(1), 21–38.
5. Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
6. Rusitschka, S., Eger, K., & Gerdes, C. (2010). Smart grid data cloud: a model for utilizing cloud computing in the smart grid domain. In: *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on* (pp. 483–488). IEEE.
7. Boneh, D., & Franklin, M. (2001). Identity-based encryption from the Weil pairing. In *Annual International Cryptology Conference* (pp. 213–229). Berlin: Springer.
8. Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques* (pp. 47–53). Berlin: Springer.
9. Green, M., & Ateniese, G. (2007). Identity-based proxy re-encryption. In *Applied cryptography and network security* (pp. 288–306). Berlin: Springer.
10. Wang, W., Xu, Y., & Khanna, M. (2011). A survey on the communication architectures in smart grid. *Computer Networks, 55*(15), 3604–3629.
11. Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing, 13*(18), 1587–1611.

12. Efthymiou, C., & Kalogridis, G. (2010). Smart grid privacy via anonymization of smart metering data. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on* (pp. 238–243). IEEE.
13. Kalogridis, G., Efthymiou, C., Denic, S. Z., Lewis, T. A., & Cepeda, R. (2010). Privacy for smart meters: towards undetectable appliance load signatures. In: *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on* (pp. 232–237). IEEE.
14. Li, H., Mao, R., Lai, L., & Qiu, R. C. (2010). Compressed meter reading for delay-sensitive and secure load report in smart grid. In: *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on* (pp. 114–119). IEEE.
15. Chu, C. K., Liu, J. K., Wong, J. W., Zhao, Y., & Zhou, J. (2013). Privacy-preserving smart metering with regional statistics and personal enquiry services. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security* (pp. 369–380). ACM.
16. Zhang, T., Lin, W., Wang, Y., Deng, S., Shi, C., & Chen, L. (2010). The design of information security protection framework to support smart grid. In *Power System Technology (POWERCON), 2010 International Conference on* (pp. 1–5), IEEE.
17. Wei, D., Lu, Y., Jafari, M., Skare, P., & Rohde, K. (2010). An integrated security system of protecting smart grid against cyber-attacks. In *Innovative smart grid technologies (ISGT), 2010* (pp. 1–7). IEEE.
18. Rogers, K. M., Klump, R., Khurana, H., Aquino-Lugo, A. A., & Overbye, T. J. (2010). An authenticated control framework for distributed voltage support on the smart grid. *IEEE Transactions on Smart Grid, 1*(1), 40–47.
19. Liang, K., Liu, J. K., Wong, D. S., & Susilo, W. (2014). An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing. In *European symposium on research in computer security* (pp. 257–272). Cham: Springer.
20. Ateniese, G., Fu, K., Green, M., & Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC), 9*(1), 1–30.
21. Ramesh, D., Mishra, R., & Edla, D. R. (2017). Secure data storage in cloud: an e-stream cipher-based secure and dynamic updation policy. *Arabian Journal for Science and Engineering, 42*(2), 873–883.
22. Hu, X., Zhang, X., Wang, J., Xu, H., Tan, W., & Yang, Y. (2017). Secure and efficient identity-based proxy signature scheme in the standard model based on computational Diffie–Hellman problem. *Arabian Journal for Science and Engineering, 42*(2), 639–649.
23. Gentry, C., & Silverberg, A. (2002). Hierarchical ID-based cryptography. In *International conference on the theory and application of cryptology and information security* (pp. 548–566). Berlin, Heidelberg: Springer.
24. Khurana, H., Hadley, M., Lu, N., & Frincke, D. A. (2010). Smart-grid security issues. *IEEE Security and Privacy, 8*(1), 81–85.
25. Li, H., Dai, Y., Tian, L., & Yang, H. (2009). Identity-based authentication for cloud computing. In *IEEE international conference on cloud computing* (pp. 157–166). Berlin: Springer.
26. Lim, H. W., & Paterson, K. G. (2005). Identity-based cryptography for grid security. In *e-Science and Grid Computing, 2005. First International Conference on* (10 pp). IEEE.
27. Baek, J., Vu, Q. H., Liu, J. K., Huang, X., & Xiang, Y. (2015). A secure cloud computing based framework for big data information management of smart grid. *IEEE Transactions on Cloud Computing, 3*(2), 233–244.
28. Ramesh, D., Mishra, R., & Pandit, A. K. (2018). An efficient stream cipher based secure and dynamic updation method for cloud data centre. In *International Conference on Soft Computing Systems* (pp. 505–516). Springer, Singapore.
29. Ramesh, D., Mishra, R., & Nayak, B. S. (2016). Cha-Cha 20: stream cipher based encryption for cloud data centre. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, ACM, 40
30. Ramesh, D., & Priya, R. (2016). Multi-authority scheme based CP-ABE with attribute revocation for cloud data storage. In *Microelectronics, Computing and Communications (MicroCom), 2016 International Conference on* (pp. 1–4). IEEE