Practical and Scalable Access Control Mechanism for the Internet of Things

Clémentine Gritti[*], Emanuel Regnath[†], Sebastian Steinhorst[‡]

Abstract

Internet of Things (IoT) promises a strong world connecting digital and physical environments. Nevertheless, such a framework comes with huge security and privacy vulnerabilities, due to the heterogeneous nature of devices and of the diversity of their provenance. Other noticeable, technical challenges in IoT are brought with the constrained resources of devices, forcing to design protocol as lightweight as possible.

In this paper, we present a new system with access control key updates and direct user revocation, that are beneficial features in IoT. Access control is done using Ciphertext-Policy Attribute-Based Encryption where attributes represent roles of devices within their networks. Moreover, we devise a novel approach, based on a binary tree, to append time credentials. This allows us to find an interesting trade-off between key update frequency and user revocation list length, as well as stressing time-sensitive data exchanged in IoT environments. The security of our scheme is proved under the Decisional Bilinear Diffie-Hellman Exponent assumption.

Future work will focus on the implementation and analysis of our solution, in order to confirm that the latter is fully deployable in IoT networks.

Keywords. Ciphertext-Policy Attribute-Based Encryption, time-based key update, user revocation.

1 Introduction

New possibilities from the Internet of Things (IoT) technology are explored every day around the world. Complex combinations of hardware, sensors, data storage, microprocessors, software and ubiquitous connectivity are now included, moving beyond mechanical and electrical components. The mechanisms and tools for IoT offer better efficiency and productivity, but expand cyber vulnerabilities and threats along with technical challenges [4]. Devices forming IoT networks are heterogeneous in their functionality [27], come from various manufacturing origins, not always well defined, and have constrained computing and communication resources [31]. Moreover, these networks are dynamic, yielding the management even more demanding. 75 billion devices will be in the IoT world by 2025, and 127 new devices are connected every second to the Internet[1]. All of these characteristics make IoT dependability (i.e. reliability and availability) challenging [23].

Yet, other concerns come with the purposes of developing IoT, that is capitalizing fresh precious information. Indeed, IoT devices continuously collect and exchange a huge amount of data, that is combined and refined through data analytics, and the resulting

[*] University of Canterbury, New Zealand, clementine.gritti@canterbury.ac.nz

[†] Technical University of Munich, Germany, emanuel.regnath@tum.de

[‡] Technical University of Munich, Germany, sebastian.steinhorst@tum.de

[1] https://safeatlast.co/blog/iot-statistics/

information takes on real value[2]. Cisco believes that IoT will produce more than 500 zettabytes of data per year from 2020, and that number will grow exponentially[3]. In addition, to improve the accuracy of IoT systems, efforts must be made on data sharing. The main drawback is the raise of security and privacy menaces [1, 19, 17].

In this paper, we are interested in developing an efficient access control system for secure data exchanges in IoT networks. Access control with identity management and authentication ensures that only authorized users are able to reach data. We aim to design a solution that takes into account data sharing concerns while overcoming IoT dependability issues. The extremely large number of IoT devices and the dynamicity of IoT networks force to go beyond basic identity assignment techniques as for Public Key infrastructure [2]. Another issue comes with trivial key management where each device either receives a public/private key pair or shares a secret key with another device; in both cases the device should maintain a substantial number of keys in order to interact with other devices. Moreover, such techniques imply a centralized architecture, raising the single point failure problem with unpredictable threats. Due to their ubiquity combined with the high configuration vulnerability, IoT devices have been involved in many cyber attacks [26, 18]. Therefore, revocation must be an essential option when elaborating a system. Then, it has been very important to achieve low latency and high reliability for many IoT use cases [30]. Devices collect time-sensitive data in various situations, where either data batch processing would produce results too late to be useful or any application where latency is a concern. For instance, some control decisions in autonomous vehicles require sub-microsecond response times. Industrial control systems require response in tens of microseconds to avoid damage and ensure safety. Temperature sensors must collect data once every few minutes and respond within a second. Electric metering requires frequent communication, low latency and high data rate [24]. Hence, an access control system should consider time as an essential feature to meet the aforementioned requirements.

This work introduces a fine grained access control scheme based on Attribute-Based Encryption (ABE), which remains lightweight and hence deployable in IoT networks. We design our system with key updates for access control and device revocation to overcome the aforementioned IoT security vulnerabilities. First, an access control based on roles permits to share collected data securely following the dynamicity of IoT networks. Devices are seen as users in our system, either encrypting data (owners) and decrypting it (requesters). Second, we encourage the participation of multiple authorities in charge of distributing key material to users based on their roles within authorities' environments, averting single point of failure. Third, we enable direct user revocation, thus always protecting sensitive data even if a user secret key is compromised. Then, we append time credentials in addition to role ones, emphasizing the ephemeral value of shared data while enabling an interesting trade-off between reasonable key update frequency and moderate user revocation list length. Thus, our solution does not require recurrent communication between users and authorities and deletion of components to expunge existing keys to produce new keys. Our approach efficiently integrates a role and time based access control scheme with IoT technologies, such that the outcome is fully implementable in the real world. We carefully prove the security of our scheme under the Decisional Bilinear Diffie-Hellman Exponent (BDHE) assumption. Moreover, we observe from the implementation

[2] https://blog.equinix.com/blog/2018/02/21/the-rise-of-iot-data-exchanges/

[3] https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html

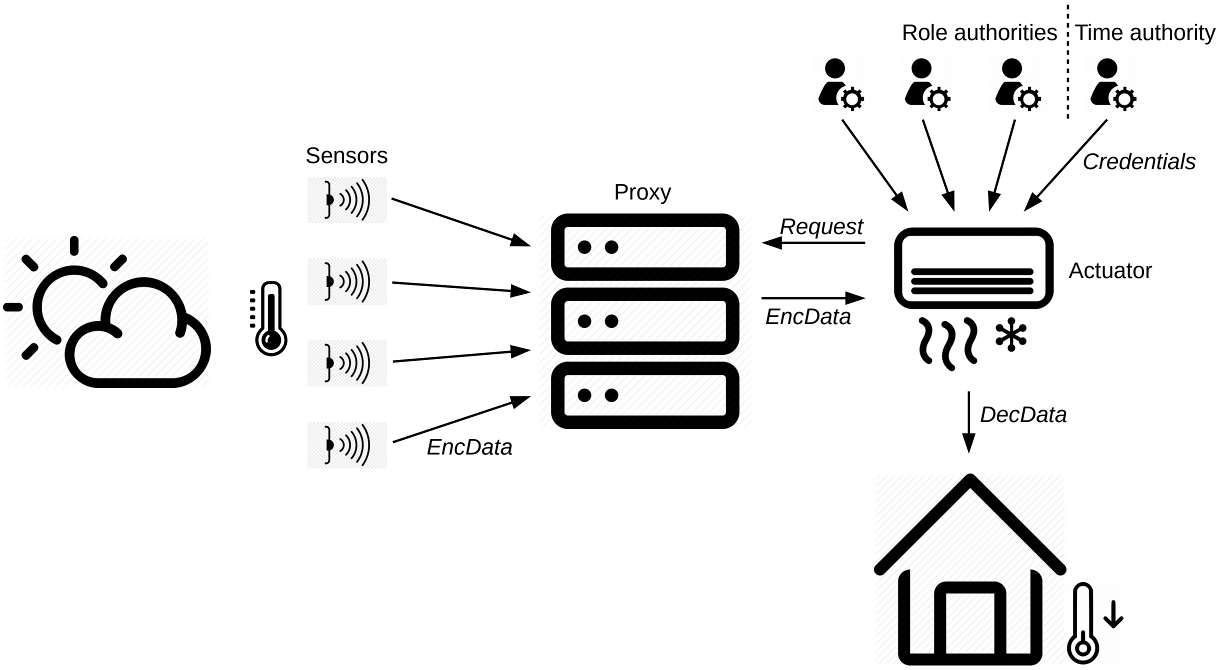

Figure 1: Our IoT scenario takes place in a smart home, where multiple temperature sensors are scattered and an actuator adjusts temperature in response to sensors' collected data.

of our system that the computational and communication results make it adjustable in IoT environments.

Figure 1 illustrates an example of our access control system in a smart home. Several temperature sensors are scattered in a house. They collect temperature data once every few minutes. They encrypt their time-sensitive data according to an access policy, containing roles and time periods. There is also an actuator (possibly indirectly, via a gateway for example) connected to these sensors. The actuator has received role and time credentials from multiple authorities.

Since sensors have limited storage capacity, we suppose that they upload their encrypted data to a proxy (e.g. a cloud server). Within the rest of the paper, we assume that the proxy exists and is intimately linked to sensors, hence we omit to mention it explicitly. This proxy plays the role of an intermediary between the sensors and actuator. The latter sends requests to the proxy for access to sensors' data every short time intervals, of the order of minutes. The proxy replies to the actuator's requests by forwarding the encrypted collected data. The actuator is able to recover the data in plan if and only if it has been granted with credentials satisfying sensors' access policies. By having the plain data, the actuator adjusts the temperature accordingly.

1.1 Related Work

Attribute-Based Encryption. Identity-Based Encryption (IBE) [32, 11, 33] is a public-key cryptographic primitive that uses some unique information about the identity of a user (e.g. the email address) as the public key of that user. The corresponding secret key is generated by a trusted authority, based on the public key.

Attribute-Based Encryption (ABE) [8, 16, 34] is a variant of IBE (first called Fuzzy IBE [29]). Now, the secret key of a user and the ciphertext are dependent upon attributes (e.g. the country of living, the position within the company, etc.). The decryption of a ciphertext is possible if and only if the attributes of the key match the attributes of the ciphertext.

There are two types of ABE schemes. In the first one, called Key-Policy ABE (KP-ABE), the user secret key is linked to an access policy and the ciphertext is associated with an attribute set, such that the attributes in that set should satisfy the policy to get successful decryption. On the opposite, in a Ciphertext-Policy ABE (CP-ABE) scheme, the user secret key is associated with an attribute set while the ciphertext is linked to the access policy. Decryption works as above.

Extended Attribute-Based Encryption. ABE has been subjected to many extensions, by including extra features while keeping security. Here, we review solutions with the features we are interested in, namely user revocation and multiple-authority setting.

Yang and Jia [35] present a multi-authority CP-ABE scheme that embeds a revocation mechanism with forward and backward securities. In this scheme, each authority has its own attribute universe, and generates keys for users according to their attributes in that universe. However, one root authority is still required to generate the secret key material for each attribute authority, hence keeping the scheme prone to single point of failure. Revoking a user is made possible by revoking one attribute granted to that user. Updating existing ciphertexts according to newly revoked users is delegated to a cloud server, thus alleviating the workload on the side of the user who generated them. Nevertheless, attribute revocation requires to update the secret keys of non-revoked users. Moreover, user revocation is decided by the authorities rather than the user who owns and has encrypted the data.

Liu et al. [21] propose to combine ABE and Time-based Proxy Re-Encryption to enable a fine-grained access control on encrypted data and scalable user revocation, while the data owner can remain offline. Revoking users is done by using time attributes. Users are given keys embedding role attributes as well as time attributes. The data owner encrypts the data according to an access policy; at this stage, time control is not enabled. Ciphertexts are uploaded to the cloud server (proxy), who updates the ciphertexts with the current time when users request data (like in a Proxy Re-Encryption scheme). If a user is not allowed to retrieve the data at the time of the request (by lack of adequate time attributes), then decryption fails. If a user has a key with the time attributes still available when requesting the data, then this user successfully decrypts. Unfortunately, the time control structure is cumbersome and not adaptable with time intervals but only with discrete timing. In addition, the data owner and cloud server must share a root secret key. While such key does not help the cloud server to obtain information on the data, it permits to re-encrypt ciphertexts with the current time, implying strong trust assumptions on that cloud server. Also, the data owner is responsible for generating the secret keys of users, such that she decides time validity for them. Hence, the data owner should be fully trusted, while in general, data owners are also users requesting other data. Therefore, some misconducts can easily happen among colluding malicious users, making the system vulnerable. Practicality also suffers from such design: in a system with N users, all of them being owners of some data, each user needs $N - 1$ keys generated by others.

Liu et al. [20] combine CP-ABE with a direct revocation approach (i.e. the most recent list of revoked users is always included in the ciphertext) and Hierarchical Identity-

Based Encryption for time period control (i.e. a tree-based mechanism). In order to avoid the revocation list growing too much as time goes by, each user obtains a key with an embedded validity time range. The users have then keys that expire on a date and one would only appear in the revocation list if she has been revoked before her key's expiration date (e.g. her key has been stolen before expiration). After key expiration, the name of the revoked user is discarded from the revocation list and a new user key is generated. Time periods are defined as a trade-off between a revocation list with reasonable length and a moderate frequency of key update. The key size depends on the validity time ranges assigned to users and on the maximum number of revoked users in the list, hence can dramatically grow. Moreover, there is a unique role authority in charge of generating user keys, promoting single point of failure. Symmetric pairings from cyclic groups of prime order are used, making the scheme less efficient and secure than using asymmetric pairings. Our scheme keeps the positive features of the LYZL scheme, namely direct revocation with the list embedded into ciphertexts and time access control with a tree-based mechanism. However, we extend the solution by appending a more robust multi-authority setting, better security and performance by using asymmetric pairings, and an improved time framework. In the rest of the paper, we refer to the scheme from [20] as the LYZL scheme.

Attribute-Based Encryption in IoT. Yao et al. [36] present a new ABE scheme based on elliptic curve cryptography and without any pairing operation. Such features enable to obtain a lightweight and secure access control protocol in IoT networks. The authors also analyze the communication and computational costs induced by their solution and observe a significant gain in terms of practicality and efficiency over original schemes [8, 34]. While our scheme still requires pairing operations, we opt for asymmetric ones over elliptic curves (rather than symmetric ones), improving security and efficiency [15]. Moreover, contrary to us, no multi-authority and revocation mechanisms are implemented in Yao et al.'s scheme [36].

Oualha and Nguyen [25] propose an access control mechanism based on CP-ABE by considering the large number of devices in IoT networks and their constrained resources. Specifically, the authors apply pre-computations techniques [12] to Bethencourt et al.'s CP-ABE [8] to reduce the computational costs induced for data encryption, that is performed by IoT devices. However, the latter require more storage space since they must retain pre-computed tuples, that do not exist in the original scheme [8]. In addition, an extra trusted authority is required to generate these pre-computed values, and a secure channel is needed between this authority and the devices to transmit them. While encryption is made computationally easier for IoT devices, nothing is said about the rest of the access control protocol, namely secret key generation and storage, as well as decryption.

Meanwhile, Ambrosin et al. [5] study the feasibility of Bethencourt et al.'s CP-ABE [8] on widely used IoT-enabling devices. The authors focus on the evaluation of encryption and decryption steps, and test these cryptographic operations on four existing IoT platforms. Their results show that CP-ABE can be adopted in IoT environments without major flaw. The authors also present a successful use case application in smart healthcare using Bethencourt et al.'s CP-ABE [8]. While Oualha and Nguyen [25] show a technique to relieve computational workload on devices' side when implementing Bethencourt et al.'s CP-ABE [8], Ambrosin et al. [5] demonstrate that this CP-ABE scheme is fully adoptable in its original version. Results from Ambrosin et al. [5] suggest us that most of existing ABE schemes can be implemented in IoT systems.

1.2 Contributions

We propose a new ABE scheme for role and time access control and user revocation, that offers the following features:

1. the participation of multiple authorities to avoid a single point of failure in the system;

2. a direct approach for user revocation to limit damages from compromised user secret keys;

3. a user control using role attributes as well as time validity ranges as a trade-off between key update frequency and revocation list length.

We prove the security of our scheme under the Decisional Bilinear Diffie-Hellman Exponent (BDHE) assumption. We also implement our solution on to be defined and observe that the results make our solution adjustable in IoT environments, where computing, communication and storage resources are highly limited.

1.3 Road Map

In the following section, we define the building blocks and tools required for our solution. In Section 3, we present our CP-ABE scheme with multiple authority setting, direct revocation and time-based access control mechanism, along with its security. In Section 4, we implement and analyze our solution. Finally, in Section 5, we conclude the paper.

2 Preliminaries

2.1 Building Blocks

Multiple authorities. We propose to enhance the LYZL scheme in [20] by involving multiple authorities. In [20], one authority, fully trusted, is in charge of setting up the system and generating the key material of users. Such configuration may be subject to single point of failure. By enabling the user public and secret parameter generation among several authorities, we reduce trust assumptions made on these authorities while enforcing the security of the scheme.

Revocation. We follow the methodology proposed in [20], where revocation is done by making the secret key of a user unusable. The term "user" refers to a device in our system. The reasons can be diverse:

1. The user has left its IoT network and hence the key should no longer be usable. For instance, the owner of the temperature sensor has disconnected it from the smart home network.

2. The user has lost its key and been attributed a new one, hence the old key should no longer be usable. For example, a misuse of the IoT device by its owner has triggered some complications, such as key loss. When rebooting the device, a new key has been generated.

3. The user has one of its attributes changed and thus has received a new key with this new attribute, and the old key should no longer be usable. For instance, one of the device's attributes has been modified from "everyone" to "adult" when some parental controls have been put in place.

Various approaches for revocation exist, such as key update for non-revoked users and cloud assistance. However, the first solution does not allow instant user revocation while the second one encounters practical issues when the number of users becomes huge.

A more interesting approach permits to revoke users by appending the identity of this user in the revocation list. The list is public, instantly updated and included in each ciphertext in its latest version. Only the users not in the revocation list and with the attributes satisfying the access policy are able to decrypt the ciphertext. The main advantage is that key update is not necessary, avoiding extra communication and computational burdens. Nevertheless, the number of users in that list grows with the time. If the number of users involved in the system is huge, then this setting becomes a practical issue. An alternative is to create a non-revocation list that includes identities of non-revoked users. Hence, the length of this list will decrease over the time. However, we claim that the number of non-revoked devices is much larger than the number of revoked ones in an IoT system, making the non-revocation list difficult to handle.

The revocation mechanism presented in [20] is a trade-off between two techniques, namely appending the revocation list to ciphertexts and updating user keys based on time intervals. Users are given keys embedding their role attributes as well as their time validity ranges. The latter define a time period with an expiry date from which users are no longer authorized to access any data. Therefore, user keys are updated after the expiry date, such that the time interval between two updates should remain reasonable. Moreover, if a user is revoked before its key expires, then its identity is added into the revocation list and kept in it until the next key update. Then, a new key is generated according to role attributes and a new time validity range. We emphasize that key update is made possible but not mandatory (e.g. a user may be revoked definitely from the network). In addition, the generation of a key after its expiration may incur new attributes or discard used ones (e.g. a temperature sensor system has evolved and includes new functionalities, hence new attributes). We let the reader to refer to the exhaustive literature review on revocation in ABE in [20].

Role attributes. In the LYZL scheme [20], an attribute universe is associated with the single authority, such that attributes are all different.

In our solution, each role authority has its own attribute universe, such that the union of all the attribute universes forms the whole universe. We assume that attribute universes are all disjoint by defining attributes as follows: Let a role be “temperature” and two authorities refer to “Room A” and “Room B” respectively. Hence, the two attributes are determined uniquely as “RoomA||temperature” and “RoomB||temperature” respectively. Such appellation enables to obtain a whole universe with distinct attributes, as wished.

In the rest of the paper, we denote $\mathcal{U}_k$ the attribute universe associated with the role authority A_k. Let $\mathcal{U} = \cup_{A_k} \mathcal{U}_k$ be the disjoint union of all authorities’ universes.

Role attribution management is thus taken by multiple authorities in our system. They are responsible to define role attributes in their respective universes, and assign them to users when generating their keys. Role key updates do not require to be frequent, since roles such as “temperature” should remain forever for a temperature sensor. Key updates are rather required to refresh the revocation list once it reaches the maximum number of revoked users.

Time attributes. The methodology proposed in [20] determines time intervals as days, months and years. The LYZL scheme supports both continuous and non-continuous time intervals; however, authors suggest that their method is only interesting in the case of continuous ones. The user encrypting the data defines a decryption time period such that only users with time credentials completely covering that time period can decrypt. For instance, if a user has time credential “15 January 2020” while the encryptor has set “January 2020” for decryption, then the former cannot decrypt since the credential does not completely cover the decryption time period. On the other side, if a user has time credential “January 2020” while the encryptor has set “from 01 to 15 January 2020” for decryption, then the former can successfully decrypt since it completely covers the decryption time period. Such properties are kept in mind when designing our time-based access control solution.

Authors in [20] suggest to use the Hierarchical Identity-Based Encryption scheme from [10] to create time validity control. Such tree-based approach allows improvement on the efficiency for continuous time intervals (claiming that user keys and ciphertexts are usually represented as time intervals). Then, a set cover approach is used to select the minimum number of nodes that represent all the valid time periods. Each node, except the root one, accounts for a time period such that leaves are days, leaves’ parents are months and leaves’ grand-parents are years. The root node is implicitly set as a starting time. Liu et al. suggest that a 2-year interval between two key updates is reasonable, and thus the tree is constructed based on two consecutive years, for instance 2020 and 2021. Therefore, the starting time is “01 January 2020” and the tree represents time until “31 December 2021”.

Let T be the depth of the tree and each node has z children. The time is thus represented as a z-ary string $\{1, 2, \cdots, z\}^{T-1}$ and a time period is denoted with a z-ary element $(\tau_1, \tau_2, \cdots, \tau_\eta)$ for some $\eta < T$. No numerical value is given throughout the paper [20]; but we propose to make some assumptions from the reading. As mentioned above, the authors suggest that a 2-year interval between two key updates is reasonable and time periods based on year, month and day are enough for their purposes (but can be extended to minute and second). Moreover, in order to simplify the description of the tree structure, each node is supposed to have z children. From there, we infer that $T = 4$, and z is common to all non-leaf nodes and set to be equal to 31 (there are at most 31 days in a month). The latter assumption implies that the root has $z = 31$ children, and nodes representing years have also $z = 31$ nodes, even if 2-year intervals are examined and 12 months form a year. Such simplification approach causes the tree construction process to be more cumbersome with $31 - 2 = 29$ dummy nodes for years, $29 * (31 - 12) = 551$ dummy nodes for months and $29 * 19 * 31 = 17081$ dummy nodes for days.

Few ideas from the tree-based structure will be kept for our system. We also opt for a tree to represent the time framework with the root implicitly embedding a starting time and leaf nodes denoting days. We now explore the differences from the tree-based method in [20] and ours:

- In the LYZL scheme [20], the initialization algorithm solely generates the parameters of the system and single authority. Since we involve multiple authorities, several algorithms are required to generate the common system parameters and the specific parameters of each authority. In particular, the authority responsible of time control creates the public and secret parameters required for the time tree.

- We choose a binary structure rather than a z-ary one. Therefore, each node has two children. We hence avoid numerous dummy nodes from choosing the maximum value among number of years, number of months and number of days.

- We focus on shorter time periods according to our IoT-based time-sensitive data scenario. The number of leaf nodes (of the form 2^i for some integer i) defines the

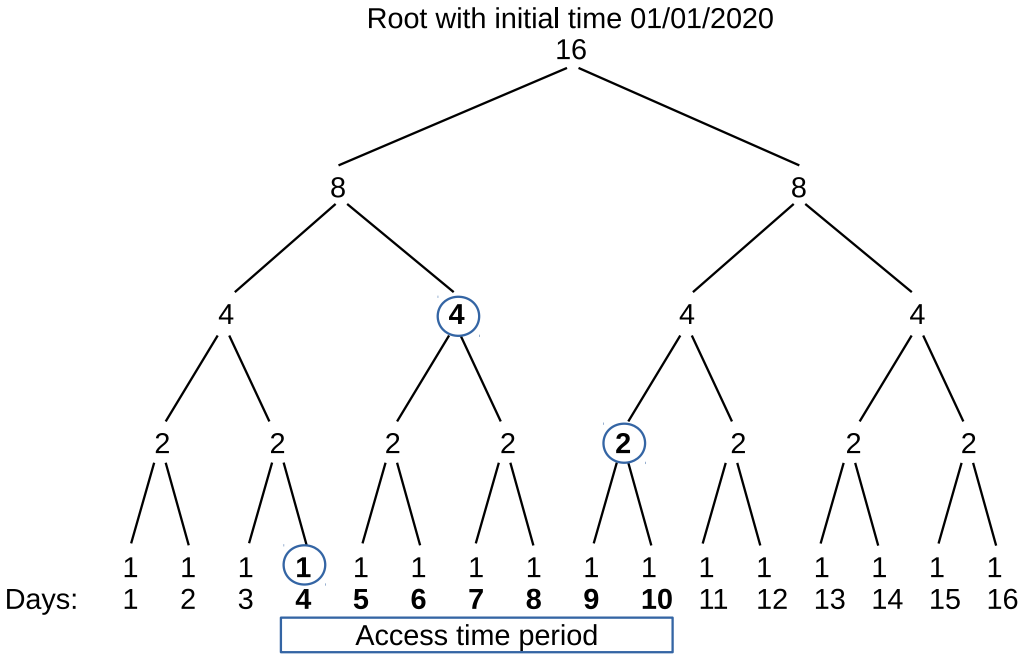
Root with initial time 01/01/2020
16
8
8
4
4
4
4
2
2
2
2
2
2
2
2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Days: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Access time period

Figure 2: Time tree from "01 January 2020" until "16 January 2020" (included). Access time period is given for 7 days, from "04 January 2020" until "10 January 2020" (included): three keys corresponding to nodes with blue-line circles are then generated.

time interval between two key updates. Therefore, in order to keep the tree with a reasonable depth T, that number must be relatively small.

- Following our binary structure, a path from the root to a node is denoted as a string in $\{0,1\}^{T-1}$ where 0 denotes the left child and 1 denotes the right child of a given node.
- To construct a tree, one needs to choose a starting time (defining the root) and the number of days between two key updates. That number correspond to the one of leaf nodes. Then, from the bottom level, we build the tree up to the root.

Figure 2 illustrates a time tree following our methodology. The tree has depth $T = 5$, resulting into 16 leaf nodes, one for exactly one day. The root embeds the starting time "01 January 2020". Therefore, the time interval starts on "01 January 2020" and ends on "16 January 2020" (included). In our tree example, a user receives time key material for a time validity range of 7 days, starting 4 days after the starting time. This means that the user has been granted for a period from "04 January 2020" until "10 January 2020" (included). The user is given three key components as illustrated by blue circles in Figure 2: one for the leaf node representing day 4, one for the grand-parent of leaf nodes from day 5 until day 8 and for the parent of leaf nodes for days 9 and 10.

Time management is taken by a dedicated time authority. The latter is responsible to define time trees for the system and assign time validity ranges to users when generating their keys. Time key updates are frequent, of the order of several days, due to our IoT-based time-sensitive data scenario.

Asymmetric bilinear pairings. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be three cyclic groups of prime order p. A pairing e is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ which satisfies the following properties:

- Bilinearity: Given $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$;
- Non-degeneracy: There exist $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ such that $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$;
- Computability: There exists an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$.

If the same group is used for the first two groups, meaning that $\mathbb{G}_1 = \mathbb{G}_2$, the pairing is called *symmetric* and is a mapping from two elements of one group to an element from a second group. Such setting is used in [20]. Otherwise, meaning that $\mathbb{G}_1 \neq \mathbb{G}_2$, the pairing is called *asymmetric*. In this case, either there is an efficiently computable homomorphism $\phi : \mathbb{G}_1 \to \mathbb{G}_2$ or there are no efficiently computable homomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$ [14].

It has be shown that designing a scheme in an asymmetric bilinear pairing setting rather than a symmetric one enables a better efficiency as well as an improved security level [15, 9]. Therefore, our solution extends the LYZL scheme [20], set with symmetric bilinear pairing, to permit an asymmetric pairing setting.

2.2 Miscellaneous.

Vector. Let $\vec{v} = (v_1, \cdots, v_R)$ be a vector in $\mathbb{Z}_p^R$ for an integer R. Let $g_1^{\vec{v}} = (g_1^{v_1}, \cdots, g_1^{v_R})^\top$ be a column vector in $\mathbb{G}_1$. Given $\vec{v}, \vec{w}$, let the product $\langle \vec{v}, \vec{w} \rangle$ be equal to $\vec{v}^\top \vec{w} = \sum_{i=1}^{R} v_i w_i$ and $(g_1^{\vec{v}})^{\vec{w}}$ be $g_1^{\langle \vec{v}, \vec{w} \rangle}$.

Bilinear group. Given as input a security parameter 1^λ, the algorithm Gen outputs the tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are multiplicative cyclic groups of prime order p and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a pairing.

Access structure [7]. Let $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ be a set of parties. A collection $\mathbb{C} \subseteq 2^{\mathcal{P}}$ is said to be monotone if for all A, B, if $A \in \mathbb{C}$ and $A \subseteq B$ then $B \in \mathbb{C}$. An access structure is a collection $\mathbb{C} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. The sets in (resp. not in) $\mathbb{C}$ are said to be *authorized* (resp. *unauthorized*).

Linear secret sharing scheme [7]. A Secret Sharing Scheme (SSS) Π over a set of parties $\mathcal{P}$ is called Linear (and denoted LSSS) if the following conditions hold:

- The shares of the parties form a vector over $\mathbb{Z}_p$;
- There are a $l \times \nu$ matrix M and a function ρ that maps the i-th row, for $i \in [1, l]$, to an associated party $\rho(i)$. Let $s \in \mathbb{Z}_p$ be a secret to be shared, and $\gamma_2, \cdots, \gamma_\nu$ be random exponents from $\mathbb{Z}_p$. Let $\vec{v} = (s, \gamma_2, \cdots, \gamma_\nu)$ be a column vector and $M\vec{v}$ be the vector of l shares of the secret s according to Π such that the share $(M\vec{v})_i$ belongs to party $\rho(i)$.

We now define the linear reconstruction property: Let Π be an LSSS for an access structure $\mathbb{C}$, $S \in \mathbb{C}$ be an authorized set and $I = \{i; \rho(i) \in S\} \subset [1, l]$. There exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. The constants ω_i can be found in time polynomial in the size of M. Moreover, for any unauthorized set $S \notin \mathbb{C}$, the secret s should be information theoretically hidden from the parties in S.

Decisional q-BDHE assumption. Given $\vec{P} = (g_1, g_1^s, g_1^a, \cdots, g_1^{a^q}, g_1^{a^{q+2}}, \cdots, g_1^{a^{2q}}, g_2, g_2^s, g_2^a, \cdots, g_2^{a^q}, g_2^{a^{q+2}}, \cdots, g_2^{a^{2q}}) \in \mathbb{G}_1^{2q+1} \times \mathbb{G}_2^{2q+1}$ and $Q \in \mathbb{G}_T$, where $s, a \in \mathbb{Z}_p$, $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, the Decisional q-Bilinear Diffie-Hellman Exponent (BDHE) problem is defined as to decide whether $Q = e(g_1, g_2)^{sa^{q+1}}$ or a random element in $\mathbb{G}_T$.

Indexing and implementing role attributes. With a correct index assignment, we ensure that one index exactly corresponds to one attribute. All role attributes are unique since they are defined according to a specific role authority representing an IoT environment, and determine a role within that environment. Then, we assign the indices for role attributes as follows: Let N be the number of role authorities and A_k be the role authority with universe $\mathcal{U}_k$ containing U_k attributes, for $k \in [1, N]$. Then, indices for attributes in the universe $\mathcal{U}_k$ associated with authority A_k are $(\sum_{j=1}^{k-1} U_j + 1), \cdots, (\sum_{j=1}^{k-1} U_j + U_k)$. To simplify the reading with indices, let $k||i = (\sum_{j=1}^{k-1} U_j + i)$ for $i \in [1, U_k]$.

In addition, let $I = \{i; \rho(i) \in S\} \subseteq [1, l]$ be defined as above, and $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be the set of constants such that if the set $\{\lambda_i\}$ contains valid shares of a value s according to the matrix M, then $\sum_{i \in I} \omega_i \lambda_i = s$. Let $\mathcal{A}$ be the set of role authorities whose attributes are in the access policy (i.e. the access structure). Let $\pi : k \to \pi(i)$ be defined as $\exists!(A_k \in \mathcal{A}, j \in [1, U_k])$ such that $\rho(i) = k||j$. Such surjective function exists since each attribute is defined uniquely in the whole universe $\mathcal{U} = \cup_{A_k} \mathcal{U}_k$.

As mentioned above, an attribute in the whole universe $\mathcal{U}$ is uniquely controlled by one authority A_k. In order to explain the functionality of the function π and to make it implementable, let us assume that there exists a publicly computable function $F_\pi : \mathcal{U} \to \mathcal{A}_k$ that maps one attribute to a specific role authority [28]. From this mapping, let a second labeling of rows be defined in the access structure $((M, \rho), \rho')$ such that it maps rows to attributes via the function $\rho(\cdot) = F_\pi(\rho'(\cdot))$.

3 A new Multi-Authority Time-Based Revocable Ciphertext-Policy Attribute-Based Encryption

3.1 Construction

Our solution contains seven algorithms, defining three phases. First, an initialization phase sets up the system. Public parameters are generated and made available to authorities and users. Then, the authorities generate their public and secret key material. That phase is run only once.

Second, the authorities create the key material for users. There are N role authorities, who are responsible of creating keys based on user roles within their respective environments. Role key updates for non-revoked users are run occasionally, say every two years. There is one time authority that generates user keys based on time validity ranges. This authority frequently updates such key material based on new time validity ranges, say every month.

Third, an encryptor chooses an access policy based on both role attributes and time period, and encrypts some data according to that policy. A user which has been granted with role and time credentials satisfying the access policy can successfully decrypt the ciphertext and recover the data.

We now give the construction of our Multi-Authority Time-Based Revocable Ciphertext-Policy Attribute-Based Encryption scheme:

$\mathsf{Setup}(1^\zeta, R)$. Let $R - 1$ be the maximum number of revoked users. On inputs the security parameter 1^ζ and R, the algorithm Setup outputs the public parameters PP.

First, run the algorithm Gen and obtain two bilinear groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order p with generators g_1 and g_2 respectively, along with a third group $\mathbb{G}_T$ of prime order p and a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Pick at random $\delta, \alpha_1, \cdots, \alpha_R \in_R \mathbb{Z}_p$ Set $\vec{\alpha} = (\alpha_1, \cdots, \alpha_R)^\top$ and $\vec{F} = g_1^{\vec{\alpha}} = (g_1^{\alpha_1}, \cdots, g_1^{\alpha_R})^\top = (f_1, \cdots, f_R)^\top$. The public parameters are $PP = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(g_1, g_2), g_1^\delta, \vec{F})$.

$\mathsf{RAKeyGen}(PP, U_k)$**.** Let U_k be the number of role attributes in the universe $\mathcal{U}_k$ associated with role authority A_k. On inputs the public parameters PP and U_k, the algorithm RAKeyGen outputs the public key PK_k and the secret key SK_k of the role authority A_k.

Pick at random $\kappa_k \in_R \mathbb{Z}_p$ and $h_{k||1}, \cdots, h_{k||U_k} \in_R \mathbb{G}_1$ (these elements $h_{k||i}$ will be used for role access control with relation to the authority A_k). The public key is $PK_k = (e(g_1, g_2)^{\kappa_k}, h_{k||1}, \cdots, h_{k||U_k})$ and the secret key is $SK_k = \kappa_k$.

$\mathsf{TAKeyGen}(PP, T)$**.** Let T be the depth of the time binary tree associated with time authority B. The time is represented as a binary string $\{0,1\}^{T-1}$. On inputs the public parameters PP and T, the algorithm TAKeyGen outputs the public key PK and the secret key SK of the time authority B.

Pick at random $\sigma \in_R \mathbb{Z}_p$ and $V_0, V_1, \cdots, V_T \in_R \mathbb{G}_1$ (these elements V_j will be used for time access control with relation to the time authority B). The public key is $PK = (e(g_1, g_2)^\sigma, V_0, V_1, \cdots, V_T)$ and the secret key is $SK = \sigma$.

$\mathsf{RUKeyGen}(PP, (PK_k, SK_k), ID, S_{ID,k})$**.** Let $S_{ID,k}$ be the role attribute set of a user with identity ID and associated with role authority A_k. Let $k||x \in S_{ID,k}$ denote the attribute uniquely defined in the whole universe $\mathcal{U} = \cup_{A_k} \mathcal{U}_k$ by determining the associated authority A_k and the role x within $\mathcal{U}_k$. On inputs the public parameters PP, the public and secret keys PK_k and SK_k of the role authority A_k, ID and $S_{ID,k}$, the algorithm RUKeyGen outputs the secret key $RSK_{ID,k}$ of the user with identity ID, role attribute set $S_{ID,k}$ and associated with authority A_k.

First, pick at random $u_k, t_k \in_R \mathbb{Z}_p$. Then, compute the following:

$$\begin{aligned} D_{k,0} &= g_2^{t_k} \\ D'_{k,0} &= g_2^{u_k} \\ D_{k,1} &= g_1^{\kappa_k} g_1^{\delta t_k} f_1^{u_k} = g_1^{\kappa_k} g_1^{\delta t_k} g_1^{\alpha_1 u_k} \\ K_{k,x} &= h_{k||x}^{t_k} \text{ for } k||x \in S_{ID,k} \\ F_{k,i} &= (f_1^{-ID^{i-1}} f_i)^{u_k} \text{ for } i \in [2, R] \end{aligned}$$

The secret key is $RSK_{ID,k} = (D_{k,0}, D'_{k,0}, D_{k,1}, \{K_{k,x}\}_{k||x \in S_{ID,k}}, \{F_{k,i}\}_{i \in [2,R]})$ and includes a description of $S_{ID,k}$.

$\mathsf{TUKeyGen}(PP, (PK, SK), ID, T_{ID})$**.** Let T_{ID} be the time validity range of the user with identity ID and associated with time authority B. On inputs the public parameters PP, the public and secret keys PK and SK of the time authority B, ID and T_{ID}, the algorithm TUKeyGen outputs the secret key TSK_{ID} of the user with identity ID, time validity range T_{ID} and associated with authority B.

Let $\mathbb{T}$ be the set cover representing T_{ID} which consists of time elements $\tau = (\tau_1, \cdots, \tau_{\eta_\tau}) \in \{0,1\}^{\eta_\tau}$ where $\eta_\tau < T$ for any $\tau \in \mathbb{T}$. First, pick at random $\beta, v_\tau \in_R \mathbb{Z}_p$ for $\tau \in \mathbb{T}$.

Then, compute the following:

$$
\begin{aligned}
D_{0,\tau} &= g_2^{v_\tau} \text{ for } \tau \in \mathbb{T} \\
D_{1,\tau} &= g_1^{\sigma} f_1^{\beta} (V_0 \prod_{j=1}^{\eta_\tau} V_j^{\tau_j})^{v_\tau} = g_1^{\sigma} g_1^{\alpha_1 \beta} (V_0 \prod_{j=1}^{\eta_\tau} V_j^{\tau_j})^{v_\tau} \text{ for } \tau \in \mathbb{T} \\
D_2 &= g_2^{\beta} \\
L_{j,\tau} &= V_j^{v_\tau} \text{ for } j \in [\eta_\tau + 1, T] \text{ and } \tau \in \mathbb{T} \\
E_i &= (f_1^{-ID^{i-1}} f_i)^{\beta} \text{ for } i \in [2, R]
\end{aligned}
$$

The secret key is $TSK_{ID} = (\{D_{0,\tau}, D_{1,\tau}\}_{\tau \in \mathbb{T}}, D_2, \{L_{j,\tau}\}_{j \in [\eta_\tau+1,T],\, \tau \in \mathbb{T}}, \{E_i\}_{i \in [2,R]})$ and includes a description of T_{ID}.

Encrypt$(PP, \{PK_k\}_{A_k \in \mathcal{A}}, PK, m, \mathcal{R}, (M, \rho), T_{dec})$**.** Let $\mathcal{A}$ be the set of role authorities whose role attributes are in the access policy. Let m be the message to be encrypted. Let $\mathcal{R} = (ID_1, \cdots, ID_r)$ be the revocation list containing $r < R$ revoked users. Let (M, ρ) be an LSSS access structure, defining the role access policy, where M is a $l \times \nu$ matrix and the function ρ associates rows of the matrix M to role attributes. Let T_{dec} be the decryption time period of the ciphertext. On inputs the public parameters PP, the public keys PK_k of the role authorities A_k in $\mathcal{A}$, the public key PK of the time authority B, m, $\mathcal{R}$, (M, ρ) and T_{dec}, the algorithm Encrypt outputs a ciphertext CT.

Let $\tau_{dec} = (\tau_1, \cdots, \tau_{\eta_{dec}}) \in \{0,1\}^{\eta_{dec}}$ be the binary representation of T_{dec}, where $\eta_{dec} < T$. First, choose a secret s from $\mathbb{Z}_p$ and pick at random $\gamma_2, \cdots, \gamma_\nu \in_R \mathbb{Z}_p$. Set the vector $\vec{v} = (s, \gamma_2, \cdots, \gamma_\nu)$. Then, for $i \in [1, l]$, compute $\lambda_i = \langle \vec{v}, M_i \rangle$, where M_i is the i-th row of M. Let $\mathcal{F}_{\mathcal{R}}(Z) = (Z - ID_1) \cdot (Z - ID_2) \cdots (Z - ID_r) = y_1 + y_2 Z + \cdots + y_r Z^{r-1} + y_{r+1} Z^r$. If $r + 1 < R$, then set the coefficients $y_{r+2}, \cdots, y_R$ equal to 0. Then, compute the following:

$$
\begin{aligned}
C_0 &= m \cdot e(g_1, g_2)^{\sigma s} \cdot \prod_{A_k \in \mathcal{A}} e(g_1, g_2)^{\kappa_k s} \\
C_0' &= g_2^s \\
C_0'' &= (f_1^{y_1} \cdots f_R^{y_R})^s \\
C_0''' &= (V_0 \prod_{j=1}^{\eta_{dec}} V_j^{\tau_j})^s \\
C_i &= g_1^{\delta \lambda_i} h_{\rho(i)}^{-s} \text{ for } i \in [1, l]
\end{aligned}
$$

The ciphertext is $CT = (C_0, C_0', C_0'', C_0''', \{C_i\}_{i \in [1,l]}, (M, \rho))$ and includes descriptions of T_{dec}, $\mathcal{A}$ and $\mathcal{R}$.

Decrypt$(PP, CT, \mathcal{R}, \{RSK_{ID,k}\}_{A_k \in \mathcal{A}}, TSK_{ID})$**.** On inputs the public parameters PP, the ciphertext CT, the revocation list $\mathcal{R}$, the role secret keys $RSK_{ID,k}$ of user with identity ID and associated with $A_k \in \mathcal{A}$ and the time secret key TSK_{ID} of user with identity ID and associated with B, the algorithm Decrypt outputs either the message m or a null sign $\perp$.

Let $\vec{X} = (1, ID, \cdots, ID^{R-1})$ for the identity ID and $\vec{Y} = (y_1, \cdots, y_R)$, where the exponents y_i have been defined during the encryption phase. Hence, $\langle \vec{X}, \vec{Y} \rangle = y_1 + y_2 ID + \cdots + y_r ID^{r-1} + y_{r+1} ID^r = \mathcal{F}_{\mathcal{R}}(ID)$. If $r + 1 < R$, then the coefficients

$y_{r+2}, \cdots, y_R$ are equal to 0. Let $S_{ID} = \cup_{A_k \in \mathcal{A}} S_{ID,k}$ be the disjoint union of all the role attribute sets $S_{ID,k}$ of the user with identity ID and associated with $A_k \in \mathcal{A}$. Let τ_{dec} be the binary representation for the decryption time period T_{dec} and $\mathbb{T}$ be the set cover representing the time validity range T_{ID}.

Let us define the following conditions:

- S_{ID} does not satisfy the access structure (M, ρ);
- $ID \in \mathcal{R}$, that is $\langle \vec{X}, \vec{Y} \rangle = \mathcal{F}_{\mathcal{R}}(ID) = 0$;
- T_{dec} is not completely covered in T_{ID}, that is τ_{dec} and all its prefixes are not in $\mathbb{T}$.

If any of the above conditions occurs, then output $\perp$ and abort. Otherwise, since $\langle \vec{X}, \vec{Y} \rangle \neq 0$, compute the following:

$$\begin{aligned}
F_k &= \prod_{i=2}^{R} F_{k,i}^{y_i} = (f_1^{-\langle \vec{X}, \vec{Y} \rangle} \prod_{i=1}^{R} f_i^{y_i})^{u_k} \\
\xi_{k,1} &= \left(\frac{e(F_k, C_0')}{e(C_0'', D_{k,0}')} \right)^{\frac{-1}{\langle \vec{X}, \vec{Y} \rangle}} = e(g_1, g_2)^{\alpha_1 s u_k} \\
E &= \prod_{i=2}^{R} E_i^{y_i} = (f_1^{-\langle \vec{X}, \vec{Y} \rangle} \prod_{i=1}^{R} f_i^{y_i})^{\beta} \\
\xi_1' &= \left(\frac{e(E, C_0')}{e(C_0'', D_2)} \right)^{\frac{-1}{\langle \vec{X}, \vec{Y} \rangle}} = e(g_1, g_2)^{\alpha_1 s \beta}
\end{aligned}$$

Let $I \subseteq [1, l]$ be defined as $\{i; \rho(i) \in S_{ID}\}$ and $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be the set of constants such that if the set $\{\lambda_i\}$ contains valid shares of a value s according to the matrix M, then $\sum_{i \in I} \omega_i \lambda_i = s$. In addition, there is a surjective function from I to $\mathcal{A}$ determined as follows: Let $\pi : k \to \pi(i)$ be defined as $\exists!(A_k \in \mathcal{A}, j \in [1, U_k])$ such that $\rho(i) = k||j$. Such function exists since each attribute is defined uniquely in the whole universe $\mathcal{U} = \cup_{A_k} \mathcal{U}_k$. Then, compute:

$$\xi_2 = \prod_{i \in I} \left(e(C_i, D_{\pi(i),0}) \cdot e(K_{\rho(i)}, C_0') \right)^{\omega_i} = \prod_{A_k \in \mathcal{A}} e(g_1, g_2)^{\delta s t_k}$$

If $\tau_{dec} = (\tau_1, \cdots, \tau_{\eta_{dec}}) \in \mathbb{T}$, then $D_{1,\tau_{dec}}$ should be one component of the secret key TSK_{ID}. Otherwise, let $\tau_{dec}' = (\tau_1, \cdots, \tau_{\eta_{dec}'})$ denote the prefix such that $\eta_{dec}' < \eta_{dec}$ and $\tau_{dec}' \in \mathbb{T}$. Then, derive a key component $D_{1,\tau_{dec}}$ from TSK_{ID} with respect to τ_{dec}' by calculating $D_{1,\tau_{dec}} = D_{1,\tau_{dec}'} \prod_{j=\eta_{dec}'+1}^{\eta_{dec}} L_{j,\tau_{dec}'}^{\tau_j}$ and set $\tau_{dec} = \tau_{dec}'$.

Finally, recover:

$$m = C_0 \cdot \xi_2 \cdot \frac{e(D_{0,\tau_{dec}}, C_0''') \cdot \xi_1'}{e(D_{1,\tau_{dec}}, C_0')} \cdot \prod_{A_k \in \mathcal{A}} \frac{\xi_{k,1}}{e(D_{k,1}, C_0')}$$

Correctness.

$$\begin{aligned}
F_k &= \prod_{i=2}^{R} F_{k,i}^{y_i} = \prod_{i=2}^{R} (f_1^{-ID^{i-1}} f_i)^{y_i u_k} = (f_1^{-(IDy_2+ID^2y_3+\cdots+ID^{R-1}y_R)} \cdot g_1^{\sum_{i=2}^{R} \alpha_i y_i})^{u_k} \\
&= (f_1^{-\langle \vec{X},\vec{Y}\rangle + y_1} \prod_{i=2}^{R} f_i^{y_i})^{u_k} = (f_1^{-\langle \vec{X},\vec{Y}\rangle} \prod_{i=1}^{R} f_i^{y_i})^{u_k}
\end{aligned}$$

$$\xi_{k,1} = \left(\frac{e(F_k, C_0')}{e(C_0'', D_{k,0}')}\right)^{\frac{-1}{\langle \vec{X},\vec{Y}\rangle}} = \left(\frac{e((f_1^{-\langle \vec{X},\vec{Y}\rangle} \prod_{i=1}^{R} f_i^{y_i})^{u_k}, g_2^s)}{e((f_1^{y_1}\cdots f_R^{y_R})^s, g_2^{u_k})}\right)^{\frac{-1}{\langle \vec{X},\vec{Y}\rangle}} = e(g_1,g_2)^{\alpha_1 s u_k}$$

$$\begin{aligned}
E &= \prod_{i=2}^{R} E_i^{y_i} = \prod_{i=2}^{R} (f_1^{-ID^{i-1}} f_i)^{y_i \beta} = (f_1^{-(IDy_2+ID^2y_3+\cdots+ID^{R-1}y_R)} \cdot g_1^{\sum_{i=2}^{R} \alpha_i y_i})^{\beta} \\
&= (f_1^{-\langle \vec{X},\vec{Y}\rangle + y_1} \prod_{i=2}^{R} f_i^{y_i})^{\beta} = (f_1^{-\langle \vec{X},\vec{Y}\rangle} \prod_{i=1}^{R} f_i^{y_i})^{\beta}
\end{aligned}$$

$$\xi_1' = \left(\frac{e(E, C_0')}{e(C_0'', D_2)}\right)^{\frac{-1}{\langle \vec{X},\vec{Y}\rangle}} = \left(\frac{e((f_1^{-\langle \vec{X},\vec{Y}\rangle} \prod_{i=1}^{R} f_i^{y_i})^{\beta}, g_2^s)}{e((f_1^{y_1}\cdots f_R^{y_R})^s, g_2^{\beta})}\right)^{\frac{-1}{\langle \vec{X},\vec{Y}\rangle}} = e(g_1,g_2)^{\alpha_1 s \beta}$$

$$\begin{aligned}
\xi_2 &= \prod_{i\in I} \left(e(C_i, D_{\pi(i),0}) \cdot e(K_{\rho(i)}, C_0')\right)^{\omega_i} \\
&= \prod_{i\in I} \left(e(g_1^{\delta\lambda_i} h_{\rho(i)}^{-s}, g_2^{t_{\pi(i)}}) \cdot e(h_{\rho(i)}^{t_{\pi(i)}}, g_2^s)\right)^{\omega_i} \\
&= \prod_{i\in I} \left(e(g_1^{\delta\lambda_i}, g_2^{t_{\pi(i)}}) \cdot e(h_{\rho(i)}^{-s}, g_2^{t_{\pi(i)}}) \cdot e(h_{\rho(i)}^{t_{\pi(i)}}, g_2^s)\right)^{\omega_i} \\
&= \prod_{i\in I} e(g_1,g_2)^{t_{\pi(i)}\delta(\lambda_i\omega_i)} \\
&= \prod_{A_k\in\mathcal{A}} e(g_1,g_2)^{s t_k \delta}
\end{aligned}$$

$$\begin{aligned}
m &= C_0 \cdot \xi_2 \cdot \frac{e(D_{0,\tau_{dec}}, C_0''') \cdot \xi_1'}{e(D_{1,\tau_{dec}}, C_0')} \cdot \prod_{A_k\in\mathcal{A}} \frac{\xi_{k,1}}{e(D_{k,1}, C_{k,0}')} \\
&= \left(m \cdot e(g_1,g_2)^{\sigma s} \cdot \prod_{A_k\in\mathcal{A}} e(g_1,g_2)^{\kappa_k s}\right) \cdot \prod_{A_k\in\mathcal{A}} e(g_1,g_2)^{s t_k \delta} \cdot \frac{e((V_0 \prod_{j=1}^{\eta_{dec}} V_j^{\tau_j})^s, g_2^{v_{\tau_{dec}}}) \cdot e(g_1,g_2)^{\alpha_1 s \beta}}{e(g_1^{\sigma} g_1^{\alpha_1\beta} (V_0 \prod_{j=1}^{\eta_{dec}} V_j^{\tau_j})^{v_{\tau_{dec}}}, g_2^s)} \\
&\quad \cdot \prod_{A_k\in\mathcal{A}} \frac{e(g_1,g_2)^{\alpha_1 s u_k}}{e(g_1^{\kappa_k} g_1^{\delta t_k} g_1^{\alpha_1 u_k}, g_2^s)} \\
&= m \cdot e(g_1,g_2)^{\sigma s} \cdot \frac{1}{e(g_1,g_2)^{\sigma s}} \cdot \left(\prod_{A_k\in\mathcal{A}} e(g_1,g_2)^{\kappa_k s} \cdot \frac{1}{e(g_1,g_2)^{\kappa_k s}}\right)
\end{aligned}$$

User collusion. The user may have multiple role secret keys $RSK_{ID,k}$, issued by different role authorities A_k, along with a time secret key TSK_{ID}, issued by B. Each key $RSK_{ID,k}$ embeds her identity ID in the component $F_{k,i}$ for authority $A_k \in \mathcal{A}$, and the key TSK_{ID} contains ID in the component E_i, for index $i \in [2, R]$. These elements are required to check whether the user belongs to the revoked list $\mathcal{R}$. If so, then some decrypting elements do not cancel out, and then decryption fails.

Moreover, the elements $F_{k,i}$ and E_i avoid user collusion. Let us suppose that a user with identity ID has the appropriate role attributes to fulfill the decryption requirements, but does not have the suitable time validity range T_{ID}. Hence, this user should not be able to successfully decrypt the ciphertext. We now assume that this user with identity ID colludes with the user with identity ID', that has the suitable time validity range $T_{ID'}$. Thus, the user with identity ID attempts to decrypt the ciphertext using her own role keys $RSK_{ID,k}$ and the time key $TSK_{ID'}$ from the user with identity ID'. We observe that the identity ID is required to generate the vector $\vec{X}$, while the element E contains the identity ID', thus the element ξ_1' is not correctly calculated (some factors are not correctly deleted), and the user with identity ID fails decrypting.

3.2 Security

In order to prove our scheme secure, we suppose that either there is at least one honest role authority whose some attributes are included in the access policy or the time authority is honest. Indeed, if all the (role and time) authorities are malicious and collude, then the key generation may be altered to the advantage of these authorities.

Our Multi-Authority Time-Based Revocable Ciphertext-Policy Attribute-Based Encryption scheme is selectively secure as long as the Decisional q-BDHE assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ [20].

3.2.1 Security Model

As in [20], we consider a selective security model for our solution. The following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{E}$ described the selective security model. In that game, $\mathcal{E}$ first submits a challenged access structure (M^*, ρ^*), a challenged revocation list $\mathcal{R}^*$, a challenged set $\mathcal{A}^*$ of role authorities whose attributes are in the challenged access policy (i.e. the access structure) and a challenged decryption time period T^*_{dec} to $\mathcal{C}$ and then receives the public parameters and authorities' public keys. The adversary is permitted to query users' secret keys that cannot be used to decrypt the challenged ciphertext CT^*.

In addition, following [13, 22], $\mathcal{E}$ selects an honest authority $A_{k^*} \in \mathcal{A}^*$ for some index k^*. Therefore, the adversary is allowed to request secret keys for a given user with identity ID and attribute set S_{ID} as long as there remains one honest authority $A_{k^*} \in \mathcal{A}^*$ such that user has insufficient attributes from this authority to decrypt. Note that we focus on the case where the honest authority is a role one; similarly, one can design the proof with the honest authority being the time one.

Initialization. The adversary submits the challenged access structure (M^*, ρ^*), challenged revocation list $\mathcal{R}^*$ and challenged decryption time period T^*_{dec} to the challenger. It must also provide the challenged set $\mathcal{A}^*$ of role authorities whose attributes are in the challenged access policy and at least one honest authority $A_{k^*} \in \mathcal{A}^*$.

Setup. $\mathcal{C}$ runs the Setup, RSKeyGen and TAKeyGen algorithms and gives to $\mathcal{E}$ the public parameters PP, the public keys PK_k for all A_k and the public key PK for B.

Query Phase 1. The adversary can make secret key queries corresponding to user with identity ID and secret keys $RSK_{ID,k}$ and TSK_{ID}such that:

- The secret keys $RSK_{ID,k}$ of the user with identity ID and associated with A_k result from the role attribute sets $S_{ID,k}$ respectively.
- The secret key TSK_{ID} results from time validity range T_{ID}.

Then, at least one of the following conditions must hold:

- Let $S_{ID} = \cup_{A_k \in \mathcal{A}^*} S_{ID,k}$ be the disjoint union of all the role attribute sets $S_{ID,k}$ of the user with identity ID and associated with $A_k \in \mathcal{A}^*$. S_{ID} does not satisfy (M^*, ρ^*), meaning that for each user with identity ID, there must be at least one honest authority $A_k^* \in \mathcal{A}^*$ from which the adversary never requests enough attributes to decrypt the challenge ciphertext.

  The honest authority A_{k^*} replies such that the corresponding role attribute set S_{ID,k^*} does not satisfy (M^*, ρ^*), meaning that the access structure (M^*, ρ^*) cannot only contain attributes from A_{k^*}.

  In addition, the adversary never queries the same authority twice with the same identity ID [13].
- $ID \in \mathcal{R}^*$, meaning that the user has been revoked.
- T^*_{dec} is not completely covered in T_{ID}, meaning that τ^*_{dec} and all its prefixes are not in $\mathbb{T}$, the set cover of T.

Challenge. The adversary submits two messages m_0 and m_1 of equal length. $\mathcal{C}$ picks a random bit $b \in \{0,1\}$ and encrypts m_b with inputs the challenged access structure (M^*, ρ^*), challenged revocation list $\mathcal{R}^*$, challenged decryption time period T^*_{dec} and challenged authority $A_{k^*} \in \mathcal{A}^*$. The resulting challenged ciphertext CT^* is given to $\mathcal{E}$.

Query Phase 2. This phase is similar to the first one.

Guess. The adversary outputs a bit $b' \in \{0,1\}$ and wins if $b' = b$.

The advantage of the adversary in the game is defined as $Adv_{\mathcal{E}} = Pr[b' = b] - 1/2$.

The revocable CP-ABE scheme is said to be *secure* if no probabilistic polynomial-time adversary has non-negligible advantage in the above game.

3.2.2 Security Proof

Assuming that the Decisional q-BDHE assumption holds, then there is no probabilistic polynomial-time adversary that can selectively break our Multi-Authority Time-Based Revocable Ciphertext-Policy Attribute-Based Encryption scheme with a challenged matrix M^* of size $l^* \times \nu^*$ for $l^*, \nu^* < q$, a challenged revocation list $\mathcal{R}^*$ for $|\mathcal{R}^*| < q - 2$ and a challenged decryption time period T^*_{dec} with binary representation $\tau^*_{dec} \in \{0,1\}^{\eta^*_{dec}}$ for $\eta^*_{dec} < T < q$, along with a set $\mathcal{A}^*$ of role authorities whose attributes are in the challenged access structure and an authority $A_{k^*} \in \mathcal{A}^*$.

Let $\mathcal{E}$ be an adversary with non-negligible advantage against our solution. Let $\mathcal{C}$ be a challenger that interacts with $\mathcal{E}$ and solves the Decisional q-BDHE problem with non-negligible probability.

Initialization. The challenger is given the tuple $\vec{P} = (g_1, g_1^s, g_1^a, \cdots, g_1^{a^q}, g_1^{a^{q+2}}, \cdots, g_1^{a^{2q}}, g_2, g_2^s, g_2^a, \cdots, g_2^{a^q}, g_2^{a^{q+2}}, \cdots, g_2^{a^{2q}}) \in \mathbb{G}_1^{2q+1} \times \mathbb{G}_2^{2q+1}$ and $Q \in \mathbb{G}_T$, and should decide whether $Q = e(g_1, g_2)^{sa^{q+1}}$ by interacting with the adversary. The latter first submits the challenged access structure (M^*, ρ^*), challenged revocation list $\mathcal{R}^*$ and challenged decryption time period T^*_{dec}, a set $\mathcal{A}^*$ of role authorities whose attributes are in (M^*, ρ^*) and a challenged honest authority $A_{k^*} \in \mathcal{A}^*$ to the challenger, such that the matrix M^* has $\nu^* \leq q$ columns, the time period has binary representation $\tau^*_{dec} = (\tau^*_1, \cdots, \tau^*_{\eta^*_{dec}}) \in \{0,1\}^{\eta^*_{dec}}$ for $\eta^*_{dec} < T_{ID} < q$ and $|\mathcal{R}^*| < q - 2$, meaning that the maximum number of revoked users $R - 1$ is set to $q - 2$.

Setup. The challenger chooses random exponents $\theta_0, \vartheta_0, \vartheta_1, \cdots, \vartheta_T \in \mathbb{Z}_p$, $\kappa'_{k^*} \in \mathbb{Z}_p$ for $A_{k^*} \in \mathcal{A}^*$ and $\kappa_k \in \mathbb{Z}_p$ for $A_k \neq A_{k^*}$. Let $g_1^\delta = g_1^a$. It implicitly sets $\kappa_{k^*} = \kappa'_{k^*} + \theta_0 a^{q+1}$ by letting:

$$e(g_1, g_2)^{\kappa_{k^*}} = e(g_1, g_2)^{\kappa'_{k^*}} e(g_1^a, g_2^{a^q})^{\theta_0}$$

Given an authority A_k, for $k||x \in [1, U_k]$, pick at random $z_{k||x} \in \mathbb{Z}_p$. Let I be the set of indices i such that $\rho^*(i) = k||x$ (and where k is such that $A_k \in \mathcal{A}^*$). The challenger programs $h_{k||x}$ as follows:

$$h_{k||x} = g_1^{z_{k||x}} \cdot g_1^{aM^*_{i,1}} \cdot g_1^{a^2 M^*_{i,2}} \cdots g_1^{a^{\nu^*} M^*_{i,\nu^*}}$$

We observe that if $I = \emptyset$ then $h_{k||x} = g_1^{z_{k||x}}$. All the parameters are randomly distributed thanks to the value $g_1^{z_{k||x}}$.

Let $|\mathcal{R}^*| = r \leq q - 2$. Let $\vec{X}_1, \cdots, \vec{X}_r$ be the corresponding vectors for the revoked list $\mathcal{R} = (ID_1, \cdots, ID_r)$, meaning that $\vec{X}_i = (1, ID_i, \cdots, ID_i^{q-2})$ for $i \in [1, r]$. Then, for each $i \in [1, r]$, let the matrix:

$$M_{\vec{X}_i} = \begin{pmatrix} -ID_i & \cdots & -ID_i^{q-2} \\ & \mathcal{I}_{q-2} & \end{pmatrix}$$

where the $\mathcal{I}_{q-2}$ is $(q-2) \times (q-2)$ identity matrix. Then, $\mathcal{C}$ chooses $\vec{B}_i \in \mathbb{Z}_p^{q-1}$ such that $\vec{B}_i \cdot M_{\vec{X}_i} = \vec{0}$. The vector $\vec{B}_i = (1, ID_i, \cdots, ID_i^{q-2}) = \vec{X}_i$ is the simplest candidate. In addition, for $i \in [r+1, q-1]$, let $\vec{B}_i = \vec{0}$. Now, let $\mathbf{B} = (\vec{B}_1 | \cdots \vec{B}_r | \vec{0} | \cdots | \vec{0})$ be a $(q-1) \times (q-1)$ matrix where the i-th column consists of $\vec{B}_i$ for $i \in [1, r]$ and of $\vec{0}$ for $i \in [r+1, q-1]$.

The challenger also defines $V_j = g_1^{\vartheta_j a^{q-j+1}}$ for $j \in [1, T]$ and $V_0 = \prod_{j=1}^{\eta^*} V_j^{-\tau^*_j} g_1^{\vartheta_0}$. It then defines the vector $\vec{\varpi} = (\varpi_1, \cdots, \varpi_{q-1})^\top = (a^q, \cdots, a^2)^\top$ where $\varpi_i = a^{q+1-i}$ and sets $g_1^{\vec{\varpi}} = (g_1^{\varpi_1}, \cdots, g_1^{\varpi_{q-1}})^\top$. It implicitly sets $\vec{\alpha} = \mathbf{B} \cdot \vec{\varpi} + \vec{\theta}$ by randomly choosing $\vec{\theta} = (\theta_1, \cdots, \theta_{q-1})^\top \in \mathbb{Z}_p^{q-1}$. The challenger finally sets $\vec{F} = g_1^{\mathbf{B} \cdot \vec{\varpi}} \cdot g^{\vec{\theta}} = (g_1^{\alpha_1}, \cdots, g_1^{\alpha_R})^\top = (f_1, \cdots, f_R)^\top$.

Query Phase 1. Let $S_{ID} = \cup_{A_k \in \mathcal{A}^*} S_{ID,k}$ be the disjoint union of all the role attribute sets $S_{ID,k}$ of the user with identity ID and associated with $A_k \in \mathcal{A}^*$. We observe that one could define $S_{ID} = \cup_{A_k} S_{ID,k}$ for all A_k (and not necessarily in $\mathcal{A}^*$); however, sets $S_{ID,k}$ for $A_k \notin \mathcal{A}^*$ do not satisfy (M^*, ρ^*) by design.

The adversary makes secret key queries corresponding to user with identity ID and secret keys $RSK_{ID,k}$, TSK_{ID} such that:

- The secret keys $RSK_{ID,k}$ result from the role attribute sets $S_{ID,k}$ respectively.
- The secret key TSK_{ID} results from time validity range T_{ID}.

Then, at least one of the following conditions must hold:

- S_{ID} does not satisfy (M^*, ρ^*) (Case 1).
- $ID \in \mathcal{R}^*$ (Case 2).
- T^*_{dec} is not completely covered in T_{ID} (Case 3).

Case 1: S_{ID} does not satisfy (M^*, ρ^*). The challenger randomly picks $\varphi \in \mathbb{Z}_p$ and finds a vector $\vec{w} = (w_1, \cdots, w_{\nu^*}) \in \mathbb{Z}_p^{\nu^*}$ such that $w_1 = -1$ and for all i where $\rho^*(i) \in S_{ID}$, $\vec{w} \cdot M^*_i = 0$ [20]. By the definition of an LSSS, such a vector must exist since S_{ID} does not satisfy the access structure (M^*, ρ^*).

Then, $\mathcal{C}$ implicitly sets $t_{k^*} = \varphi + \theta_0(w_1 a^q + w_2 a^{q-1} + \cdots + w_{\nu^*} a^{q-\nu^*+1})$ and picks at random $t_k \in \mathbb{Z}_p$ for $A_k \neq A_{k^*}$. The challenger also chooses u_k at random for A_k.

It first computes $D'_{k,0} = g_2^{u_k}$ for all A_k. It also calculates $D_{k,0} = g_2^{t_k}$ and $D_{k,1} = g_1^{\kappa_k} g_1^{\delta t_k} g_1^{\alpha_1 u_k}$ for $A_k \neq A_{k^*}$, and $D_{k^*,0} = g_2^{\varphi} \prod_{i=1}^{\nu^*} (g_2^{a^{q+1-i}})^{w_i \theta_0} = g_2^{t_{k^*}}$ along with:

$$\begin{aligned}
D_{k^*,1} &= g_1^{\kappa'_{k^*}} g_1^{a\varphi_{k^*}} \prod_{i=2}^{\nu^*} (g_1^{a^{q+2-i}})^{w_i \theta_0} g_1^{\alpha_1 u_{k^*}} \\
&= g_1^{\kappa'_{k^*}} g_1^{\theta_0 a^{q+1}} g_1^{a\varphi_{k^*}} g_1^{-\theta_0 a^{q+1}} \prod_{i=2}^{\nu^*} (g_1^{a^{q+2-i}})^{w_i \theta_0} g_1^{\alpha_1 u_{k^*}} \\
&= g_1^{\kappa_{k^*}} g_1^{a\varphi_{k^*}} g_1^{w_1 \theta_0 a^{q+1}} \prod_{i=2}^{\nu^*} (g_1^{a^{q+2-i}})^{w_i \theta_0} g_1^{\alpha_1 u_{k^*}} \text{ where } w_1 = -1 \\
&= g_1^{\kappa_{k^*}} g_1^{a\varphi_{k^*}} \prod_{i=1}^{\nu^*} (g_1^{a^{q+2-i}})^{w_i \theta_0} g_1^{\alpha_1 u_{k^*}} \\
&= g_1^{\kappa_{k^*}} \left(g_1^{\varphi_{k^*}} \prod_{i=1}^{\nu^*} (g_1^{a^{q+1-i}})^{w_i \theta_0} \right)^a g_1^{\alpha_1 u_{k^*}} \\
&= g_1^{\kappa_{k^*}} g_1^{a t_{k^*}} g_1^{\alpha_1 u_{k^*}} \\
&= g_1^{\kappa_{k^*}} g_1^{\delta t_{k^*}} g_1^{\alpha_1 u_{k^*}}
\end{aligned}$$

For all $\tau = (\tau_1, \cdots, \tau_{\eta_\tau}) \in \mathbb{T}$, it randomly picks $\beta, v_\tau \in \mathbb{Z}_p$ and sets $D_2 = g_2^{\beta}$, $D_{0,\tau} = g^{v_\tau}$ and $D_{1,\tau} = g_1^{\sigma} g_1^{\alpha_1 \beta} (V_0 \prod_{j=1}^{\eta_\tau} V_j^{\tau_j})^{v_\tau}$.

If $k||x \in S_{ID,k}$ for which there is no index i such that $\rho^*(i) = k||x$ (and where k is such that $A_k \notin \mathcal{A}^*$), then the challenger sets $K_{k||x} = D_{k,0}^{z_{k||x}}$. Otherwise (i.e. $k||x \in S_{ID,k}$ for which there is an index i such that $\rho^*(i) = k||x$ and where k is such that $A_k \in \mathcal{A}^*$) [34], then $\mathcal{C}$ computes:

$$K_{k||x} = D_{k,0}^{z_{k||x}} \cdot \prod_{j=1}^{\nu^*} g_1^{a^j \varphi_k} \Big(\prod_{l=1, l \neq j}^{\nu^*} (g^{a^{q+1+j-l}})^{w_l \theta_0} \Big)^{M^*_{i,j}}$$

Finally, $\mathcal{C}$ sets $F_{k,i} = (f_1^{-ID^{i-1}} \cdot f_i)^{u_k}$ and $E_i = (f_1^{-ID^{i-1}} \cdot f_i)^{\beta}$ for all A_k and $i \in [2, R]$, and $L_{j,\tau} = V_j^{v_\tau}$ for $j \in [\eta_\tau + 1, T]$ and $\tau \in \mathbb{T}$.

Case 2: $ID \in \mathcal{R}^*$**.** For $j \in [1, r]$, $ID_j \in \mathcal{R}^*$ be the identity of the secret key that the adversary queries [20, 6]. The challenger defines $\tilde{\beta}_j = \beta_j - \theta_0 a^j$ and $\tilde{u}_{k^*,j} = u_{k^*,j} - \theta_0 a^j$ for a random exponent $\beta_j, u_{k^*,j} \in \mathbb{Z}_p$. It also chooses at random $u_{k,j} \in \mathbb{Z}_p$ for $A_k \neq A_{k^*}$.

From the equation $\vec{\alpha} = \mathbf{B} \cdot \vec{\varpi} + \vec{\theta}$, the first coordinate of the vector $\vec{\alpha}$ is the following:

$$\alpha_1 = \sum_{i=1}^{r} \varpi_i + \theta_1 = \sum_{i=1}^{r} a^{q+1-i} + \theta_1$$

Then, the challenger computes $D_2 = g_2^{\beta_j} (g_2^{a^j})^{-\theta_0} = g_2^{\tilde{\beta}_j}$, $D'_{k^*,0} = g_2^{u_{k^*,j}} (g_2^{a^j})^{-\theta_0} = g_2^{\tilde{u}_{k^*,j}}$ and $D'_{k,0} = g_2^{u_{k,j}}$ for $A_k \neq A_{k^*}$. For all $\tau = (\tau_1, \cdots, t_{\eta_\tau}) \in \mathbb{T}$, it randomly chooses $v_\tau \in \mathbb{Z}_p$, and computes $D_{0,\tau} = g_2^{v_\tau}$ along with:

$$\begin{aligned} D_{1,\tau} &= g_1^{\sigma} g_1^{\alpha_1 \beta_j - \alpha_1 \theta_0 a^j} (V_0 \prod_{j=1}^{\eta_\tau} V_j^{\tau_j})^{v_\tau} \\ &= g_1^{\sigma} g_1^{\alpha_1 \tilde{\beta}_j} (V_0 \prod_{j=1}^{\eta_\tau} V_j^{\tau_j})^{v_\tau} \end{aligned}$$

In addition, it picks at random $t_k \in \mathbb{Z}_p$ for all A_k, and calculates $D_{k,1} = g_1^{\kappa_k} g_1^{\delta t_k} g_1^{\alpha_1 u_{k,j}}$ for $A_k \neq A_{k^*}$, along with:

$$\begin{aligned} D_{k^*,1} &= g_1^{\kappa'_{k^*}} f_1^{u_{k^*,j}} (g_1^{a^j \theta_1} \prod_{i=1, i\neq j}^{r} g_1^{a^{q+1-i+j}})^{-\theta_0} g_1^{a t_{k^*}} \\ &= g_1^{\kappa'_{k^*}} g_1^{\alpha_1 u_{k^*,j}} g_1^{\theta_0 a^{q+1}} (g_1^{\theta_1} \prod_{i=1}^{r} g_1^{a^{q+1-i}})^{-\theta_0 a^j} g_1^{\delta t_{k^*}} \\ &= g_1^{\kappa_{k^*}} g_1^{\delta t_{k^*}} g_1^{\alpha_1 u_{k^*,j}} (g_1^{\alpha_1})^{-\theta_0 a^j} \\ &= g_1^{\kappa_{k^*}} g_1^{\delta t_{k^*}} g_1^{\alpha_1 u_{k^*,j} - \alpha_1 \theta_0 a^j} \\ &= g_1^{\kappa_{k^*}} g_1^{\delta t_{k^*}} g_1^{\alpha_1 \tilde{u}_{k^*,j}} \end{aligned}$$

Let $\mathbb{F}_{k,j} = (F_2, \cdots, F_R)^\top$ be the secret key component for the identity ID_j. We recall that $\vec{\varpi} = (\varpi_1, \cdots, \varpi_{q-1})^\top = (a^q, \cdots, a^2)^\top$ with $\varpi_i = a^{q+1-i}$ and $g_1^{\vec{\varpi}} = (g_1^{\varpi_1}, \cdots, g_1^{\varpi_{q-1}})^\top$. First, we observe that $\mathcal{C}$ can compute $g_1^{a^j M_{\vec{X}_j}^\top \mathbf{B} \vec{\varpi}}$ because the j-th column of $M_{\vec{X}_j}^\top \mathbf{B}$ is equal to $\vec{0}$. The challenger computes $\mathbb{F}_{k,j} = g_1^{u_{k,j} M_{\vec{X}_j}^\top \vec{\alpha}}$ for $A_k \neq A_{k^*}$, as well as:

$$\begin{aligned} \mathbb{F}_{k^*,j} &= g_1^{u_{k^*,j} M_{\vec{X}_j}^\top \vec{\alpha}} \cdot g_1^{-\theta_0 a^j M_{\vec{X}_j}^\top \mathbf{B} \vec{\varpi}} \cdot g_1^{-\theta_0 a^j M_{\vec{X}_j}^\top \vec{\theta}} \\ &= g_1^{u_{k^*,j} M_{\vec{X}_j}^\top \vec{\alpha}} \cdot g_1^{-\theta_0 a^j M_{\vec{X}_j}^\top \vec{\alpha}} \\ &= g_1^{(u_{k^*,j} - \theta_0 a^j) M_{\vec{X}_j}^\top \vec{\alpha}} \\ &= g_1^{\tilde{u}_{k^*,j} M_{\vec{X}_j}^\top \vec{\alpha}} \end{aligned}$$

It also calculates $\mathbb{E}$ as follows:

$$\begin{aligned}\mathbb{E}_j &= g_1^{\beta_j M_{\vec{X}_j}^{\top}\vec{\alpha}} \cdot g_1^{-\theta_0 a^j M_{\vec{X}_j}^{\top}\mathbf{B}\vec{\varpi}} \cdot g_1^{-\theta_0 a^j M_{\vec{X}_j}^{\top}\vec{\theta}} \\ &= g_1^{\beta_j M_{\vec{X}_j}^{\top}\vec{\alpha}} \cdot g_1^{-\theta_0 a^j M_{\vec{X}_j}^{\top}\vec{\alpha}} \\ &= g_1^{(\beta_j - \theta_0 a^j) M_{\vec{X}_j}^{\top}\vec{\alpha}} \\ &= g_1^{\tilde{\beta}_j M_{\vec{X}_j}^{\top}\vec{\alpha}}\end{aligned}$$

We recall that $R = q - 1$ and we denote $M_{\vec{X}_j,i-1}^{\top}$ as the $(i-1)$-th row of $M_{\vec{X}_j}^{\top}$, then for $i \in [2, R]$, we have $F_{k,i} = (f_1^{-ID_j^{i-1}} \cdot f_i)^{u_{k,j}}$ for $A_k \neq A_{k^*}$, and the following:

$$\begin{aligned}F_{k^*,i} &= g_1^{\tilde{u}_{k^*,j} M_{\vec{X}_j,i-1}^{\top}\vec{\alpha}} \\ &= g_1^{\tilde{u}_{k^*,j}(-ID_j^{i-1}\alpha_1 + \alpha_i)} \\ &= (f_1^{-ID_j^{i-1}} \cdot f_i)^{\tilde{u}_{k^*,j}}\end{aligned}$$

$$\begin{aligned}E_i &= g_1^{\tilde{\beta}_j M_{\vec{X}_j,i-1}^{\top}\vec{\alpha}} \\ &= g_1^{\tilde{\beta}_j(-ID_j^{i-1}\alpha_1 + \alpha_i)} \\ &= (f_1^{-ID_j^{i-1}} \cdot f_i)^{\tilde{\beta}_j}\end{aligned}$$

Finally, the challenger computes $K_{k||x} = h_{k||x}^{t_k}$ for $k||x \in S_{ID,k}$ and $L_{j,\tau} = V_j^{v_\tau}$ for $j \in [\eta_\tau + 1, T]$ and $\tau \in \mathbb{T}$.

Case 3: τ_{dec}^* and all its prefixes are not in $\mathbb{T}$. For all $\tau = (\tau_1, \cdots, \tau_{\eta_\tau}) \in \mathbb{T}$, let $\tau_{\eta_\tau+1}, \cdots, \tau_q = 0$ and $\tau_{\eta_{dec}^*+1}^*, \cdots, \tau_q^* = 0$. Let $\eta' \leq \eta_{dec}^*$ be the smallest index such that $\tau_{\eta'} \neq \tau_{\eta'}^*$.

$\mathcal{C}$ randomly chooses $t_k, u_k \in \mathbb{Z}_p$ for $A_k \neq A_{k^*}$ along with $t_{k^*}, u_{k^*} \in \mathbb{Z}_p$, sets $u_{k^*} = u'_{k^*} - \frac{\theta_0}{\alpha_1} a^{\eta'}$. It then computes $D_{k,0} = g_2^{t_k}$ for all A_k. It also calculates $D'_{k^*,0} = g_2^{u'_{k^*} - \frac{\theta_0}{\alpha_1} a^{\eta'}} = g_2^{u_{k^*}}$ along with:

$$\begin{aligned}D_{k^*,1} &= g_1^{\kappa'_{k^*}} g_1^{\theta_0 a^{q+1-\eta'}} g_1^{\delta t_{k^*}} g_1^{\alpha_1 u'_{k^*}} \\ &= g_1^{\kappa'_{k^*}} g_1^{\theta_0 a^{q+1}} g_1^{\delta t_{k^*}} g_1^{\alpha_1 u'_{k^*}} g_1^{-\theta_0 a^{\eta'}} \\ &= g_1^{\kappa_{k^*}} g_1^{\delta t_{k^*}} g_1^{\alpha_1 u_{k^*}}\end{aligned}$$

It sets $D'_{k,0} = g_2^{u_k}$ and $D_{k,1} = g_1^{\kappa_k} g_1^{\delta t_k} g_1^{\alpha_1 u_k}$ for $A_k \in \mathcal{A} \setminus \{A_{k^*}\}$. In addition, the challenger picks at random $\beta, v_\tau \in \mathbb{Z}_p$ and calculates $D_2 = g_2^{\beta}$ and $D_{0,\tau} = g_2^{v_\tau}$. We recall that $V_j = g_1^{\vartheta_j a^{q-j+1}}$ for $j \in [1, T]$ and $V_0 = \prod_{j=1}^{\eta^*} V_j^{-\tau_j^*} g_1^{\vartheta_0}$. For all τ, it sets:

$$\begin{aligned}
D_{1,\tau} &= g_1^{\sigma} g_1^{\alpha_1\beta} g_1^{\vartheta_0 v_\tau} g_1^{\vartheta_{\eta'} a^{q-\eta'+1}(\tau_{\eta'}-\tau^*_{\eta'}))v_\tau} \prod_{j=\eta'+1}^{\eta_\tau+1} g_1^{\vartheta_j a^{q-j+1}\tau^*_j v_\tau} \\
&= g_1^{\sigma} g_1^{\alpha_1\beta} g_1^{(\vartheta_0+\vartheta_{\eta'} a^{q-\eta'+1}(\tau_{\eta'}-\tau^*_{\eta'}))v_\tau} \prod_{j=1}^{\eta'-1} V_j^{-\tau^*_j v_\tau} \prod_{j=1}^{\eta'-1} V_j^{\tau_j v_\tau} \prod_{j=\eta'+1}^{\eta_\tau+1} g_1^{\vartheta_j a^{q-j+1}\tau^*_j v_\tau} \\
&\quad \text{since } \tau_j = \tau^*_j \text{ if } j < \eta' \\
&= g_1^{\sigma} g_1^{\alpha_1\beta} (\prod_{j=1}^{\eta'} V_j^{-\tau^*_j} g_1^{\vartheta_0})^{v_\tau} \prod_{j=1}^{\eta'} V_j^{\tau_j v_\tau} \prod_{j=\eta'+1}^{\eta_\tau+1} g_1^{\vartheta_j a^{q-j+1}\tau^*_j v_\tau} \\
&= g_1^{\sigma} g_1^{\alpha_1\beta} (V_0 \prod_{j=1}^{\eta'} V_j^{\tau_j})^{v_\tau} \prod_{j=\eta'+1}^{\eta_\tau+1} V_j^{\tau_j v_\tau} \\
&= g_1^{\sigma} g_1^{\alpha_1\beta} (V_0 \prod_{j=1}^{\eta_\tau+1} V_j^{\tau_j})^{v_\tau} \\
&= g_1^{\sigma} g_1^{\alpha_1\beta} (V_0 \prod_{j=1}^{\eta_\tau} V_j^{\tau_j})^{v_\tau} \text{ since } \tau_{\eta_\tau+1} = 0
\end{aligned}$$

The challenger also sets $K_{k||x} = h_{k||x}^{t_k}$ for $k||x \in S_{ID,k}$. For $i \in [2, R]$, it computes $E_i = (f_1^{-ID^{i-1}} \cdot f_i)^{\beta}$, $F_{k^*,i} = (f_1^{-ID^{i-1}} \cdot f_i)^{u_{k^*}} = (f_1^{-ID^{i-1}} \cdot f_i)^{u'_{k^*} - \frac{\theta_0}{\alpha_1} a^{\eta'}}$ and $F_{k,i} = (f_1^{-ID^{i-1}} \cdot f_i)^{u_k}$ for $A_k \neq A_{k^*}$. For $j \in [\eta_\tau + 1, T]$ and $\tau \in \mathbb{T}$, the challenger computes $L_{j,\tau} = g_1^{\vartheta_j a^{q-j+1} v_\tau} = V_j^{v_\tau}$.

Challenge. The adversary submits two messages m_0 and m_1 of equal length. $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$ and encrypts m_b under the access structure (M^*, ρ^*), the revocation list $\mathcal{R}^*$, the challenged authority set $\mathcal{A}^*$, the challenged authority $A_{k^*} \in \mathcal{A}^*$ and decryption time period T^*_{dec} with binary representation τ^* as follows. The challenger first computes:

$$\begin{aligned}
C_0 &= m_b \cdot Q^{\theta_0} \cdot e(g_1^s, g_2^{\kappa'_{k^*}}) \cdot e(g_1^s, g_2^{\sigma}) \cdot \prod_{A_k \in \mathcal{A}^* \setminus \{A_{k^*}\}} e(g_1^s, g_2^{\kappa_k}) \\
C'_0 &= g_2^s
\end{aligned}$$

It then creates C''_0 as follows [6]. Let $\mathcal{R}^* = (ID_1, \cdots, ID_r)$ and $\mathcal{F}_{\mathcal{R}^*}(Z) = (Z - ID_1) \cdots (Z - ID_r) = y_1 + y_2 Z + \cdots + y_r Z^{r-1} + y_{r+1} Z^r$. If $r + 1 < R$, then the coefficients $y_{r+2}, \cdots, y_R$ are set to be equal to 0. Let $\vec{Y} = (y_1, \cdots, y_R)^\top$ satisfy $\langle \vec{X}_j, \vec{Y} \rangle = 0$ for $j \in [1, r]$. We claim that $\vec{Y}^\top \cdot \mathbf{B} \cdot \vec{\varpi} = 0$ [20]. Hence, we obtain that $\langle \vec{Y}, \vec{\alpha} \rangle = \langle \vec{Y}, \vec{\theta} \rangle$. It then sets $C''_0 = (g_1^s)^{\langle \vec{Y}, \vec{\theta} \rangle}$.

We also observe that the terms g^{a^i} in V_i are canceled out since the challenged time is $\tau^* = (\tau^*_1, \cdots, \tau^*_{\eta^*})$. The challenger computes $C'''_0 = (g_1^s)^{\vartheta_0}$.

From [34], the terms C_i are computed as follows. Since the terms $h^s_{\rho^*(i)}$ contain terms of the form $g^{a^i s}$ that cannot be simulated, the secret splitting technique is thus required, and the latter terms can be canceled out. The challenger chooses exponents $\gamma'_2, \cdots, \gamma'_{\nu^*} \in \mathbb{Z}_p$ and then shares the secret s using the vector:

$$\vec{v}^* = (s, sa + \gamma'_2, \cdots, sa^{\nu^*-1} + \gamma'_{\nu^*}) \in \mathbb{Z}_p^{\nu^*}$$

This permits the terms $h^s_{-\rho^*(i)}$ cancel out with the terms $g_1^{a\lambda_i}$. Then, for $i \in [1, \nu^*]$, the challenger computes $C_i = (g_1^s)^{-z_{\rho^*(i)}} \cdot \prod_{j=2}^{\nu^*} (g^a)^{M^*_{i,j}\gamma'_j}$. In order to see the correct simulation of the terms C_i, we first define:

$$\begin{aligned} \lambda_i^* &= \langle \vec{v}^*, \vec{M}_i^* \rangle \\ &= sM^*_{i,1} + (sa + \gamma'_2)M^*_{i,2} + \cdots + (sa^{\nu^*-1} + \gamma'_{\nu^*})M^*_{i,\nu^*} \end{aligned}$$

Thus, the correct distribution of C_i should be as follows:

$$\begin{aligned} C_i &= g_1^{a\lambda_i^*} h^{-s}_{\rho^*(i)} \\ &= g_1^{a(sM^*_{i,1}+(sa+\gamma'_2)M^*_{i,2}+\cdots+(sa^{\nu^*-1}+\gamma'_{\nu^*})M^*_{i,\nu^*})} \cdot g_1^{-sz_{\rho^*(i)}} \cdot g_1^{-(saM^*_{i,1}+\cdots+sa^{\nu^*}M^*_{i,\nu^*})} \\ &= (g_1^s)^{-z_{\rho^*(i)}} \cdot g_1^{aM^*_{i,2}+\cdots+aM^*_{i,\nu^*}} \\ &= (g_1^s)^{-z_{\rho^*(i)}} \cdot \prod_{j=2}^{\nu^*} (g^a)^{M^*_{i,j}\gamma'_j} \end{aligned}$$

Query Phase 2. The same as in Phase 1.

Guess. The adversary outputs a guess $b' \in \{0, 1\}$ for b. If $b = b'$, then the challenger outputs 0 to guess that $Q = e(g_1, g_2)^{sa^{q+1}}$. Otherwise, $\mathcal{C}$ outputs 1 to guess that Q is a random element of the group $\mathbb{G}_T$.

When Q is equal to $e(g_1, g_2)^{sa^{q+1}}$, then $\mathcal{C}$ gives a perfect simulation, and its advantage is the same than the adversary's one. When Q is a random element of $\mathbb{G}_T$, then the message m_b is completely hidden from $\mathcal{E}$, and thus $Pr[\mathcal{C}(\vec{P}, Q \in_R \mathbb{G}_T) = 0] = 1/2$. Hence, the challenger can solve the Decisional q-BDHE problem with non-negligible advantage.

From selective security to static security. We have proved our scheme selectively secure in the standard model, meaning that the adversary submits at the really beginning of the game a challenged access structure (M^*, ρ^*), a challenged authority set $\mathcal{A}^*$, a challenged honest authority $A^* \in \mathcal{A}^*$, a challenged revocation list $\mathcal{R}^*$ and a challenged decryption time period T^*_{dec} to $\mathcal{C}$, before receiving the public parameters and authorities' public keys from the challenger.

A possible improvement is to consider static security, where all challenged items and queries submitted by $\mathcal{E}$ are sent to the challenger directly after seeing the public parameters. Such improvement would be done following Rouselakis and Waters' technique [28]. Their technique enables the challenger of the security reduction to separate an unauthorized set of the matrix rows and pass over this set for the remaining of the reduction. $\mathcal{C}$ then ignores the contributions of these rows even in the construction of the challenged ciphertext.

4 Evaluation

4.1 Theoretical Analysis and Comparison

Table 1 compares the efficiency of our scheme and the LYZL scheme [20]. Let $N + 1$ be the number of authorities in our system (i.e. N role authorities and 1 time authority). Let $R - 1$ be the maximum number of revoked users and T be the depth of the tree. Let

Scheme	Public key material	User secret key material	Ciphertext	Decryption time (# of pairings)
LYZL [20]	$(U+R+T+3)\mathbb{G}$ $+\mathbb{G}_T$	$(R+1+S$ $+\frac{1}{2}T(T+3))\mathbb{G}$	$(l+3)\mathbb{G}+\mathbb{G}_T$	$4+2l$
Ours	$(U+R+T+3)\mathbb{G}_1$ $+\mathbb{G}_2+(N+1)\mathbb{G}_T$	$(N(R+1)+S$ $+\frac{1}{2}T(T+3))\mathbb{G}_1$ $+2(N+1)\mathbb{G}_2$	$(l+2)\mathbb{G}_1$ $+\mathbb{G}_2+\mathbb{G}_T$	$3N+2l$ $+2$

Table 1: Size of keys and ciphertexts, and number of pairing operations during decryption.

U be the number of attributes in the whole universe $\mathcal{U}$, and S be the total number of role attributes of the user. Let l denote the number of attributes used in the decryption.

The public key material contains the public parameters along with the authorities' public keys. The user secret key material contains the user keys issues by all the involved authorities.

At first sight, the scheme in [20] seems to be more efficient than ours. We easily observe that extra elements in our case are due the multi-authority setting. We recall that a single authority is responsible of generating user key material in [20], while $N+1$ authorities are involved in our system. By setting $N=1$, the performance of our system is equivalent to the one of the LYZL scheme. Hence, our attempt to overcome single points of failure is at the expense of practicality.

Yet, the tree storage costs do not appear in Table 1. We recall that Liu et al. [20] suggest one common value z as the number of children per node. By setting $z=31$, many dummy nodes are created, and storage costs get cumbersome.

Let us compare the two tree-based methods with an IoT-related example. Suppose that an actuator is granted to request data from its connected sensors on a period of 7 days, starting on "04 January 2020". In both schemes, we suppose that the starting time (root of the tree) is "01 January 2020". Following the tree-based method used in [20] with leaf nodes as days (and $T=4$ representing year, month and day levels), the actuator obtains 7 keys, one for each day. Following our method with a time interval of 16 days as in Figure 2 (i.e. $T=5$), the actuator receives 3 keys. Therefore, our technique is more efficient when dealing with data valuable over short time periods, say on a daily basis.

While the LYZL solution [20] is interesting in some cases, e.g. within a company where the system is setup in a narrow, private environment and time periods are of the order of months or years, it does not fit our IoT scenario which involves time-sensitive data and numerous heterogeneous devices. On the other hand, our solution is attractive in IoT environments, with a profitable framework for short time periods and an advantageous security level meeting IoT requirements.

4.2 Implementation and Practical Analysis

Timing and memory consumption for encryption processes should be as low as possible since the latter are executed by resource-constrained devices; while timing and memory for system setup and key generation can be significantly higher since they are called by powerful entities only once.

Choice of parameters. In an IoT environment, access policies contain up to 30 attributes, and devices are allocated around 10 attributes [5, 36]. Let the number of roles authorities N be in the set $\{1,2,5,10\}$. Therefore, we set $U=30$ and S equal to 5, 10 and 15. We consider the number of attributes used in the decryption l equal to $\lceil \frac{S}{2} \rceil$, that is 3, 5 and 8 respectively. We consider a maximum of $R-1$ revoked users, for $R=10$, 20 and 30.

From the tree design proposed in [20] and above discussions, we suggest a tree of depth $T=5$, with 16 leaves. We compare such suggestion with other depths, i.e. $T=4$, 6 and 7.

Future work. We aim to implement our protocol and analyze the timing and memory consumption. The implementation will use the Python-based Charm Crypto library [3], an open source framework developed for rapid prototyping of cryptographic systems.

Our goal is to present the practicality of the scheme in an IoT environment while considering the limited capacities of such framework, in terms of computation, communication and storage. We will examine timing and memory consumption at encryption and decryption phases.

5 Conclusion

In this paper, we designed a system with access control key updates and direct user revocation, that are beneficial features in IoT. Access control is done using Ciphertext-Policy Attribute-Based Encryption where attributes represent roles of devices within their networks. Moreover, we devise a novel approach, based on a binary tree, to append time credentials. This allows us to find an interesting trade-off between key update frequency and user revocation list length, as well as stressing time-sensitive data exchanged in IoT environments. The security of our scheme is proved under the Decisional Bilinear Diffie-Hellman Exponent assumption. The implementation results will show that our solution is fully deployable in IoT networks.

References

[1] M. Abomhara and G. M. Køien. Security and privacy in the internet of things: Current status and open issues. In *2014 International Conference on Privacy and Security in Mobile Systems (PRISMS)*, pages 1–8, May 2014.

[2] C. Adams and S. Lloyd. *Understanding Public-Key Infrastructure: Concepts, Standards, and Deployment Considerations.* Macmillan Technical Publishing, 1999.

[3] J. A. Akinyele, M. D. Green, and A. D. Rubin. Charm: A framework for rapidly prototyping cryptosystems. Cryptology ePrint Archive, Report 2011/617, 2011.

[4] Z. H. Ali, H. A. Ali, and M. M. Badawy. Article: Internet of things (iot): Definitions, challenges and recent research directions. *International Journal of Computer Applications*, 128(1):37–47, October 2015. Published by Foundation of Computer Science (FCS), NY, USA.

[5] M. Ambrosin, A. Anzanpour, M. Conti, T. Dargahi, S. R. Moosavi, A. M. Rahmani, and P. Liljeberg. On the feasibility of attribute-based encryption on internet of things devices. *IEEE Micro*, 36(6):25–35, Nov 2016.

[6] N. Attrapadung, B. Libert, and E. De Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In *Proceedings of the 14th International*

Conference on Practice and Theory in Public Key Cryptography Conference on Public Key Cryptography, PKC'11, pages 90–108, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution.* PhD thesis, Israel, 1996.

[8] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, pages 321–334, Washington, DC, USA, 2007. IEEE Computer Society.

[9] O. Blazy and C. Chevalier. Spreading alerts quietly: New insights from theory and practice. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ARES 2018, pages 30:1–30:6, New York, NY, USA, 2018. ACM.

[10] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer Berlin Heidelberg, 2005.

[11] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of the 21th Annual International Conference on Advances in Cryptology*, CRYPTO'01, pages 213–229. Springer Berlin Heidelberg, 2001.

[12] V. Boyko, M. Peinado, and R. Venkatesan. Speeding up discrete log and factoring based schemes via precomputations. In K. Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pages 221–235, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[13] M. Chase. Multi-authority attribute based encryption. In *Proceedings of the 4th Conference on Theory of Cryptography*, TCC'07, pages 515–534, Berlin, Heidelberg, 2007. Springer-Verlag.

[14] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113 – 3121, 2008. Applications of Algebra to Cryptography.

[15] A. Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In *Proc. of the 11th International Conference on Applied Cryptography and Network Security*, ACNS'13, pages 357–372, Berlin, Heidelberg, 2013. Springer-Verlag.

[16] J. Herranz, F. Laguillaumie, and C. Ráfols. Constant size ciphertexts in threshold attribute-based encryption. In P. Nguyen and D. Pointcheval, editors, *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 19–34. Springer Berlin Heidelberg, 2010.

[17] Y. H. Hwang. Iot security & privacy: Threats and challenges. In *Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security*, IoTPTS '15, pages 1–1, New York, NY, USA, 2015. ACM.

[18] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[19] D. Kozlov, J. Veijalainen, and Y. Ali. Security and privacy threats in iot architectures. In *Proceedings of the 7th International Conference on Body Area Networks*, BodyNets '12, pages 256–262, ICST, Brussels, Belgium, Belgium, 2012. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[20] J. K. Liu, T. H. Yuen, P. Zhang, and K. Liang. Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list. In B. Preneel and F. Vercauteren, editors, *Applied Cryptography and Network Security*, pages 516–534, Cham, 2018. Springer International Publishing.

[21] Q. Liu, G. Wang, and J. Wu. Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. *Information Sciences*, 258:355 – 370, 2014.

[22] R. Longo, C. Marcolla, and M. Sala. Key-policy multi-authority attribute-based encryption. In A. Maletti, editor, *Algebraic Informatics*, pages 152–164, Cham, 2015. Springer International Publishing.

[23] D. Macedo, L. A. Guedes, and I. Silva. A dependability evaluation for internet of things incorporating redundancy aspects. In *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, pages 417–422, April 2014.

[24] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer. A comparative study of lpwan technologies for large-scale iot deployment. *ICT Express*, 5(1):1 – 7, 2019.

[25] N. Oualha and K. T. Nguyen. Lightweight attribute-based encryption for the internet of things. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6, Aug 2016.

[26] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow. Iotpot: Analysing the rise of iot compromises. In *Proceedings of the 9th USENIX Conference on Offensive Technologies*, WOOT'15, pages 9–9, Berkeley, CA, USA, 2015. USENIX Association.

[27] C. Pham, Y. Lim, and Y. Tan. Management architecture for heterogeneous iot devices in home network. In *2016 IEEE 5th Global Conference on Consumer Electronics*, pages 1–5, Oct 2016.

[28] Y. Rouselakis and B. Waters. Efficient statically-secure large-universe multi-authority attribute-based encryption. In R. Böhme and T. Okamoto, editors, *Financial Cryptography and Data Security*, pages 315–332, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[29] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'05, pages 457–473, Berlin, Heidelberg, 2005. Springer-Verlag.

[30] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch. Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture. *IEEE Communications Magazine*, 55(2):70–78, February 2017.

[31] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder. Management of resource constrained devices in the internet of things. *IEEE Communications Magazine*, 50(12):144–149, December 2012.

[32] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of the Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO'84, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[33] B. Waters. Efficient identity-based encryption without random oracles. In *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'05, pages 114–127, Berlin, Heidelberg, 2005. Springer-Verlag.

[34] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography Conference on Public Key Cryptography*, PKC'11, pages 53–70, Berlin, Heidelberg, 2011. Springer-Verlag.

[35] K. Yang and X. Jia. Expressive, efficient, and revocable data access control for multi-authority cloud storage. *IEEE Transactions on Parallel and Distributed Systems*, 25(7):1735–1744, July 2014.

[36] X. Yao, Z. Chen, and Y. Tian. A lightweight attribute-based encryption scheme for the internet of things. *Future Gener. Comput. Syst.*, 49(C):104–112, Aug. 2015.