Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 5816765, 17 pages
https://doi.org/10.1155/2018/5816765

*Research Article*

# A Secure and Scalable Data Communication Scheme in Smart Grids

**Chunqiang Hu**,[1,2,3] **Hang Liu**,[3] **Liran Ma**,[4] **Yan Huo**,[5] **Arwa Alrawais**,[6] **Xiuhua Li**,[7] **Hong Li**,[8] **and Qingyu Xiong**[1,2]

[1]*Key Laboratory of Dependable Service Computing in Cyber Physical Society, Chongqing University, Ministry of Education, Chongqing, China*
[2]*School of Software Engineering, Chongqing University, Chongqing, China*
[3]*Department of Electrical Engineering & Computer Science, The Catholic University of America, Washington, DC, USA*
[4]*Department of Computer Science, Texas Christian University, Fort Worth, TX, USA*
[5]*School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing, China*
[6]*Department of Computer Science, The George Washington University, Washington DC, USA*
[7]*Department Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada*
[8]*Beijing Key Laboratory of IOT Information Security Technology, Institute of Information Engineering, CAS, Beijing, China*

Correspondence should be addressed to Chunqiang Hu; hcq0394@gmail.com and Qingyu Xiong; xiong03@cqu.edu.cn

Received 5 August 2017; Accepted 13 November 2017; Published 19 March 2018

Academic Editor: Chaokun Wang



The concept of smart grid gained tremendous attention among researchers and utility providers in recent years. How to establish a secure communication among smart meters, utility companies, and the service providers is a challenging issue. In this paper, we present a communication architecture for smart grids and propose a scheme to guarantee the security and privacy of data communications among smart meters, utility companies, and data repositories by employing decentralized attribute based encryption. The architecture is highly scalable, which employs an access control Linear Secret Sharing Scheme (LSSS) matrix to achieve a role-based access control. The security analysis demonstrated that the scheme ensures security and privacy. The performance analysis shows that the scheme is efficient in terms of computational cost.

## 1. Introduction

The concept of smart grid gained tremendous attention among researchers and utility providers in recent years. With such a technology, advanced developments such as sensing, control, digital communications, and networking are integrated into the power systems to effectively and intelligently control and monitor the power grid. Generally speaking, the power grid consists of three major components: power generation, power transmission, and power distribution [1]. Typically, wired communications such as optical networking are adopted to support the power backbone consisting of the power generation and transmission systems [2]; but for the power distribution network, which provides power directly to customers, both wired and wireless communications are adopted.

Smart grid brings new features into the power grid: renewable-based generation, demand-response, wide area protection, and smart metering, just to name a few [3]. Within a smart grid, utility companies can send alerts to notify customers and may further ask them to reduce their power consumption by temporarily turning off some devices during the periods of peak energy consumption [4]. The certain critical control actions can be sent from the control center to smart meters, in which the actions are expected to be taken immediately for safe operations, and the wide area protection schemes can be deployed to prevent cascaded failures and provide better interconnections. However, despite the attractive features provided by smart grid technologies, challenges, especially those in cyber security and privacy [5], are still present. For example, it has been reported that the pervasively adopted integrated Supervisory Control and Data

Acquisition (SCADA)/Energy Management systems [6] are vulnerable to significant security threats [7–10].

As paper [11] pointed out, we need new technologies to protect the confidentiality of the customer's data. Also, customer's privacy should be preserved when data are collected for marketing purpose. It has been demonstrated by [12] that, even without a priori knowledge of household activities, it is still possible to extract customers' usage patterns from the data uploaded by smart meters once every 15 minutes.

Utility companies need customer's energy consumption data for billing purpose. Third-party service providers may need to collect electricity usage records of certain smart devices to monitor device's status and detect potential problems. Some other data analysis companies may need user's energy consumption data to do market research. From customer's perspective, customer should have control over their own data. It means that customer knows and controls the access to his own energy consumption data. If the data is needed for marketing purpose, customer should be informed and guaranteed that his own data are anonymized. Traditionally, smart meter needs to learn receiver's identity (e.g., smart meters should know the certificate of the utility company) and decides whether to send its data or not. For such a large communication network, it may not desirable for smart meters to learn all the identities. And the wide used public-key infrastructure based on X.509 protocol on Internet does not provide enough security guarantees since a fake or stolen certificate may cause tremendous damage and loss in smart grid communication network. On the other hand, all the data can be uploaded into a data repositories [13–15], which store customers' data and distribute them to the third-party service providers under the supervision of a fine-grained access control. It is the data repositories' responsibility to enforce the access control policies and distribute customers' data based on customer's choice and the related regulations and laws, which certainly put tremendous burden on the data repository servers since the compromise of a data repository server reveals all the data it maintains.

To tackle the challenges, we take a fundamentally different approach by employing attribute based crypto system: attribute based encryption (ABE) enables the smart meters to encrypt its data on a set of descriptive attributes, which determine the access privilege of the data. All the legitimate users that may have different identities but possess appropriate sets of attributes can decrypt the data independently. This successfully implements a secure multicast of the customer's data to multiple users, and the smart meters even do not need to know the receiver's detailed identity. Attribute based signature (ABS), in which a signature attests not to the identity of the individual who endorsed a message, but instead to a (possibly complex) claim regarding the attributes she possesses [16], provides a strong unforgeability guarantee for the verifier that the signature was produced by a single party whose attributes satisfy the claim being made. Also, the signature reveals nothing about the identity and even the attributes of the signer beyond what is explicitly revealed by the claim being made. This successfully solves the problem of data anonymity, so that the marketing companies only know that the data comes from the desired group of customer and customer's identity is fully preserved.

Attribute based encryption, more specifically, Ciphertext-Policy Attribute Based Encryption (CP_ABE) [17], provides a secure multicasting and role-based access control. Data stored on data repositories are encrypted and the compromise of data repositories only leaks the encrypted data. It does not need to use a software approach that checks an entities' privilege and decide whether access is granted or not. Attribute based signature is more preferable than other privacy preserving signature schemes such as group signatures [18, 19], ring signatures [20], and mesh signatures [21]; that is, ABS is more practical and provides a stronger guarantee on privacy. Group signature needs a predefined group of people and a group manager. Ring signature needs a predefined group of people too. And the group should be large enough to achieve anonymity. As for mesh signature, it explicitly allows collusion [16], which is not desirable in our case.

ABE and ABS need attribute authority to issue secret keys for attributes so the entity with proper set of secret keys can decrypt and sign a message. In a large scale communication network like smart grid, the attribute authority might become the bottleneck of the entire system. It is desirable to have attribute authority distributed. The decentralized attribute based encryption proposed by Lewko and Waters [22] makes multiauthority possible, and attribute authority does not need to trust each other in the system. Multi-authority ABS has been proposed by Maji et al. [16] that enables multiauthority settings too. In our paper, we mainly focus on implementing and analyzing the decentralized ABE and multiauthority ABS in smart grid communication network.

The contributions of this paper are summarized as follows:

(1) We propose a secure and scalable communication architecture involving multiple authorities, smart meters, data consumers, and data repositories for smart grid systems. Our architecture emphasizes customers' control on their data and privacy.

(2) We implemented decentralized attribute based encryption scheme [22] and multiauthority attribute based signature [16] scheme. We described the communication protocols to achieve customer controlled access control and data anonymity.

(3) We measured the performance of the implemented schemes on different types of curves and groups. We analyzed the efficiency of the implemented schemes and provide future research directions.

The remainder of this paper is structured as follows. In Section 2, we discuss the related work. In Section 3, we introduce the required preliminaries and the system model. Section 4 proposes the secure communication mechanism and presents a scheme to ensure access control for the sensitive data. Section 5 gives performance analysis, followed by the conclusions in Section 6.

## 2. Related Work

In smart grid communication network, security problems mainly lie in the subjects of sensor networks, wireless networks, and Internet. A significant amount of research has been carried out to protect the smart grid systems. Multicast authentication schemes such as TELSA, Biba, HORS, and OTS [3, 23] were proposed for authenticating entities such as utility companies and control centers when messages or control commands are sent to smart meters. To authenticate smart meters or other smart devices to the control center, batch verification schemes [24–27] were developed to improve the efficiency. Data aggregation based on homomorphic encryption, secret sharing, and other technologies [13, 25, 26] was designed to aggregate customers' data and to protect their privacy.

Recently, ABE has received significant amount of attention in securing smart grids because it does not require certificates and it can be used to construct a fine-grained access control mechanism. Actually, the original motivation for ABE scheme is to design an error-tolerant (or fuzzy) identity-based encryption scheme [28] that could be applied to biometric identities. However, the original threshold ABE scheme in [28] is not very impressive as it is limited from designing more general systems. A more general idea called key-policy attribute based encryption (KP_ABE) was proposed by Goyal et al. [29] to embed a general secret sharing scheme for a monotonic access tree instead of the Shamir secret sharing scheme used in [28]. Later, Bethencourt et al. proposed the Ciphertext-Policy Attribute Based Encryption (CP_ABE or BSW CP_ABE) scheme [17] that reverses the KP_ABE construction: the encrypted data (the ciphertext) carries an access structure over attributes; meanwhile, a user's private key is associated with a set of descriptive attributes. The owner or the encryptor now has more control over the data by constructing an access structure for every data to be encrypted.

Later, ABE has been utilized to fit practical problems. Pirretti et al. implemented the threshold ABE system [30] while Chase [31] provided a construction for a multiauthority attribute based encryption system. A decentralized Ciphertext-Policy Attribute Based Encryption scheme was proposed in [22], which deals with the fact that, in practice, there may be more than one attribute authority. And we implemented the decentralized ABE in prime order group in this paper and further analyze the computational cost in different curves and groups.

ABS was introduced by Maji et al. in [16] to achieve a strong unforgeability guarantee for the verifier, which means that the signature was produced by a single party whose attributes satisfy the claim being made. And the privacy of the singer is fully preserved since the signature reveals nothing about the identity or attributes of the signer beyond what is explicitly revealed by the claim being made. However, the security proof in [16] is in generic model group. Later Li et al. proposed an ABS scheme that is selective secure in standard model. But the scheme deals with only $(t, l)$-threshold, which means that it may not be as expressive as Maji et al.'s ABS scheme, which uses an monotone access structure. Moreover, since we prefer large universe construction in smart grid communication network, it is hard and unpractical to implement schemes that are secure in standard model (usually we need to have a polynomial $P(x)$ with degree $d$ and the size of public parameters grows with $d$). We implemented and analyzed Maji et al.'s multiauthority ABS [16] scheme in this paper. One has to notice that, in multiauthority attribute based crypto system, attribute authorities are completely independent from each other, which is a desirable feature for large scale, distributed smart grid communication network.

As a promising technique, identity/attribute based crypto system has been proposed to solve problems in smart grid communication network. A scheme that employs IBE to provide a zero-configuration encryption and authentication solution for end-to-end secure communications was proposed in [32]. The concept of IBE was utilized by [25] to construct a signature and later verify the signature. KP_ABE was adopted by [33] to broadcast a single encrypted message to a specific group of users. Reference [13] utilizes the Linear Secret Sharing to construct the access policy [22, 34] and then enforce access control. However, most of the works done before have no implementation and real life performance analysis. This paper serves as a step that brings the discussion to a more practical stage: implementation and performance analysis. Essentially, the decentralized ABE scheme and multiauthority ABS scheme have their own set of parameters. There are works which have been done to combine ABE with ABS [35], which can be a potential future research direction.

## 3. Preliminaries

In this section, we mainly introduce the preliminaries related to our actual implementation. Theoretical preliminaries can be found in [16, 22, 36, 37].

*3.1. Bilinear Maps.* Let $\mathbb{G}, \mathbb{G}_T$ be cyclic groups of prime order $r$. Let $g$ be a generator of $\mathbb{G}$. A symmetric bilinear map [38] $e$ is an efficiently computable function:

$$e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T, \tag{1}$$

such that

Float 1: *(Nondegeneracy)* $e(g, g) \neq 1$;

Float 2: *(Bilinearity)* $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}$.

A asymmetric bilinear map is that $e$ is an efficiently computable function:

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T, \tag{2}$$

and the property of Nondegeneracy and Bilinearity still hold. We run our implementation on both symmetric and asymmetric pairings and analysis the efficiency.

*3.2. Access Structure.* We mainly discuss monotone access structure (MAS) [39] here.

Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, \ldots, P_n\}}$ is monotone if $\forall B, C$, $B \in \mathbb{A}$ and $B \subseteq C$ imply $C \in \mathbb{A}$. An monotone access structure is a monotone collection $\mathbb{A}$

of nonempty subsets of $\{P_1, \ldots, P_n\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets. For example, let $P = \{A, B, C, D\}$, $\{\{A, B\}, \{B, C\}, \{C, D\}, \{A, B, C\}, \{A, B, D\}, \{A, C, D\}, \{A, B, C, D\}\}$ be a MAS. More importantly, we use a Boolean formula (with only AND and OR gates) to describe a MAS. For example, we are using ($A$ AND $B$) OR ($B$ AND $C$) OR ($C$ AND $D$) to represent the MAS mentioned before.

We are more familiar with $(t, l)$-threshold gate and a threshold gate in [17] can be represented as Boolean formula. For example, an (2, 3)-threshold gate of $\{A, B, C\}$ can be expressed as ($A$ AND $B$) OR ($B$ AND $C$) OR ($A$ AND $C$). In this paper, we are using Boolean formula to express an access structure.

Further, we are using the linear secret sharing schemes (LSSS) proposed in [39, 40], which means we will parse a Boolean formula into a access matrix $A$ and a mapping $\rho()$, where $A$ is called the share-generating matrix and $\rho()$ maps rows of the matrix to the elements in the Boolean formula. Formally, $A$ has $\ell$ rows and $n$ columns, and the $x^{\text{th}}$ row of $A$ will be mapped to an elements in Boolean formula by the function $\rho(x)$. When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, $A \cdot v$ is the vector of $\ell$ shares of the secret $s$. The share $(A \cdot v)_x$ belongs to the element $\rho(x)$.

We use the converting method in [22] and the detailed algorithm is described in Section 5.6. Here is an example: consider an access structure ($A$ AND ($D$ OR ($B$ AND $C$))); the corresponding access matrix and $\rho()$ will be

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{pmatrix} \overset{\rho()}{\longmapsto} \begin{bmatrix} A \\ D \\ B \\ C \end{bmatrix}. \tag{3}$$

For an authorized set $\{A, B, C\}$, the corresponding matrix $A_{\text{matched}}$

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{pmatrix} \tag{4}$$

has a vector $(1, 0, \ldots, 0)$ in their span. In other words, there is a vector $c_x$, which in this case is $(1, 1, 1)$, and $c_x \times A_{\text{matched}} = (1, 0, \ldots, 0)$. In this case,

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \times \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}; \tag{5}$$

$(c_x \times A_{\text{matched}}) \times v^T = s$ means that once we have $A_{\text{matched}}$ and $c_x$, we can recover $s$. The processing described above is called *linear reconstruction*.

Note that we do not lose any efficiency by using the LSSS matrix as opposed to the previously used tree access structure descriptions in [17]. The reason is that the computational cost is directly related to the number of attributes involved in the encryption or sign, and the computational cost of linear reconstruction or polynomial interpolation is negligible. Section 5 will go through a detailed analysis of computational cost.

*3.3. Security Notions and Models.* There are two security notions in identity-based encryption: selective-ID secure and fully secure. Selective secure, introduced by Canetti et al. [41, 42], is weaker than fully secure, which was introduced by Boneh and Franklin in [43]. Generally speaking, fully secure means that the scheme is secure even if the adversary adaptively selects identity to attack based on previous secret keys. For selective secure, the adversary must commit ahead of time to the identity that he will attack. In other words, adversary in fully secure is more powerful since he can query even after he receives the identity to attack.

There are several security models for public-key crypto system. The *random oracle model* was first introduced by Bellare and Rogaway [44]. It assumes that the adversary has the access right to a public, truly random hash function$H$, which is based on SHA-1. Random oracle model is very useful in practice, but from a theoretical perspective, the *standard model* is more preferred. In the standard mode, security is proven using only standard complexity assumptions. For example, [45] is built on *Decisional Bilinear Diffie-Hellman Assumption* and *Computational Diffie-Hellman Assumption*.

Even if standard model is desirable from the perspective of theory, random oracle model is more practical especially when it comes to large universe construction. Paper [46] is fully secure under standard model. But we need to random a set of group elements for attributes in the system. It means that attributes are defined at the setup and published in the public parameters. We call this kind of construction as "small universe construction." In practice, especially in a communication network like smart grid, it is desirable to dynamically use any attribute as we want. The easy way to do this is to use a hash function that we model as a random oracle to map an attribute to a group element. However, we end up with a scheme that secure in random oracle model.

If we still adopt the standard model, we can use a polynomial $P(x)$ with degree $d$ [46] and map attributes in $\mathbb{Z}_q$ to elements in $\mathbb{G}$ by setting $H(\text{attribute}) := g^{P(\text{attribute})}$, where $g$ is the generator of group $\mathbb{G}$. The public parameters would then include $\{g^{P(x)}\}$ for $d + 1$ points $x$ so that $H(\text{attribute})$ could be computed for any attribute by polynomial interpolation. One has to notice that, in practice, we not only need to map an attribute into a group element, but also need to map an identity (which we call it uid in this paper) into a group element. Since $P(x)$ is a $(d + 1)$-wise independent function modulo primes, the system is vulnerable to collusion attacks when a user has $d + 1$ secret keys or more than $d + 1$ users get together to collude. To prevent this from happening, we need to set $d$ large enough so that no users will have more than $d + 1$ secret keys and it is impossible for more than $d + 1$ users to get together and collude. This will boost the size of public parameters and the assumption that no more than $d+1$

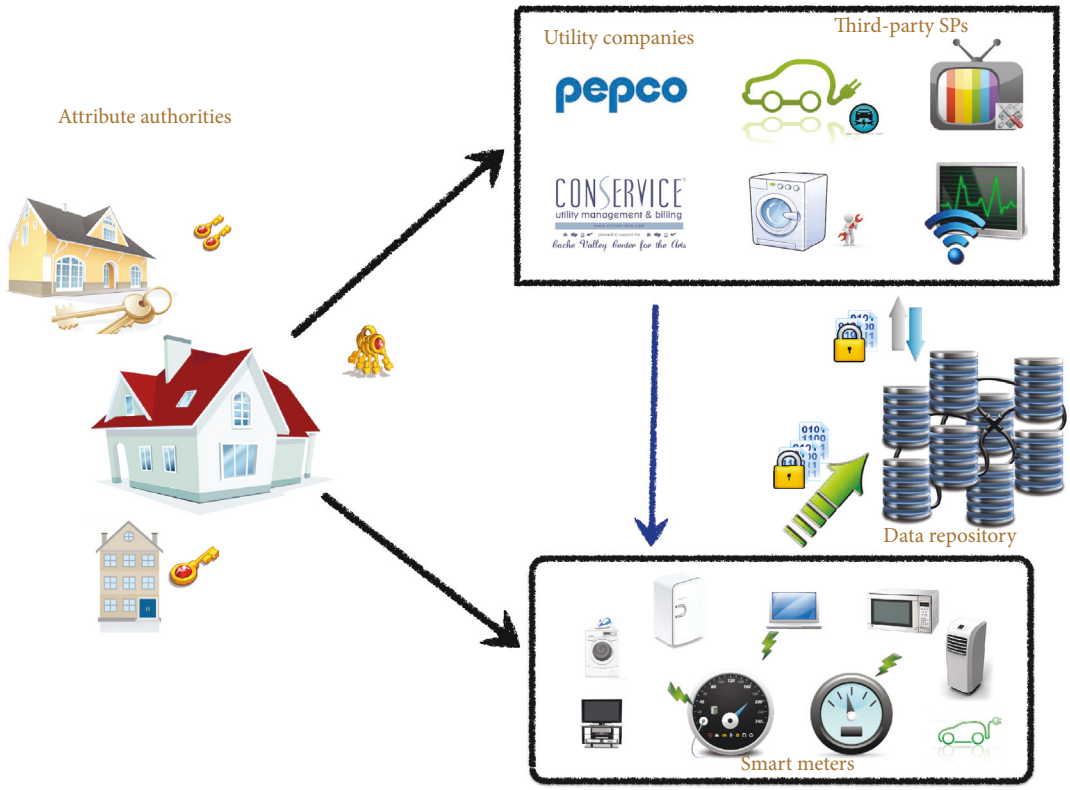


Figure 1: A communication architecture in smart grid systems (Hu et al. (2017) [14]).

users will collude sounds less convincing than random oracle model and a SHA-1 hash function.

*3.4. Generic Group Model, Composite, and Prime Order Groups.* Besides random oracle model and standard model, there is a model called *generic group model*, proposed by Shoup [47]. The model relies on hardness of problems related to finding the discrete logarithm in a group with bilinear pairings. In the model, algorithms can only manipulate group elements via canonical group operations (including the bilinear pairing). We are using prime order groups here in our paper since prime order subgroups of general elliptic curve groups are good examples of groups where all known attacks against the discrete log problem are not significantly better than attacks in the generic group. The multiauthority ABS [16] is secure in the generic group model. The decentralized ABE [22] in prime order groups is secure in generic group model too.

*Bilinear groups of composite order* were introduced by Boneh et al. [48]. Since the elliptic curve group order $n$ must be infeasible to factor in composite order group, it must be at least 1024 bits. On the other hand, the size of a prime order elliptic curve group that provides an equivalent level of security is 160 bits. It is not practical to implement the decentralized ABE scheme on composite order group since group operations and especially pairing computations are prohibitively slow on composite order curves [49]. A Tate pairing on a 1024-bit composite order elliptic curve is roughly 50 times slower than the same pairing on a comparable prime order curve [49]. The small universe construction of decentralized ABE is fully secure in standard model in composite order groups. However, we implemented the decentralized ABE scheme in prime order group and the security reduced to the generic group model.

In summary, we implemented the decentralized ABE and multiauthority ABS that are secure in generic group model. We test and analyze the performance of implementation under both symmetric groups and asymmetric groups. And we are using LSSS matrix and linear reconstruction in our implementation.

## 4. Architecture and Protocol

In this section, we introduce our architecture and communication protocol. Generally speaking, we use decentralized ABE to achieve a fine-grained access control on data collected by smart meters. Also, multiauthority ABS has been used to achieve data anonymity when data consumers or marketing companies need data from certain area or subset of smart meters while user's privacy needs to be preserved.

*4.1. System Model.* We consider the architecture in Figure 1 as the basis of our following discussion. Figure 1 reproduced from Hu et al. (2017) [14]. There are different entities in the communication structure: attribute authorities (AAs), smart meters, data repositories, and data consumers. Data

consumers mainly refer to the utility companies (UCs) and third-party service providers (TPDCs). The following sections are a brief introduction to all the entities.

*(1) Attribute Authorities (AAs).* AAs are responsible for generate and distribute secret keys for smart meters and data consumers. There are multiple AAs in the system and they may not know each other or trust each other. An AA is only responsible for generating secret keys for attributes. We assume that every entity in the system has a unique identifier (GID or uid), and any entity should prove its identity to AA if it needs secret key for its attributes. In this, we do not discuss how to obtain the GID or uid for an entity and how to prove its identity for AAs. Generally speaking, in a communication network like smart grid, every entity (e.g., smart meter) has a unique ID and registered in certain government authorities. The distribution of secret keys can be done by preestablished channel.

Note that a signature trustee should be deployed besides AAs in a multiauthority ABS system. The signature trustee is responsible for issuing an "ID" to the entity. We model the signature trustee as an attribute authority in our architecture.

*(2) Smart Meters.* Smart meters are the key entities in a smart grid communication network. Smart meters collect user's energy consumption information and other pieces of information. In a home area network, smart meters are the center controller. Smart meters monitor the activities of every smart device in the home area. In our architecture, smart meters mainly collect user smart devices' energy consumption information. The total energy consumption can be used by UCs to charge the bill. Energy consumption by some smart devices (e.g., e-cars, TV, and PCs) can be used by third-party SPs to analysis device's working status and diagnose potential problems. Also, TPSPs can use those data to do market analysis and further guide the marketing. However, in this case, anonymity should be enforced to preserve user's privacy and we are proposing multiauthority ABS to achieve data's anonymity.

Each smart meter has a unique officially certified ID, which registers in the system. The communication between smart meters (and any other entities that need secret keys from AAs) and AAs is preestablished secured channel, which is out of our paper. Identity-based encryption/signature systems are an intriguing candidate to establish a secure channel between smart meters and AAs since every single entity is uniquely identified. We leave the integration of identity-base encryption/signature schemes as one of the future works. In the same time, smart meters use attribute based encryption to encrypt its data to achieve a user defined fine-grained access control. For example, the user can construct an access structure ("ARLINGTON.22202" and "ARLINGTON.UC") and encrypt data with it. Only the entity that has corresponding valid set of key can decrypt the data. The data consumers may need data for market purpose and want to protect users' privacy too. Smart meters can sign a data with the secret key for attributes and claim that the secret keys it process satisfy the predicate, which is the access structure or access matrix. One has to notice that we trust smart meters to honestly encrypt and sign a message. The compromise of a smart meter may cause some misbehavers. For example, the attacker controls some smart meters to encrypt and sign any data at any frequency. Further mechanisms should be adopted to secure smart meter and detect the attacks, which is also beyond our discussion here.

*(3) Data Repositories.* Data repositories are storage facility that stores the encrypted or signed data. In attribute based crypto system, the data needs to be encrypted or signed once and later any entities with appropriate set of secret keys can decrypt. Instead of store all the data themselves, smart meters can upload the data to the data repositories and data consumers can retrieve the data from the repository. Data repositories should have higher network throughout capacity. It is certainly more reasonable to have some data repositories with high network bandwidth than having all communication between smart meters and data consumers directly, which may require every smart meter to have higher network processing capacity.

The deployment of data repositories does not affect the confidentiality of the data encrypted under an access structure. The data uploaded by smart meters are encrypted with ABE and only the entities with appropriate set of secret keys can be decrypted. ABE reduces the trust we traditionally put on a data repositories, which has software to enforce the access policy based on the records to describe every entity's privilege. ABE's key feature is the fine-grained access control provided by underlying cryptography algorithms. The data repositories handle the request and deliver the data. Even if a data repository is compromised, the data are safe since they are encrypted.

Note that the data is already protected by ABE and we do not need to have a secured channel between data repositories and other entities. However, the assumption is that every entity in the system has a unique identifier and every entity has the ability to verify the sender's identity. This can be done with identity-based encryption/signature, of which we leave the integration as one of the feature works.

*(4) Data Consumers.* Data consumers refer to utility companies (UCs) and third-party service providers (TPSPs). Generally speaking, UCs need the data collected by smart meters to do the billing. TPSPs may need the data collected by smart meters regarding a specific device to understand their working status and detect potential problems. Also, TPSPs may need data to do market research while they protect user's privacy. Briefly, if data consumers need data, they can retrieve data from data repositories and decrypt it if they have required secret keys to satisfy the access structure. Data consumers verify the signature on data during anonymity data collection, too.

*4.2. Protect Smart Meter's Data with ABE.* We implemented appendix D of paper [22]. Decentralized ABE scheme will enable user defined access control to the data. We will talk about how we use decentralized ABE scheme to protect the data collected by smart meters in this subsection. The scheme

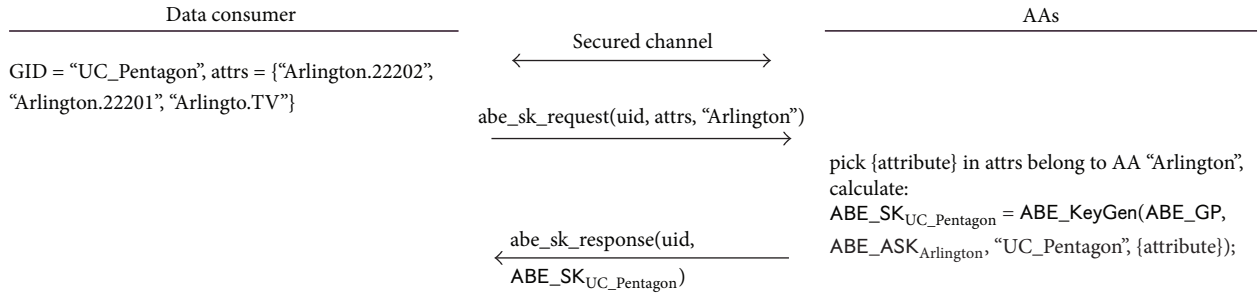


Figure 2: DABE: secret key generation.

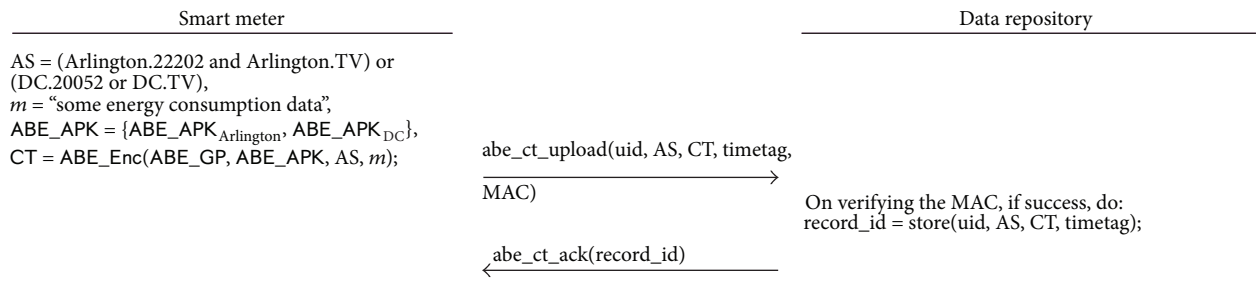


Figure 3: DABE: encryption.

we implemented can be found at Appendix of this paper and the following subsections briefly describe the algorithm and the communication protocol.

*(1) Global Setup.* Global setup in DABE will output ABE_GP, which contains the generators, an hash function we model as a random oracle. Also, $e(g_1, g_2)$ is precalculated.

*(2) Authority Setup.* We describe the AAs as the issuer of secret keys for attributes. One has to notice that AAs are independent with each other and even if two AAs issue secret keys for the same attribute called "TV," they are essentially different and one should specify which AA the attribute belongs to during the encryption and decryption. We are using the format "Arlington.TV" to represent an attribute. The first part of the attribute name is the name of the AA and the second part is the description of the attribute. In this way, attribute "WashingtonDC.TV" is different from "Arlington.TV" and it becomes much more clear during the encryption and decryption regarding which AA an attribute belongs to. In the attribute authority setup, the AA will generate two random exponents for each attribute and publishes PK, which contains all the public keys for attributes and AA will save exponents as the secret key. For example, given an input,

{ABE_GP, attributes = {"22201", "22202", "*CAR*", "*TV*"}, *AA*name = "Arlington"}

Algorithm ABE_AuthoritySetup() will output:

$\mathsf{ABE\_APK}_{\text{Arlington}}$ := {"Arlington.22201": $\{\mathbb{G}_T \times \mathbb{G}_1\}$, "Arlington.22202": $\{\mathbb{G}_T \times \mathbb{G}_1\}$, "Arlington.CAR": $\{\mathbb{G}_T \times \mathbb{G}_1\}$, "Arlington.TV": $\{\mathbb{G}_T \times \mathbb{G}_1\}$}

$\mathsf{ABE\_ASK}_{\text{Arlington}}$ := {"Arlington.22201": $\{\mathbb{Z}_q \times \mathbb{Z}_q\}$, "Arlington.22202": $\{\mathbb{Z}_q \times \mathbb{Z}_q\}$, "Arlington.CAR": $\{\mathbb{Z}_q \times \mathbb{Z}_q\}$, "Arlington.TV": $\{\mathbb{Z}_q \times \mathbb{Z}_q\}$}

as the APK and ASK for AA "Arlington," which are python dictionaries indexed by the name of the attribute (the concatenation of AA's name and attribute's name) and $\{\mathbb{G}_T \times \mathbb{G}_1\}$, mean the that they contain an element in $\mathbb{G}_T$ group and an element in $\mathbb{G}_1$ group.

*(3) Attribute Generation.* In order to decrypt a data block encrypted by smart meters with an access structure, data consumers need to process a proper set of secret keys. Data consumers obtain secret keys from AAs first. We assume that data consumers and AAs can establish a secured communication by other ways via identity-based encryption/signature or traditional PKI. Figure 2 illustrates the protocol between data consumer and AAs. Data consumer "UC_Pentagon" needs secret keys for attribute "Arlington.22202," "Arlington.22201," and "Arlington.TV." The AA "Arlington" will first check if the attributes belong to it or not and it will only generate secret keys for attributes it has.

*(4) Encryption.* Smart meters can upload encrypted data to data repository. Data will be encrypted by ABE with an access structure (AS). The *AS* will be converted into a access matrix *A* in the encryption algorithm. Figure 3 illustrates the protocol between smart meter and data repository. There is no need to establish a secured channel forehead since the data transmitted are already encrypted. The MAC in the protocol serves as a proof of sender's identity and protects the integrity of the payload and so do all the MAC described in the following section. If we have identity-based signature in our system, we can use the identity-based signature to sign a digest of the payload. We leave the integration of identity-based encryption/signature as one of the future works.

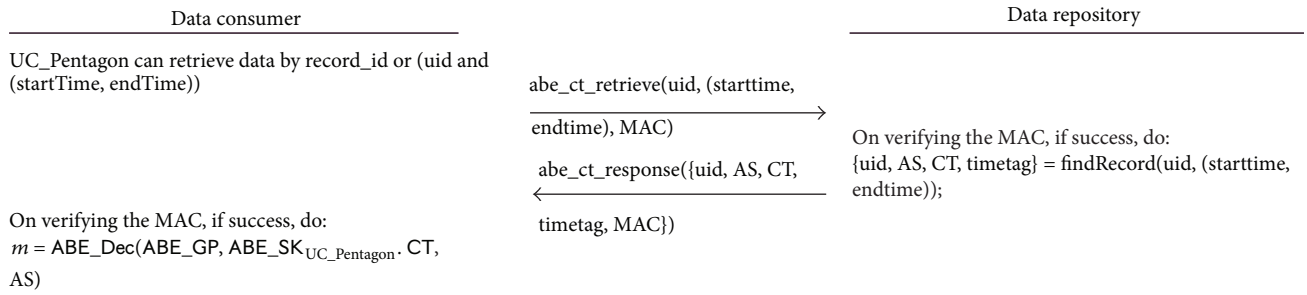


Figure 4: DABE: decryption.

*(5) Decryption.* Data consumer can retrieve the data from data repositories by using the record_id or (uid and (startTime, endTime)). The data repository will return the ciphertext. On receiving the ciphertext, data consumer will decrypt the data with the secret keys it has. Figure 4 illustrated the communication between data consumer and data repository. One has to notice that data consumer will have secret keys from different AAs. And the decryption should distinguish keys from different AAs.

*4.3. Protect Data Anonymity by ABS.* We use the ABS to provide data anonymity and achieve sender's verification. On verifying the signature, the receiver knows that the secret keys the sender have satisfy the access structure and nothing more. ABS provides a strong privacy guarantee. The following subsection describes the communication between entities. The code can be found in Appendix of the paper and we will only highlight the communication protocols in the following subsections. There are researchers working on ABE and ABS that share the same set of parameters [35], but for now, we treat ABE and ABS as separate systems, which means that the global parameters, keys, and other parameters are different.

*(1) Global Setup.* The difference between decentralized ABE and multiauthority ABS is that ABS has one more entities, which is called "signature trustee." Signature trustee will issue a token to a user based on its "uid" and the token must be provided when a user requests secret keys from AAs. In our implementation, we model the signature trustee as one of the AAs (AA "signature trustee") too. And AA "signature trustee" will run the $\mathsf{ABS\_GlobalSetup}(t_{\max}) \rightarrow \mathsf{ABS\_GP}, \mathsf{ABS\_TSK}$. AA "signature trustee" will save $\mathsf{ABS\_TSK}$ and publish $\mathsf{ABS\_GP}$. $t_{\max}$ is the max number of columns in an access structure, which is related to the numbers of AND gate in the access structure. In our implementation, we first give $t_{\max}$ a value and the value can be changed to a larger value in the future if needed.

*(2) Authority Setup.* The authority setup of multiauthority ABS is similar to the authority setup of decentralized ABE except that there is no need to explicitly specify the set of attributes at the setup. One has to notice that the decentralized ABE and multiauthority ABS scheme we implemented are both in large universe construction, which means that we can have as much attributes as we want. AA in multiauthority can issue keys for any attributes. However, in $\mathsf{ABS\_Sign}()$ and $\mathsf{ABS\_Verify}()$, one must explicitly specify the source of the attributes, which means that, for every attribute, one needs to specify which AA it belongs to.

*(3) Token Register and Attribute Generation.* Before entities request secret keys for attributes, the entity needs to register itself at AA "signature trustee." The signature trustee will produce a token for an entity. With the token, an entity can request secret keys from any AAs in the system. One has to notice that secret key for attribute "Arlington.22201" in multiauthority ABS is different from the secret key for "Arlington.22201" in decentralized ABE system even for the same entity. They belong to different scheme and we donate them separately as $\mathsf{ABS\_SK}_{\mathrm{uid}}$ and $\mathsf{ABE\_SK}_{\mathrm{uid}}$. Also, the communication happens in a secured channel. Figure 5 illustrates that the smart meter "SM_RiverhouseApt" requests its token and secret keys from AAs.

*(4) Sign.* To sign a message $m$, the smart meter must have proper set of secret keys. If it does not have, $\mathsf{ABS\_Sign}()$ will abort at the first stage. Also, *AS* will be parsed into an access matrix $A$ with a mapping function $\rho()$. As what we did in decentralized ABE, the *AS* here also explicitly tells the AA of an attribute by using an attribute like "Arlington.22201." Signed data will be uploaded to the data repository too. Figure 6 illustrates the communication between smart meters and data repository.

*(5) Verify.* In verify, if the $\mathsf{ABS\_Verify}()$ returns reject, the verification failed. The verification is successful if it passes all the "checkpoint." Figure 7 illustrates the communication between data consumer and data repository.

*4.4. Combine ABE and ABS.* If a customer wants his smart meter to anonymously sign a data and, in the meanwhile, control the access by an access structure, the smart meter can combine ABE and ABS. It can either sign first and then encrypt or encrypt first and then sign. Since every entity in the system can verify a signature but only the entities with proper set of secret keys can decrypt, our recommendation is to sign and then encrypt. The reason is simple: for those entities that cannot decrypt, we do not want them to know that the signature ever existed. From the perspective of data analysis companies, it can only collect data that intended to been sent to them.

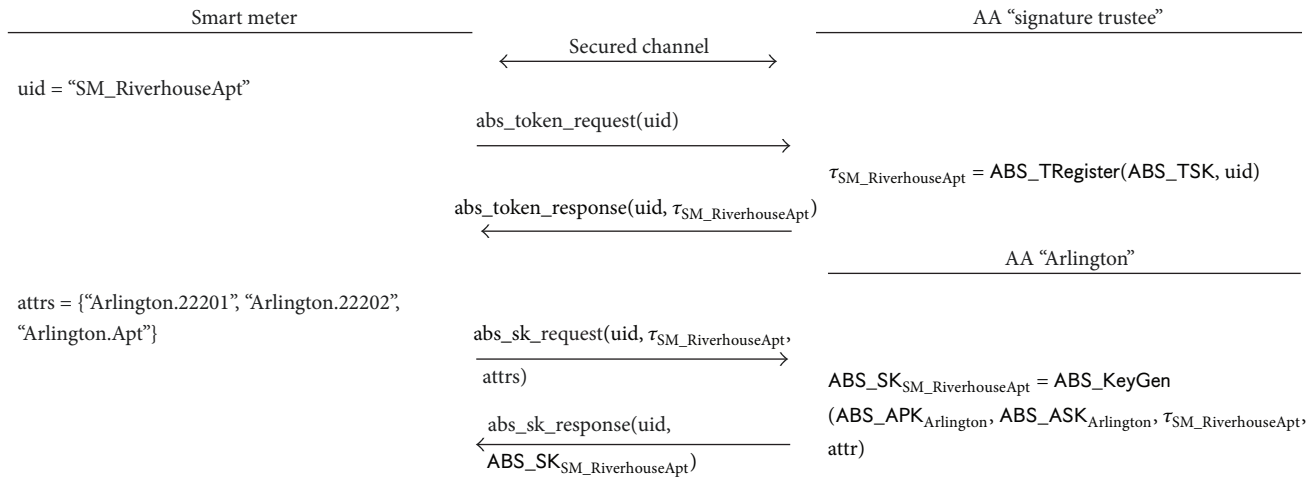


Figure 5: Multiauthority ABS: token and secret key generation.

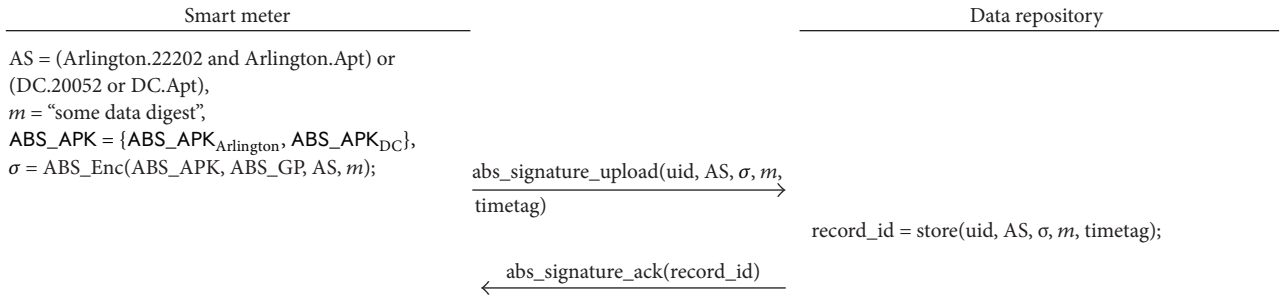


Figure 6: Multiauthority ABS: sign.

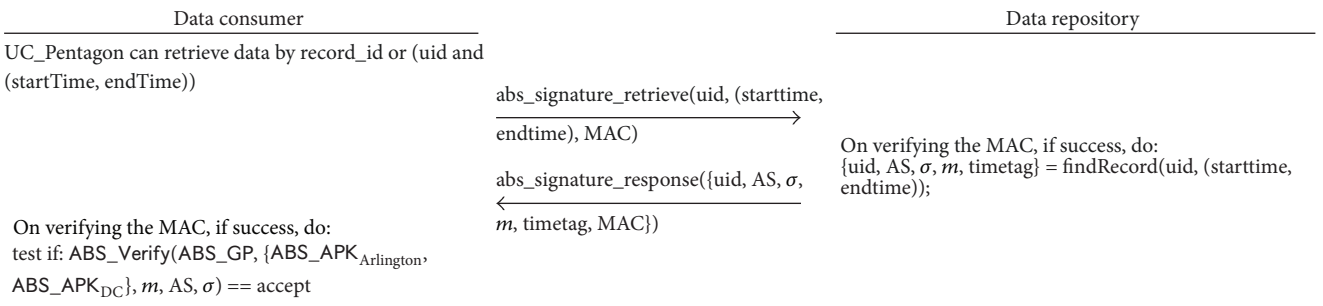


Figure 7: Multiauthority ABS: verify.

*4.5. Eliminate the MAC in the Protocol.* We use a MAC in the communication protocol. Actually, if we have identity-based signature (IBS) in our system, we can use IBS to sign the digest of the payload. To integrate an IBS into our current architecture, we may need a trustee that certifies every identity. We leave it as a future research direction.

In our current architecture, one can remove the MAC by using our ABS and set the access structure to be the sender's identity. The sender (in this case, a smart meter) can obtain a secret key for its identity and sign the message with an access structure that involved only its identity. However, the computational cost of doing this is larger than using IBS and we discourage this particular method.

*4.6. Security Analysis.* Both schemes we implemented are secure in generic group model. In actual large university construction of attribute based crypto system, security in standard is hard to achieve since we need to introduce a polynomial and assumptions that no more than certain amount of user will get together and collude. Generic group model and random oracle model are practical in real-left applications.

## 5. Performance Analysis

We implemented the decentralized attribute based encryption scheme in prime order group, the scheme in appendix

D in [22]. Also, we implemented the multiauthority ABS in Section 4.2 of paper [16]. This section discusses the implementation details and performance analysis.

*5.1. Implementation Details.* The implementation is based on a python library, Charm crypto [50], which is framework that is prototyping advanced cryptosystems such as IBE and IBS. The core mathematical functions behind Charm are from the Stanford Pairing-Based Cryptography (PBC) library [38], which is a free C library that performs the mathematical operations underlying pairing-based cryptosystems. At the same time, there is a project called TinyPBC [51] that has a better performance in terms of elements multiplication in groups. The efficiency of multiplication was improved by a factor of 4-5 and so does the Exponential operation. However, the current release of Charm does not have TinyPBC integrated. We are still using the PBC library for underlying mathematical operations.

The implementation of the decentralized ABE scheme is a little bit different from the original scheme due to the fact that the original paper describes the scheme in symmetric groups. We implemented the decentralized ABE scheme in asymmetric groups and add some precalculated values into the public parameters to reduce the computational cost in Enc() and Dec(). The detailed implementation can be found at Appendix. Since we are using the prime order group instead of the composite order group, the scheme implemented is secure in generic group model. As mentioned before, using composite order groups will largely increase the element size in groups. The computation cost will be boosted especially when we want higher security level; for example, A Tate pairing on a 1024-bit composite order elliptic curve is roughly 50 times slower than the same pairing on a comparable prime order curve [49]. As argued above, generic group model and random oracle model are practical in real life applications. The implementation of multiauthority ABS can also be found at Appendix. Some notations have been changed to avoid confusion.

We are running the code on 32-bit Ubuntu 12.04, which is a virtual machine running in VMWare fusion on a MACbook Air with 1.8 GHz Intel i5 and 4 GB memory. The virtual machine has access to one core of CPU and maximum 1 GB of memory. The PBC library provides a preprocessing mode for Exponential and Pairing. However, we did not use any preprocessing here since Charm did not integrate it. One has to notice that the preprocessing improves the performance by precalculating some value, which means that the preparation itself takes a long time. Preprocessing is recommend when there are a lot of Exponential and Pairing operations to compensate the cost of preparation itself.

*5.2. Groups and Curves.* We implement based on both symmetric groups and asymmetric groups. We will use "SS512" to denote the symmetric group that has a 160-bit order and 512-bit long of base field. A group with order of 160 bits equals 80 bits of NIST symmetric encryption security. For asymmetric groups, we use MNT curve [52] with degree 6 and BN curve [53] with degree 12. To have 80 bits of symmetric security, we use "MNT159," which has a 159-bit base field size in $\mathbb{G}_1$. The field size of $\mathbb{G}_2$ should be 6 times longer than the field size of $\mathbb{G}_1$. However, the PBC library actually implemented $\mathbb{G}_2$ to be 3 times longer. One has to know that the shorter an element in a group is, the faster the multiplication will be and so does the Exponential. As we will see in the following subsections, choosing groups and curves has a great influence to efficiency. The BN curve has a field size of 160 bits in $\mathbb{G}_1$ and the NIST symmetric security is 80 bits too. The degree of BN curve is 12, which means that the operation in $\mathbb{G}_T$ group is more expensive than operations in $\mathbb{G}_T$ in MNT curve, which has a degree of 6. Table 1 is the real world benchmark in Charm of different operations in different groups and curves.

Table 1: Benchmark on curves and groups, time unit is ms. Run 1000 trials and the average is recorded.

| | $\mathbb{G}_1$ Exp. | $\mathbb{G}_2$ Exp. | $\mathbb{G}_T$ Exp. | Pairing |
|---|---|---|---|---|
| SS512 | 3.7289 ms | 3.6653 ms | 0.4652 ms | 3.9136 ms |
| MNT159 | 1.1093 ms | 9.8533 ms | 2.6308 ms | 8.5181 ms |
| BN | 1.1875 ms | 2.3718 ms | 10.5363 ms | 46.1763 ms |

People care about the number of Pairing in an identity/attribute based scheme. Table 1 shows that the Exponential operation consumes equal computational cost. Usually the Pairing operation takes longer than Exponential, but the underlying mathematical function of Charm, which is the PBC library, has no optimizations to the multiplication operation, and the Exponential takes longer than we expect. More discussions of optimization can be found at Section 5.5. For now, the python based Charm crypto is our choice to do the implementation.

From Table 1, the $\mathbb{G}_1$ Exponential is expensive in SS512 since the field size of SS512 is 512 bits while MNT159 and BN have 160 bits of field size. Also, the $\mathbb{G}_2$ Exponential in MNT159 is expensive compared to SS512 even if the element in $\mathbb{G}_2$ is only 3 times longer than $\mathbb{G}_1$, which is about 480 bits. In terms of $\mathbb{G}_2$ Exponential, BN curve is better. BN curve is better in both $\mathbb{G}_1$ Exponential and $\mathbb{G}_2$ Exponential. That is why BN curve is a good candidate when the top priority is to minimize bandwidth (e.g., shorter signature) and faster the schemes that have most of the operation in $\mathbb{G}_1$ and $\mathbb{G}_2$. Another advantage of BN curve is that if finite field discrete log algorithms improve further, MNT curves need to use larger fields, but BN can still remain short [38]. However, $\mathbb{G}_T$ Exponential and Pairing in BN curve take much more longer time than in SS512 and MNT159. If a scheme has heavier operations in $\mathbb{G}_T$ and a large amount of Pairing, we should avoid using BN curve. Different identity/attribute based crypto schemes have different amount of Exponential and Pairing operations in key generation (sometimes called key extraction), encryption, decryption, signature, and verification. We are going to analyze the performance of the decentralized ABE scheme and multiauthority ABS scheme in the following subsection.

*5.3. Performance of Decentralized ABE.* Different curves have different computational costs for Exponential operation in groups. The chosen curves will affect the performance of decentralized ABE scheme. Since Table 1 lists the Exponential

Table 2: Number of operations of decentralized ABE scheme.

| KeyGen() | Enc() | Dec() |
|---|---|---|
| $2\mathbb{G}_2$/attribute | $1\mathbb{G}_T + (3\mathbb{G}_1 + 2\mathbb{G}_T) \cdot \ell$ | $(1\mathbb{G}_T + 2\text{Pairing}) \cdot \ell_r$ |

Table 3: Key generation per attribute of decentralized ABE scheme.

| SS512 | MNT159 | MNT159.S |
|---|---|---|
| 15.61 ms | 53.08 ms | 2.31 ms |

operations in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, we start with the number of Exponential operations in KeyGen(), Enc(), and Dec() of decentralized ABE scheme. Table 2 lists the number of operations for KeyGen(), Enc(), and Dec() of the scheme we implemented, which can be found at Appendix.

$\ell$ is the number of attributes involved in the access structure, and it is also the number of rows in an access matrix. $\ell_r$ is the number of required attributes to decrypt a message. The receiver may not need all the attributes in the access structure to decrypt the message since the minimal set that satisfies the access structure will work.

The key generation needs $2\mathbb{G}_2$ per attribute. The Exponential in $\mathbb{G}_2$ in asymmetric groups is slower than Exponential in $\mathbb{G}_1$. To make the key generation faster, one can play a trick and swap $\mathbb{G}_1$ with $\mathbb{G}_2$. After we swapped $\mathbb{G}_1$ with $\mathbb{G}_2$, the key generation is operations in $\mathbb{G}_1$. However, Enc() will have $3\mathbb{G}_2$ per attribute instead of $3\mathbb{G}_1$. Table 3 is the running time of key generation under SS512, MNT159, and MNT159.S. MNT159.S means that we swapped $\mathbb{G}_1$ with $\mathbb{G}_2$ in the scheme. The swap will not affect the security of the scheme. It will affect only the efficiency and the length of parameters. Note that there are some inconsistency between Tables 1 and 3. The reason that the key generation in MNT159 is about 25 ms longer than we expect is that we need to map an identity to an element in $\mathbb{G}_2$ using a random oracle. And the time of mapping depends on the target group ($\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, or $\mathbb{Z}_q$) and the curve (SS512, MNT, or BN) been used. And the mapping is the reason to the variance in Figure 9 too.

In Figures 8 and 9, the error bar means the standard deviation of the Enc() and Dec(). As we expected, the running time of Enc() and Dec() grows with the number of attributes involved and the number of attributes required, respectively. One can see that MNT159 has the best performance in Enc(), but the worst in KeyGen(). As for Dec(), SS512 is better. MNT159 and MNT159.S should have the same performance in Dec() according to Table 3 since Dec() involves no Exponential in both $\mathbb{G}_1$ and $\mathbb{G}_2$. However, in Dec(), we do need to map an identity to an element in the target group (its $\mathbb{G}_2$ in MNT159 and $\mathbb{G}_1$ in MNT159.S), and as mentioned before, the mapping takes 25 ms when mapping the identity to an element in $\mathbb{G}_2$ in MNT159. This explains why the red line is about 25 ms above the brown line in Figure 9.

In the architecture we proposed, the KeyGen() is performed by the attribute authorities (AAs), and the Enc() and Dec() are performed by smart meters or data consumers. The intuition is that Enc() and Dec() are performed distributively, and the AAs might have bottleneck issues with the fact that there are a large amount of users need secret keys from the AAs. Situation becomes worse if we take key and user revocation into consideration. For example, if the secret keys issued by the AA have a time tag attached (e.g., Arlington.TV.Jan 2013), which means that this attribute will expire in certain amount of time and users should obtain the secret key for the next time period of this attribute by contacting the AA or

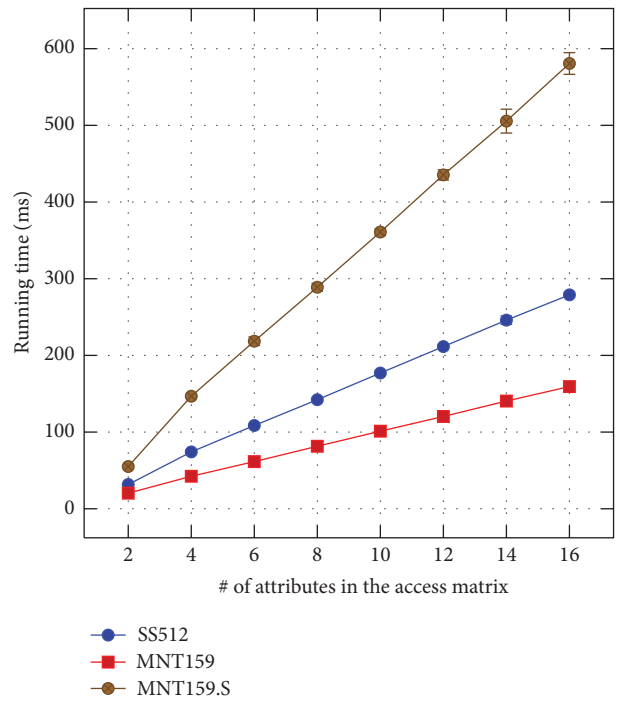


Figure 8: Decentralized ABE: Enc().

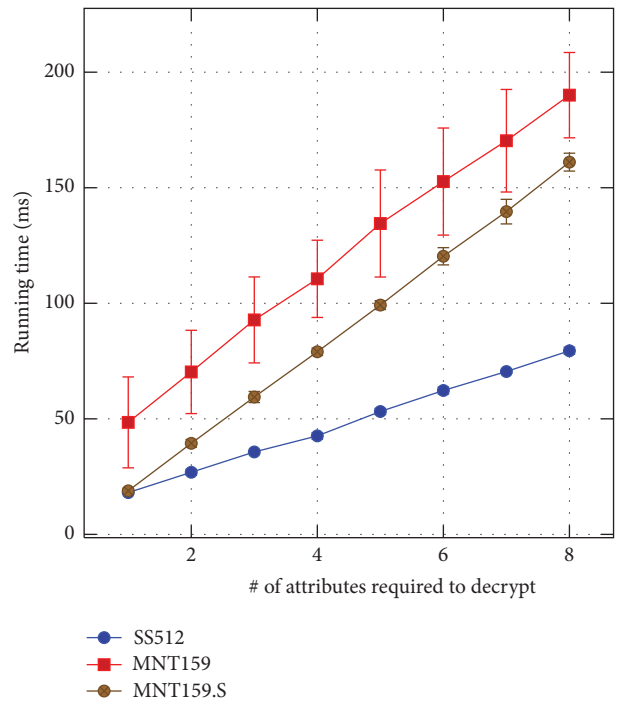


Figure 9: Decentralized ABE: Dec().

we integrate some real-time user (or key) revocation scheme just as paper [54] did, the KeyGen() will certainly cause a lot of pains to the AAs. Our recommendation here is to use MNT159 curve and swap $\mathbb{G}_1$ with $\mathbb{G}_2$ to achieve the best efficiency in KeyGen(). Even the performance of MNT159.S in Enc() is the worst, it will be acceptable due to the fact that the encryption will only need to be performed once and the computation is totally distributed.

*5.4. Performance of Multiauthority ABS.* We also implemented the multiauthority ABS [16] and ran the performance test on our implementation. The difference between the decentralized ABE scheme and multiauthority ABS scheme is that the multiauthority ABS scheme has a signature trustee, which handles the user registration part. Given the token from signature trustee, AAs can generate secret keys for attributes the user requested. Since the TSetup() and ASetup() only happen once, we mainly focus on the TRegister(), AttrGen(), Sign(), and Verify(). One has to notice that the verification we implemented is the probabilistic verification mentioned in Section 3.3.1 of paper [16], which has at most $1/p$ probability to make a false positive. The computational cost of verification reduced by one degree: from $\ell \cdot t + 2$ to $\ell + 4$, where $\ell$ is the number of rows in the access matrix and $t$ is the number of columns.

Table 4 summarizes the number of operations for TRegister(), AttrGen(), Sign(), and Verification() of the scheme we implemented, which can also be found at the Appendix.

$\ell$ is the number of attributes involved in the access structure, and it is also the number of rows in an access matrix. $t$ is the number of columns of the access matrix. $t$ increases by one when the algorithm meets an "AND" gate in an access structure. $\ell_r$ is the number of required attributes to sign a message.

Also, we start with the AttrGen(), which may be the bottleneck of our system. The TRegister() has the same amount of computational cost to the AttrGen() according to Table 4. However, we need to use a random oracle to map the identity into an element in groups and the discussion in the previous subsection. However, this mapping may take a long time. Meanwhile, a user needs to contact the signature trustee to get this token, then the user needs more than secret key for the attributes. The computational cost of TRegister() should be less than the computational cost of AttrGen(). We focus on AttrGen() now and we can generalize the performance of TRegister() from Table 5.

As what has been discussed in the previous subsection, MNT159.S swaps $\mathbb{G}_1$ with $\mathbb{G}_2$, and BN curves are brought into discussion as it has its advantages in $\mathbb{G}_1$ and $\mathbb{G}_2$ Exponential operations.

From Figures 10 and 11, error bar means standard deviation. Computational cost in Sign() and Verify() is higher than the Dec() and Enc() in the decentralized ABE scheme just as we expected. The multiauthority ABS signature has a lot more Exponential operations in $\mathbb{G}_1$ and $\mathbb{G}_2$. Particularly in Sign(), it grows with $\ell \cdot t$. It also grows with the number of attributes required to sign. In the access structure we using, the number of required attributes to sign is $\ell_r = \ell/2$. The verification is the probabilistic verification which has a reasonable and negligent probability to produce a false positive. Both MNT159.S and BN.S have a better performance in AttrGen(). As for Sign(), BN.S has the lowest cost since the Exponential operations in $\mathbb{G}_1$ and $\mathbb{G}_2$ are less expensive than other schemes. MNT159.S has better performance in the verification. If considering the performance of Sign() as the priority, BN curve should be used and $\mathbb{G}_1$ should be swapped with $\mathbb{G}_2$. However, the sender needs only to generate one signature for a message and verification might happen more than one time. One can also consider the verification as the priority; MNT159.S would be a better choice.

In the smart grid communication network, the AttrGen() is centralized and may be the bottleneck. Both MNT159.S and BN.S can fit the task. If the efficiency of Sign() matters, one should use BN curve and swap $\mathbb{G}_1$ with $\mathbb{G}_2$. Also, the signature size can be reduced to compare with MNT curve: in $\mathbb{G}_1$, elements have the same length. However, the elements in $\mathbb{G}_2$ are 2 times longer than elements in $\mathbb{G}_1$ in BN curve instead of 3 times longer in MNT curve. If the resource on smart meters is very limited, BN curve will be a good choice.

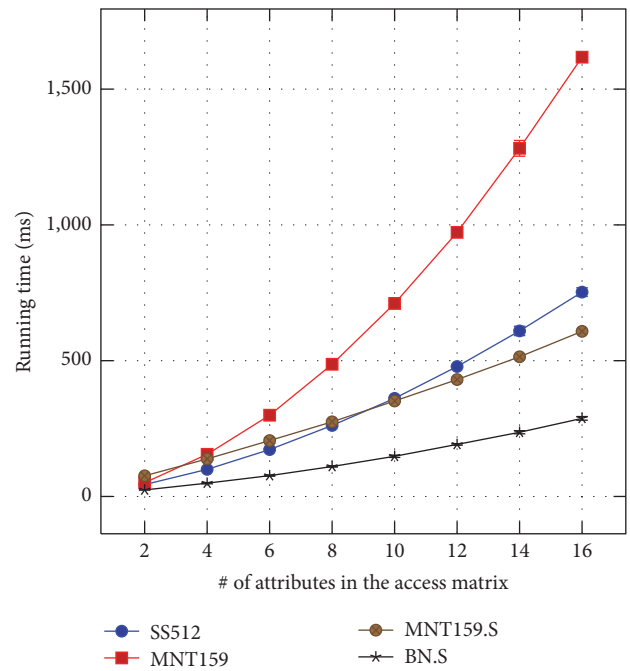


Figure 10: Multiauthority ABS: Sign().

Table 4: Number of operations of multiauthority ABS scheme.

| | |
|---|---|
| TRegister() | $1\mathbb{G}_1$/user |
| AttrGen() | $1\mathbb{G}_1$/attribute |
| Sign() | $2\mathbb{G}_1 + 3(\ell_r)\mathbb{G}_1 + 2(\ell - \ell_r)\mathbb{G}_1 + 2(\ell \cdot t)\mathbb{G}_2$ |
| Verify() | $1\mathbb{G}_1 + 2(\ell \cdot t + t)\mathbb{G}_2 + (\ell + 4)$Pairing |

Table 5: Key generation per attribute of multiauthority ABS scheme.

| SS512 | MNT159 | MNT159.S | BN.S |
|---|---|---|---|
| 3.67 ms | 9.72 ms | 1.13 ms | 2.30 ms |

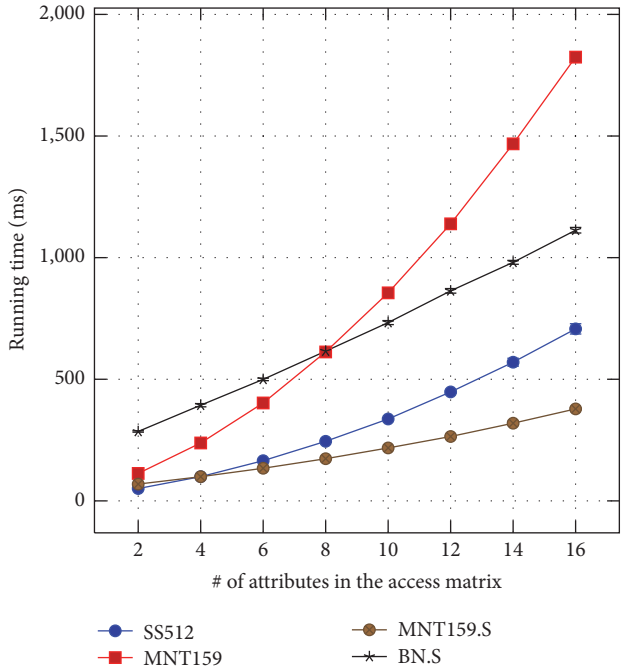


Figure 11: Multiauthority ABS: Verify().

Table 6: Preprocessing in PBC library.

| Average, ms | MNT159 | SS512 |
|---|---|---|
| G1 Exponential | 1.18 | 4.25 |
| After preprocessing | 0.16 | 0.59 |
| Preprocessing itself | 5.47 | 20.37 |
| G2 Exponential | 10.07 | 4.08 |
| After preprocessing | 1.45 | 0.55 |
| Preprocessing itself | 46.74 | 19.01 |
| Pairing | 8.48 | 4.33 |
| After preprocessing | 6.92 | 1.83 |
| Preprocessing itself | 1.68 | 3.86 |

However, if the efficiency of verify() is the priority, MNT159.S should be used. In the scenario that data consumers need to collect anonymity data from a group of users that satisfy an access structure, the verification is performed per user and MNT159.S will save a lot of computational cost.

*5.5. More about Efficiency.* Efficiency can be further improved by using the preprocessing provided by PBC library or using the Lopez-Dahab algorithm [55], which is TinyPBC's optimization on multiplication. Both of them are not in Charm's current release. In PBC library, we can prepare an element for Exponential operation or Pairing operation. For example, if we preprocess the generator $g_1$, the exponential operation based on $g_1$ will be roughly 6-8 time faster, which is shown in Table 6. See Table 6 for details.

According to TinyPBC's implementation on multiplication, the speed of multiplication will be 4–6 times faster. If we combine the preprocessing and Lopez-Dahab algorithm, we expect the implemented scheme to be 20 times faster. Further work needs to be done to optimize the underlying mathematical functions to make the multiplication and pairing faster.

*5.6. Converting an AS to an Matrix More Efficiently.* Algorithm 1 is converting an access structure to an access matrix.

From decentralized ABE and multiauthority ABS scheme, the size of matrix influences the efficiency. The project [56] reduces the size of access matrix and the computational cost of our implemented schemes will reduce too. We leave the implementation of a more efficient transformation of access matrix as one of the potential future research directions too.

In summary, we run the implementation on different curves and groups and analyze the performance in this section. Note that the decentralized ABE and multiauthority ABS do not share the common parameters such as group generator and so on. Reference [35] has some initial work in combining ABE with ABS, and ABE shares public parameters, even the secret keys with ABS. Combined ABE and ABS can be a potential next step in our future work. However, once we combine the ABE and ABS, even the storage for secret keys reduced, computational cost would be higher than using ABE and ABS schemes, separately. Different schemes have different performance under different curves and groups.

## 6. Conclusion

In this paper, we describe a smart grid communication architecture and then present a secure and scalable data communication scheme in smart grids, which is employed decentralized attribute based encryption. The security analysis demonstrated that the scheme ensures security and privacy. The performance analysis shows that the scheme is efficient in terms of computational cost.

Our future research lies in the following directions: design a decentralized CP_ABE scheme with constant size of ciphertext length to reduce the storage and communication cost. Examine more attacks on the architecture we proposed and defend those attacks. Cooperate our current scheme with other broadcast authentication schemes and signature schemes to make a more comprehensive and applicable architecture. The communication architecture for smart grids proposed in this paper serves at the basis of our future research and we shall further propose new approaches to enhance and extend this architecture.

## Appendix

Here are some detailed implementation.

## A. Duplicated Attributes in an AS

For the duplicated attributes in an AS, we will extend the attribute and make them different. For example, if we have two "Arlington.22202" in *AS*, we will encode them into "Arlington.22202_1" and "Arlington.22202_2." However, in the encryption, we will treat "Arlington.22202_1" as "Arlington.22202" and later in the decryption or sign, if the entities

(1) Let $v = (1)$, which represents the root of $T$.
(2) Let $c = 1$, which is a counter for the converting process.
(3) $A$ is the LSSS matrix, initialized to null; $\rho()$ is the function that maps the rows of $A$ to attributes.
(4) BooleanFormula2LSSS(root($T$),$v$, $c$, $A$, $\rho()$);
(5) **function** (*BooleanFormula2LSSS* (*node*, $v, c, A, \rho()$))
(6) **if** *node* is an AND gate **then**
(7) pad $v$ with 0's (if necessary) to make its length = $c$;
(8) label the left child with the vector $v \parallel 1$, where $\parallel$ means concatenation;
(9) label the right child with the vector $(0, \ldots, 0) \parallel -1$, where $(0, \ldots, 0)$ denotes the zero vector of length $c$;
(10) $c = c + 1$;
(11) **else if** *node* is an OR gate **then**
(12) label the left child and the right child with the vector $v$;
(13) **end if**
(14) **for** *node*'s child node as $n_c$ **do**
(15) **if** $n_c$ is an attribute **then**
(16) $A$.push($v$); // add $v$ to the end of matrix $A$
(17) add the mapping $\rho(v) = n_c$;
(18) **else**
(19) BooleanFormula2LSSS($n_c$,$v$, $c$, $A$, $\rho()$)
(20) **end if**
(21) **end for**
(22) **end function**

Algorithm 1: Boolean formula 2LSSS (*node*, $v, c, A, \rho()$).

have the secret key for "Arlington.22202," it can decrypt or sign both "Arlington.22202_1" and "Arlington.22202_2."

## B. The Decentralized ABE Scheme Implemented

The differences between the scheme we implemented and the original paper are as follows:

(i) The original paper was described under symmetric group settings. We implemented it under asymmetric group settings.
ii. Hash function $H$ maps an identity into an element in $\mathbb{G}_2$.
iii. Secret keys for attributes are elements in $\mathbb{G}_2$.
iv. $e(g_1, g_2)$ is precalculated in our implementation.

Here is the scheme we implemented:

(1) ABE_GlobalSetup($\lambda$):

$g_1 \leftarrow_R \mathbb{G}_1$, $g_2 \leftarrow_R \mathbb{G}_2$, $H: \{0,1\}^* \rightarrow \mathbb{G}_2$;
return: ABE_GP := $\{g_1, g_2, H, e(g_1, g_2)\}$;

(2) ABE_AuthoritySetup(ABE_GP, {attribute},$AA$name):

ABE_APK$_{AA\text{name}}$ = {}; # init a python dictionary
2. ABE_ASK$_{AA\text{name}}$ = {}; # init a python dictionary
**for** $i$ **in**{attribute}:

$\alpha_i, y_i \leftarrow_R \mathbb{Z}_q$;
ABE_APK[$AA$name.$i$] = $\{e(g_1, g_2)^{\alpha_i}, g_1^{y_i}\}$;
ABE_ASK[$AA$name.$i$] = $\{\alpha_i, y_i\}$;

return: ABE_APK$_{AA\text{name}}$, ABE_ASK$_{AA\text{name}}$;

(3) ABE_KeyGen(ABE_GP, ABE_ASK, uid, attribute):

ABE_SK$_{\text{uid}}$ = {}; # init a python dictionary
**for** $i$ **in**{attribute}:

ABE_SK$_{\text{uid}}[i] = g_1^{\alpha_i} H(\text{uid})^{y_i}$;

return: ABE_SK$_{\text{uid}}$;

(4) ABE_Enc(ABE_GP, {ABE_APK}, AS, $m$):

parse(AS) $\rightarrow$ $(A, \rho())$; # we implemented the converting in Appendix C of the paper.
$\overrightarrow{v} \leftarrow_R \mathbb{Z}_q^\ell$; # $A$ is a $n \times \ell$ matrix.
$\overrightarrow{w} \leftarrow_R \mathbb{Z}_q^\ell$, $w[0] = 0$;
$\overrightarrow{r} \leftarrow_R \mathbb{Z}_q^\ell$;
**for** $A_x$ as row of $A$:

$\lambda_x = A_x \cdot \overrightarrow{v}$;
$w_x = A_x \cdot \overrightarrow{w}$;

$C_1, C_2, C_3$ = {}, {}, {};
$C_0 = \text{Me}(g_1, g_2)^{v[0]}$;
**for** $x$ as row of $A$:

$C_1[\rho(x)] = e(g_1, g_2)^{\lambda_x} e(g_1, g_2)^{\alpha_{\rho(x)} r_x}$;
$C_2[\rho(x)] = g_1^{r_x}$;
$C_3[\rho(x)] = g_1^{y_{\rho(x)} r_x} g_1^{w_x}$;

return: (CT = $\{C_0, C_1, C_2, C_3\}$, $AS$);

(5) ABE_Dec(ABE_GP, ABE_SK$_{\text{uid}}$, CT, $AS$):

parse($AS$) $\rightarrow$ $(A, \rho())$;
find subset $A'$ of $A$ has $(1, 0, \ldots, 0)$ as the span.
$\overrightarrow{c} \cdot A' = (1, 0, \ldots, 0)$;
**for** $x$ as row of $A'$:
numerator $*$ = $((C_{1,x} \cdot e(C_{3,x}, H(GID))/e(C_{2,x}, K_{\rho(x),GID}))^{c_x}$
return: ($m = C_0$/numerator).

## C. The Implement of Multiauthority ABS Scheme

The differences between the scheme we implemented and the original paper are as follows:

1. $G$ in original paper is $\mathbb{G}_1$ in our implementation. $H$ is $\mathbb{G}_2$.
2. We have two hash functions. $H_1$: $\{0,1\}^* \rightarrow \mathbb{Z}_q$ will be used to map attributes into elements in $\mathbb{Z}_1$; $H_2$: $\{0,1\}^* \rightarrow \mathbb{G}_1$ will be used to map identity into elements in $\mathbb{G}_1$.
3. Secret keys for attributes are elements in $\mathbb{G}_2$.
4. In Sign(), the computing of $S_i$ will not compute $(K_u[H_1(\rho(i))])^{v[i]})^{r[0]}$ if $v[i]$ is 0, which means that the signer does not have the corresponding secret key for the attribute. We save 1 $\mathbb{G}_1$ Exponential by doing so.

Here is the scheme we implemented:

(1) ABS_GlobalSetup($t_{\max}$):

$g_1, C \leftarrow_{\text{R}} \mathbb{G}_1$, $h_0, \ldots, h_{t_{\max}} \leftarrow_{\text{R}} \mathbb{G}_2$;
$H_1$: $\{0,1\}^* \rightarrow \mathbb{Z}_q$, $H_2$: $\{0,1\}^* \rightarrow \mathbb{G}_1$;
$\alpha_0 \leftarrow_{\text{R}} \mathbb{Z}_q$, $A_0 = h_0^{\alpha_0}$;
ABS_GP := $((\mathbb{G}_1, \mathbb{G}_2), H_1, H_2, g_1, A_0, h_0, \ldots, h_{t_{\max}}, C)$;
ABS_TSK := $(\alpha_0)$;
return: ABS_GP, ABS_TSK;

(2) ABS_AuthoritySetup(ABS_GP, $AA$name):

$a, b \leftarrow_{\text{R}} \mathbb{Z}_q$;
$A$, $B$ = list(), list(); # $A$, $B$ are two list.
**for** $j$ in $[1, t_{\max}]$:
$A_j = h_j^a$, $B_j = h_j^b$;
ABS_ASK$_{AA\text{name}}$ := $(a, b)$;
ABS_APK$_{AA\text{name}}$ := $\{A, B\}$;
return: ABS_ASK$_{AA\text{name}}$, ABS_APK$_{AA\text{name}}$;

(3) ABS_TRegister(ABS_TSK, uid):

$K_{\text{base}} = H_2(\text{uid})$, $K_0 = K_{\text{base}}^{1/\alpha_0}$;
return: $\tau_{\text{uid}} = (K_{\text{base}}, K_0)$

(4) ABS_KeyGen(ABS_APK, ABS_ASK, $\tau_{\text{uid}}$, {attr}):

$K_u$ = {}; # init a python dictionary
**for** $i$ **in** {attr}:
$K_u[i] == K_{\text{base}}^{1/(a+bH_1(i))}$;
return: ABS_SK$_{\text{uid}}$ := $\{K_{\text{base}}, K_0, K_u\}$;

(5) ABS_Sign(ABS_GP, {ABS_APK}, ABS_SK$_{\text{uid}}$, $AS, m$):

**if** ABS_SK$_{\text{uid}}$ does not satisfies $AS$:
Abort;
parse($AS$) $\rightarrow$ $(A, \rho())$; # $A$ is an $\ell \times t$ matrix
$\mu = H_1(m \parallel AS)$
$\overrightarrow{v} \cdot M = (1, 0, \ldots, 0)$;
$\overrightarrow{r} \leftarrow_{\text{R}} \mathbb{Z}_q^{\ell+1}$;
$Y := K_{\text{base}}^{r[0]}$, $W := K_0^{r[0]}$;
$S, P$ = list(), list(); # init two list.
**for** $i$ in$[1, \ell]$:
$S_i := ((K_u[H_1(\rho(i))])^{v[i]})^{r[0]} \cdot (Cg_1^{\mu})^{r[i]}$;
**for** $i$ in$[1, t]$:
$P_j = \prod_{i=1}^{l} (A_j B_j^{H_1(\rho(i))})^{M_{ij} \cdot r[i]}$;
return: $\sigma = (Y, W, S, P)$;

(6) ABS_Verify(ABS_GP, {ABS_APK}, $AS, m, \sigma$):

parse($AS$) $\rightarrow$ $(A, \rho())$; # $A$ is an $\ell \times t$ matrix
$\mu = H_1(m \parallel AS)$
**if** $Y == 1$:
return: reject;
$r_1, r_2, \ldots, r_l \leftarrow_{\text{R}} \mathbb{Z}_q$;
**if** $e(W, A_0)! = e(Y, h_0)$:
return: reject;
**if** $\prod_{i=1}^{\ell} e(S_i, \prod_{j=1}^{t} (A_j B_j^{H_1(u(i))})^{M_{ij} \cdot r_j})! = e(Y, h_1^{r_1}) e(Cg_1^{\mu}, \prod_{j=1}^{t} P_j^{r_i})$ :
return: reject;
return: accept.

## Conflicts of Interest

The funding did not lead to any conflicts of interest regarding the publication of this manuscript.

## Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under Grants 61702062, 61471028, 61672119, and 61771077, National Program on Key Basic Research Project of China (973 Program), the State Key Program of National Natural Science of China (no. U1766215), the Major Science and Technology Program of Guangxi (Grant no. GKAA17129002), and the Key Research Program of Chongqing Science and Technology Commission (Grant no. CSTC2017jcyjBX0025).

## References


[1] S. M. Kaplan, "Electrical power transmission," in *Background and policy issues , TheCapital.Net*, vol. Government, p. 42, Electrical power transmission, Background and policy issues, 2009.

[2] X. D. Wang and P. Yi, "Security framework for wireless communications in smart distribution grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 809–818, 2011.

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

[4] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power & Energy Magazine*, vol. 7, no. 2, pp. 52–62, 2009.

[5] "Guidelines for smart grid cyber security (vol. 1 to 3)," NIST, Tech. Rep. NIST IR-7628, Aug 2010, http://csrc.nist.gov/publications/PubsNISTIRs.html.

[6] G. Baker and A. Berg, "Supervisory control and data acquisition (scada) systems," *The Critical Infrastructure Protection Report*, vol. 1, no. 6, pp. 5-6, 2002.

[7] C. Zimmer and F. Mueller, "Fault Tolerant Network Routing through Software Overlays for Intelligent Power Grids," in *Proceedings of the 2010 IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 542–549, Shanghai, China, December 2010.

[8] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, article 13, 2011.

[9] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, 2017.

[10] Y. Deng, C. Hu, R. Deng, and D. Liang, "A secure communication architecture in the smart grid," in *Proceedings of the 2017 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS)*, pp. 668–672, Dalian, July 2017.

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

[12] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proceedings of the the 2nd ACM Workshop*, p. 61, Zurich, Switzerland, November 2010.

[13] S. Ruj, A. Nayak, and I. Stojmenovic, *A security architecture for data aggregation and access control in smart grids*, 1111.2619, ArxivpreprintarXiv, 2011.

[14] C. Hu, Y. Huo, L. Ma, H. Liu, S. Deng, and L. Feng, "An Attribute-Based Secure and Scalable Scheme for Data Communications in Smart Grids," in *Wireless Algorithms, Systems, and Applications*, vol. 10251 of *Lecture Notes in Computer Science*, pp. 469–482, Springer International Publishing, Cham, 2017.

[15] C. Hu, X. Cheng, Z. Tian, J. Yu, K. Akkaya, and L. Sun, "An Attribute-Based Signcryption Scheme to Secure Attribute-Defined Multicast Communications," in *Security and Privacy in Communication Networks*, vol. 164 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 418–437, Springer International Publishing, Cham, 2015.

[16] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures," in *Topics in cryptology*-CT-RSA *2011*, vol. 6558 of *Lecture Notes in Comput. Sci.*, pp. 376–392, Springer, Heidelberg, Germany, 2011.

[17] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, May 2007.

[18] D. Chaum and E. van Heyst, "Group Signatures," in *Advances in Cryptology — EUROCRYPT '91*, vol. 547 of *Lecture Notes in Computer Science*, pp. 257–265, Springer Berlin Heidelberg, Berlin, Heidelberg, 1991.

[19] C. Hu, X. Cheng, J. Yu, Z. Tian, and R. Bie, Achieving privacy preservation and billing via delayed information release , submittedtoIEEETransactionsonServiceComputing,.

[20] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in Cryptology—ASIACRYPT*, vol. 2248 of *Lecture Notes in Comput. Sci.*, pp. 552–565, Springer, 2001.

[21] X. Boyen, "Mesh signatures: how to leak a secret with unwitting and unwilling participants," in *Advances in cryptology—EUROCRYPT 2007*, vol. 4515 of *Lecture Notes in Comput. Sci.*, pp. 210–227, Springer, Berlin, 2007.

[22] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology—EUROCRYPT 2011*, vol. 6632 of *Lecture Notes in Computer Science*, pp. 568–588, Springer, Heidelberg, Germany, 2011.

[23] M. Kgwadi and T. Kunz, "Securing RDS broadcast messages for smart grid applications," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 4, no. 4, pp. 412–426, 2011.

[24] H. Guo, Y. Wu, H. Chen, and M. Ma, "A batch authentication protocol for v2?g communications," in *in NewTechnologies, Mobility and Security(NTMS, 2011 4th IFIP International Conference*, pp. 1–5, IEEE, Paris, France, 2011, 1em plus 0.5em minus 0.4em.

[25] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1632, 2012.

[26] D. Li, Z. Aung, J. R. Williams, and A. Sanchez, "Efficient authentication scheme for data aggregation in smart grid with fault tolerance and fault diagnosis," in *Proceedings of the IEEE PES Innovative Smart Grid Technologies (ISGT '12)*, pp. 1–8, IEEE, January 2012.

[27] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, May 2017.

[28] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology – EUROCRYPT 2005*, vol. 3494 of *Lecture Notes in Computer Science*, pp. 457–473, Springer, Berlin, Germany, 2005.

[29] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98, November 2006.

[30] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," *Journal of Computer Security*, vol. 18, no. 5, pp. 799–837, 2010.

[31] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography*, vol. 4392 of *Lecture Notes in Computer Science*, pp. 515–534, Springer, Berlin, Germany, 2007.

[32] H. K. So, S. H. Kwok, E. Y. Lam, and K. Lui, "Zero-Configuration Identity-Based Signcryption Scheme for Smart Grid," in *Proceedings of the 2010 1st IEEE International Conference on*

*Smart Grid Communications (SmartGridComm)*, pp. 321–326, Gaithersburg, MD, USA, October 2010.

[33] Z. M. Fadlullah, N. Kato, R. Lu, X. Shen, and Y. Nozaki, "Toward secure targeted broadcast in smart grid," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 150–156, 2012.

[34] C. Hu, H. Li, Y. Huo, T. Xiang, and X. Liao, "Secure and Efficient Data Communication Protocol for Wireless Body Area Networks," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 2, no. 2, pp. 94–107, 2016.

[35] C. Chen, J. Chen, H. W. Lim, Z. Zhang, and D. Feng, "Combined public-key schemes: the case of abe and abs," in *Provable security*, vol. 7496 of *Lecture Notes in Comput. Sci.*, pp. 53–69, Springer, Heidelberg, 2012.

[36] C. Hu, X. Liao, and X. Cheng, "Verifiable multi-secret sharing based on LFSR sequences," *Theoretical Computer Science*, vol. 445, pp. 52–62, 2012.

[37] C. Hu, A. Alhothaily, A. Alrawais, X. Cheng, C. Sturtivant, and H. Liu, "A secure and verifiable outsourcing scheme for matrix inverse computation," in *Proceedings of the IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, May 2017.

[38] B. Lynn, *On the implementation of pairing-based cryptosystems [Ph.D. thesis]*, Stanford University, Calif, USA, 2007, Ph.D. dissertation.

[39] A. Beimel, *Secure schemes for secret sharing and key distribution [Ph.D. thesis]*, Israel Institute of Technology, Technion, Haifa, Israel, 1996, Ph.D. dissertation.

[40] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, "Body area network security: a fuzzy attribute-based signcryption scheme," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 37–46, 2013.

[41] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," in *Advances in cryptology—EUROCRYPT 2003*, vol. 2656 of *Lecture Notes in Comput. Sci.*, pp. 255–271, Springer, Berlin, 2003.

[42] C. H. Tan, "Chosen ciphertext security from identity-based encryption without strong condition," in *Advances in information and computer security*, vol. 4266 of *Lecture Notes in Comput. Sci.*, pp. 292–307, Springer, Berlin, 2006.

[43] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO 2001*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, 2001.

[44] M. Bellare and P. Rogaway, "Random oracles are practical," in *Proceedings of the the 1st ACM conference*, pp. 62–73, Fairfax, Virginia, United States, November 1993.

[45] B. Waters, "Efficient identity-based encryption without random oracles," in *Advances in cryptology—EUROCRYPT 2005*, vol. 3494 of *Lecture Notes in Comput. Sci.*, pp. 114–127, Springer, Berlin, 2005.

[46] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Public Key Cryptography—PKC 2011*, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds., vol. 6571 of *Lecture Notes in Computer Science*, pp. 53–70, Springer, Berlin, Germany, 2011.

[47] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *Advances in cryptology—EUROCRYPT '97 (Konstanz)*, vol. 1233 of *Lecture Notes in Comput. Sci.*, pp. 256–266, Springer, Berlin, 1997.

[48] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography*, vol. 3378 of *Lecture Notes in Computer Science*, pp. 325–341, Springer, Berlin, Germany, 2005.

[49] D. M. Freeman, "Converting pairing-based cryptosystems from composite-order groups to prime-order groups," in *Advances in Cryptology—EUROCRYPT 2010*, vol. 6110 of *Lecture Notes in Computer Science*, pp. 44–61, Springer, Berlin, Heidelberg, Germany, 2010.

[50] J. A. Akinyele, C. Garman, I. Miers et al., "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

[51] L. B. Oliveira, D. F. Aranha, C. P. L. Gouvêa et al., "TinyPBC: pairings for authenticated identity-based non-interactive key distribution in sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 485–493, 2011.

[52] A. Miyaji, M. Nakabayashi, and S. Takano, "Characterization of Elliptic Curve Traces Under FR-Reduction," in *Information Security and Cryptology — ICISC 2000*, vol. 2015 of *Lecture Notes in Computer Science*, pp. 90–108, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[53] P. S. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in *Selected areas in cryptography*, vol. 3897 of *Lecture Notes in Comput. Sci.*, pp. 319–331, Springer, Berlin, 2006.

[54] X. Liang, R. Lu, X. Lin, and X. S. Shen, "Ciphertext policy attribute based encryption with efficient revocation," Tech. Rep., University of Waterloo, 2010.

[55] J. López and R. Dahab, "High-speed software multiplication in $textBbbF_{2^m}$," in *Progress in cryptology—INDOCRYPT 2000 (Calcutta)*, vol. 1977 of *Lecture Notes in Comput. Sci.*, pp. 203–212, Springer, Berlin, Germany, 2000.

[56] Z. Liu and Z. Cao, "On efficiently transferring the linear secret-sharing scheme matrix in ciphertext-policy attribute-based encryption , Cryptology ePrint Archive," Tech. Rep. 2010/374, Tech., 2010.