Ciphertext Outdate Attacks on the Revocable Attribute-Based Encryption Scheme with Time Encodings

Kwangsu Lee

Abstract

Cloud storage is a new computing paradigm that allows users to store their data in the cloud and access them anytime anywhere through the Internet. To address the various security issues that may arise in the cloud storage accessed by a large number of users, cryptographic encryption should be considered. Currently, researches on revocable attribute-based encryption (RABE) systems, which provide user revocation function and ciphertext update function by extending attribute-based encryption (ABE) systems that provide access control to ciphertexts, are actively being studied. Recently, Xu et al. proposed a new RABE scheme that combines ABE and identity-based encryption (IBE) schemes to efficiently handle ciphertext update and user revocation functionality. In this paper, we show that there is a serious security problem in Xu et al.’s RABE scheme such that a cloud server can obtain the plaintext information of stored ciphertexts by gathering invalidated credentials of revoked users. Additionally, we also show that the RABE scheme of Xu et al. can be secure in a weaker security model where the cloud server cannot obtain any invalidated credentials of revoked users.

Keywords: Cloud storage, Access control, Attribute-based encryption, Revocation, Ciphertext update.

1 Introduction

Cloud storage is a computing paradigm that stores data in a centralized cloud and allows users to access these data anytime anywhere on the Internet using simple client devices. The main advantages of cloud storage include flexible accessibility, ease management, and cost savings. Despite these advantages, cloud storage is inevitably experiencing a variety of security issues because it stores data in an external cloud storage that is outside of the control of a data owner. The key reason that the cloud storage security differs from the existing computer server security is that a cloud server that provides the cloud storage service is not fully trusted so that the cloud server can access the stored data and leak the sensitive information [2,5].

The easiest way to keep users data secure in cloud storage is to encrypt the data and store it in the cloud. In this case, in order to share the encrypted data with many users, it is needed to effectively control access to the encrypted data according to the authority of the dynamically changing user. That is, the cloud storage system needs to revoke some users whose credentials are no longer valid so that revoked users cannot access data. In addition, the cloud storage system should be able to prevent previously revoked users to gain access to encrypted data that were created long ago by using their old private keys after colluding with the cloud server.

To solve these problems in cloud storage, we can use attribute-based encryption (ABE), which provides access control to ciphertexts. Boldyreva et al. [1] proposed a revocable ABE (RABE) scheme that extends

*Sejong University, Seoul, Korea. Email: kwangsu@sejong.ac.kr.
the ABE scheme by providing the ability to revoke a user’s private key. Sahai et al. [6] proposed a revocable-
storage ABE (RS-ABE) scheme by extending the concept of RABE that provides the ciphertext update
functionality to prevent previously revoked users from accessing previously created ciphertexts in cloud
storage. After that, Lee et al. [3] proposed efficient RS-ABE schemes that can update ciphertexts more
efficiently by combining a self-updatable encryption (SUE) scheme and an ABE scheme. Therefore, RS-
ABE schemes, which provide user revocation and ciphertext update, can be a solution to the problem of
cloud storage described above. Recently, Xu et al. [7] proposed an RABE scheme that combines an ABE
scheme with an IBE scheme by introducing new time encoding functions to efficiently support ciphertext
update than the existing RS-ABE schemes. Compared with the most efficient RS-ABE scheme of Lee et al.,
the RABE scheme of Xu et al. is more efficient in terms of ciphertext size and update key size.

In this paper, we show that it is possible to break the security of the RABE scheme of Xu et al. [7]. A
key feature of cloud storage is that a cloud server is not fully trusted [5]. In other words, the cloud server
faithfully performs the tasks requested by users, but is curious about the information of the users’ data.
Thus the cloud server should also be considered as an inside attacker. However, Xu et al. have overlooked
that the cloud server can be this type of attackers. Suppose that a cloud server is an inside attacker in the
RABE scheme of Xu et al. Then the cloud server first gathers revoked credentials (revoked private keys) of
users. Note that these revoked credentials are usually no harm to the system since they are safely disabled
in publicly broadcasted key updates. Next, the cloud server derives a new ciphertext associated with past
time from stored original ciphertexts in cloud storage without modifying the original ciphertexts. Then the
cloud server can sufficiently decrypt the derived ciphertext with past time by using the revoked credentials
and publicly available key updates. Thus the RABE scheme of Xu et al. cannot be secure against this cloud
server.

The organization of this paper is as follows: In Section 2, we first review the RABE scheme of Xu et al.
and their security model. Then, in Section 3, we discuss our ciphertext outdate attack to the RABE scheme
of Xu et al. by exploiting the time encoding functions of Xu et al. Finally, we conclude in Section 4.

2 Revocable Attribute-Based Encryption

In this section, we review the RABE scheme of Xu et al. [7] and the security model of their RABE.

2.1 Xu et al.’s Construction

Before explaining the RABE scheme of Xu et al. [7], we first define the time encoding functions proposed
by them. The TEncode function converts a time period $t$ to a bit string $bt$ of length by appending
zero value to the prefix of the bit string. The CEncode function converts a time period $t$ to an encoded
bit string $et$ by finding the first zero value and then converts all remaining values to zero. The definitions of
these two time encoding functions are described follows:

TEncode($t, T$): It takes a decimal number $t$. It encodes $t$ to a bit string $bt$. While $|bt| < \log_2 T$, it performs
$bt = 0 || bt$. It returns the bit string $bt$.

CTEncode($t, T$): It takes a decimal number $t$. Let $[k]$ be the set $\{1, 2, \ldots, k\}$. It first sets an encoded string
$et$ as empty one. It next obtains a bit string $bt$ by running TEncode($t, T$) and sets $chk = false$. For
each $j \in [\log_2 T]$, it performs the following steps: if $bt[j] = 1$ and $chk = false$, then it sets $et[j] = 1$;
otherwise it sets $chk = true$ and $et[j] = 0$. It returns the encoded string $et$. 
For example, we let the maximum time is $T = 2^5$, and two time periods are $t = 5$ and $t^* = 7$. In this case, the function $\text{TEncode}(t = 5, T)$ returns $bt = 00101$, the function $\text{TEncode}(t^* = 7, T)$ returns $bt^* = 00111$, and the function $\text{CTEncode}(t^* = 7, T)$ returns $et^* = 00000$.

The RABE scheme of Xu et al. follows the existing design methodology of previous RABE schemes that combines an ABE scheme, a tree-based broadcast scheme, and an IBE scheme in bilinear groups. In addition, Xu et al. have changed the structure of ciphertext to provide ciphertext update functionality by devising a new ciphertext encoding method. To use a tree-based broadcast scheme, two additional functions $\text{Path}$ and $\text{KUNodes}$ should be defined. The function $\text{Path}$ returns a set of nodes in a binary tree that are in the path from the root node to the specified leaf node, and the function $\text{KUNodes}$ returns a set of nodes that are root nodes of sub-trees where the leaf nodes of all sub-trees can cover the set of all non-revoked leaf nodes in the binary tree. It is required that if a leaf node $v$ is not revoked then $\text{Path}(v) \cap \text{KUNode}(RL) \neq \emptyset$ and if $v$ is revoked then $\text{Path}(v) \cap \text{KUNode}(RL) = \emptyset$ where $RL$ is the set of revoked leaf nodes. We omit the detailed descriptions of $\text{Path}$ and $\text{KUNodes}$ functions. For the more detailed definition of these functions, see the work of Boldyreva et al. The RABE scheme of Xu et al. is described as follows:

**Setup** $(1^k, N, T, n)$: Let $\lambda$ be the security parameter, $N$ be the maximum number of users, $T$ be the bounded system life time, and $n$ be the maximum number of attributes. It obtains a bilinear group $(g, \mathbb{G}, \mathbb{G}_T, e)$ by running $\mathcal{G}(1^k)$ where $p$ is prime order of the groups. Let $g$ be a generator of $\mathbb{G}$. It selects a random exponent $\alpha$ and sets $g_1 = g^\alpha$. It also chooses random elements $g_2, \{T_i\}_{i \in [n+1]}, U_0, \{U_j\}_{j \in [\log_2 T]} \in \mathbb{G}$ and defines $T(x) = g_2^{\sum_{i=1}^{n+1} T_i^{\lambda_i(x)}}$ where $\Delta_i(x) = \prod_{j \in J, j \neq i} T_j^{-\lambda_j(x)}$. It sets a binary tree $BT$ with at least $N$ number of leaves. Finally, it outputs a revocation list $RL = \emptyset$, a state $ST = BT$, a master key $MK = \alpha$, and public parameters $PP = ((g, \mathbb{G}, \mathbb{G}_T, e), g, g_1, g_2, \{T_i\}_{i \in [n+1]}, U_0, \{U_j\}_{j \in [\log_2 T]}).

**GenKey(id, A, MK, ST, PP)**: Let $id$ be an identity and $A = (M, p)$ be an access policy for attributes where $M$ is a $d \times \ell$ matrix. It assigns the user identity $id$ to a leaf node $\theta \in BT$. For each node $x \in \text{Path}(\theta)$, it performs the following steps: 1) It fetches $\alpha_x$ from the node $x$. If $\alpha_x$ is not defined before, then it chooses a random $\alpha_x \in \mathbb{Z}_p$ and stores it in the node $x$. 2) Let $\bar{u}$ be a random $\ell$ dimensional vector over $\mathbb{Z}_p$ such that $1 \cdot \bar{u} = \alpha_x$. For each row $i$ in the matrix $M$, it chooses a random exponent $r_i$ and sets a partial private key $PSK_{id,x} = (\{K_{i,0} = g_2^{M_i \cdot \bar{u} T(i)^r}, K_{i,1} = g_0^r\}_{i \in [id]}).$ Finally, it outputs a private key $SK_{id} = \{PSK_{id,x} \}_{x \in \text{Path}(\theta)}$ and an updated state $ST = BT$.

**UpdateKey(t, RL, MK, ST, PP)**: Let $t$ be a revocation epoch and $RL$ be the revocation list. It obtains a bit string $bt$ by running $\text{TEncode}(t, T)$. Let $\mathbb{V}_t$ be the set of all $j$ for which $bt[j] = 0$. For each node $x \in \text{KUNodes}(BT, RL, t)$, it performs the following steps: 1) It fetches $\alpha_x$ from the node $x$. If $\alpha_x$ is not defined before, then it chooses a random $\alpha_x \in \mathbb{Z}_p$ and stores it in the node $x$. 2) It chooses a random exponent $r$ and obtains a partial key update $PKU_{i,x} = (U_0 = g_2^{\alpha_x - \alpha_j} (U_0 \prod_{j \in \mathbb{V}_t} U_j)^{\theta}, U_1 = g^{\theta})$. Finally, it outputs a key update $KU_{i} = \{PKU_{i,x} \}_{x \in \text{KUNodes}(BT, RL, t)}$.

**DeriveDK(SK_{id}, PKU_{i, PP})**: Let $SK_{id} = \{PSK_{id,x} \}_{x \in \text{Path}(\theta)}$ and $KU_{i} = \{PKU_{i,x} \}_{x \in \text{KUNodes}(BT, RL, t)}$. If $\text{Path}(\theta) \cap \text{KUNodes}(BT, RL, t) = \emptyset$, then it outputs $\bot$. Otherwise, it finds a unique node $x \in \text{Path}(\theta) \cap \text{KUNodes}(BT, RL, t)$ and retrieves $PSK_{id,x}$ and $PKU_{i,x}$ for the node $x$ from $SK_{id}$ and $KU_{i}$ respectively. Finally it outputs a decryption key $DK_{id,j} = \{PSK_{id,x} \}_{x \in \text{Path}(\theta)}$.

**Encrypt(S, t, m, PP)**: Let $S$ be an attribute set, $t$ be time, and $m$ be a message. It obtains an encoded string $et$ by running $\text{CTEncode}(t, T)$. Let $\mathbb{V}_t$ be the set of all $j$ for which $et[j] = 0$. It chooses a random exponent $s \in \mathbb{Z}_p$ and outputs an original ciphertext $CT_i = (C = e(g_1, g_2)^{s} \cdot m, C_1 = g^{s}, \{C_{2,i} = T(i)^{s}\}_{p(i) \in S}, E_1 = U_0^{s}, \{E_{2,j} = U_j^{s}\}_{j \in \mathbb{V}_t})$. 

3
We describe the security model of the RABE scheme as defined by Xu et al. [7]. The selective IND-RABE-CPA security is defined as the following game between a challenger C and an adversary A:

**Init:** A first submits a challenge attribute set S*.

**Setup:** C generates an empty revocation list RL, a state ST, a master key MK, and public parameters PP by running the setup algorithm Setup(λ, N, T, n), and then it gives PP to A.

**Phase 1:** A may adaptively request private key, key update, and revocation queries to the following oracles.

- The private key generation oracle takes an identity id and an access structure A as input, and returns a private key SKid by running GenKey(id, A, MK, ST, PP).

- The key update oracle takes time t as input, and returns a key update KUt by running UpdateKey(t, RL, MK, ST, PP).

- The revocation oracle takes a revoked identity id and time t as input, and updates the revocation list by running Revoke(id, t, RL, ST).

**Challenge:** A submits challenge time t* ∈ T and two challenge messages m0, m1 of the same size with the following constraints:
Lemma 3.1. Let st be a bit string in \{0, 1\}^{|T|} such that \(A(S^*) = 1\) was queried to the private key generation oracle, then the revocation of the identity id must be queried on time \(t\) such that \(t \leq t^*\) to the revocation oracle.

- If a non-revoked user with the identity id whose access structure \(A\) satisfies the challenge attribute set \(S^*\), then id should not be previously queried to the private key generation oracle.

\(C\) flips a random bit \(b \in \{0, 1\}\) and creates a challenge ciphertext \(CT^*\) by running \(\text{Encrypt}(S^*, t^*, m^*_b, PP)\).

**Phase 2:** \(A\) continues to request private key, key update, and revocation queries. \(C\) handles the queries as the same as before and following the restrictions defined in the challenge phase.

**Guess:** Finally \(A\) outputs a bit \(b'\).

An RABE scheme is selectively IND-RABE-CPA secure if for any probabilistic polynomial time adversary \(A\), the advantage of \(A\) in the above RABE game defined as \(\text{Pr}[b = b'] - \frac{1}{2}\) is negligible in the security parameter \(\lambda\).

### 3 Ciphertext Outdate Attack

In this section, we show that there is an effective adversary against the RABE scheme of Xu et al. [7]. To do this, we first analyze the properties of two time encoding functions, \(\text{TEncode}\) and \(\text{CTEncode}\), proposed by Xu et al. through the following two lemmas. The key to the following two lemmas is that a challenge original ciphertext associated with challenge time \(t^*\) can be changed to a ciphertext element associated with the past time \(t\).

**Lemma 3.1.** Let \(st\) be a bit string in \(\{0, 1\}^{|T|}\) and \(V_{st}\) be the set of all \(j\) such that \(s[t[j] = 0\). There exist time periods \(t, t^* \in T\) such that \(t < t^*\) and \(V_{bt} \subseteq V_{et^*}\) where \(bt\) is obtained from \(\text{TEncode}(t, T)\) and \(et^*\) is obtained from \(\text{CTEncode}(t^*, T)\).

**Proof.** For the notational simplicity, we set \(T = 2^{|T|}\). To prove this lemma, we first randomly choose time periods \(t, t^*\) satisfying \(0 < t < t^* < 2^{|T|} - 1\). Then, we run \(\text{TEncode}(t, T)\) to get a bit string \(bt \in \{0, 1\}^{\tau}\) and \(\text{TEncode}(t^*, T)\) to get another bit string \(bt^* \in \{0, 1\}^{\tau}\). Since the time periods \(t\) and \(t^*\) are smaller than \(2^{|T|} - 1\), the first bit value \(bt[1]\) and \(bt^*[1]\) of the two bit strings \(bt\) and \(bt^*\) have the same bit 0. Now let's analyze the bit string \(et^*\) obtained by running \(\text{CTEncode}(t^*, T)\). In the \(\text{CTEncode}\) algorithm, the algorithm finds the first position with a bit value of 0 in the \(bt^*\) bit string and then sets all subsequent bit values to a value of zero. Thus, the resulting bit string \(et^*\) becomes a bit string with 0 value in all positions since \(bt^*[1] = 0\) is already fixed. Therefore, the set \(V_{et^*}\) consists of \(\{1, 2, \ldots, \tau\}\) and the set of \(V_{bt}\) should be a subset of \(\{1, 2, \ldots, \tau\}\) since \(0 < t\).

As an example, let us look at the encoding results for two time periods \(t = 5\) and \(t^* = 7\) when the maximum time is \(T = 2^5\). Since \(T = 2^5\), the function \(\text{TEncode}(t = 5, T)\) returns a bit string \(bt = 00101\), the function \(\text{TEncode}(t^* = 7, T)\) returns a bit string \(bt^* = 00111\), and the function \(\text{CTEncode}(t^* = 7, T)\) returns the bit string \(et^* = 00000\). Because \(t = 5 < t^* = 7\) and \(V_{bt} = \{1, 2, 4\} \subseteq V_{et^*} = \{1, 2, 3, 4, 5\}\), we can easily show that two time periods \(t = 5\) and \(t^* = 7\) are one example of Lemma 3.1.

**Lemma 3.2.** If there exist time periods \(t, t^* \in T\) such that \(t < t^*\) and \(V_{bt} \subseteq V_{et^*}\) where \(bt\) is obtained from \(\text{TEncode}(t, T)\) and \(et^*\) is obtained from \(\text{CTEncode}(t^*, T)\), then a ciphertext element \(E_t\) for time \(t\) can be derived from an original ciphertext \(CT^*\) for time \(t^*\).
Proof. As the same as in Lemma\[3.1] we randomly choose time periods \( t \) and \( t^* \) to satisfy \( 0 < t < t^* < 2^{\tau-1} \). In the RABE scheme of Xu et al., the original ciphertext \( CT^* \) for the time \( t^* \) includes ciphertext elements \( E_1 \) and \( E_{2,j} \) for all \( j \in V_{et^*} \). As shown in the previous Lemma\[3.1] the set \( V_{et^*} \) is defined as a set of all indices from 1 to \( \tau \) since all bit values of \( et^* \) are composed of 0. In the description of the UpdateCT algorithm, the ciphertext element \( E_i \) can be derived by composing the elements \( E_1 \) and \( E_{2,j} \) for all \( j \in V_{bt} \) of \( CT^* \). Therefore, it is possible to construct the element \( E_i \) since \( V_{bt} \subseteq V_{et^*} \) is satisfied by Lemma\[3.1] \( \Box \)

By using the previous two lemmas, we show that a cloud server which stores original ciphertexts generated by data owner can obtain sensitive information of the original ciphertexts by gathering revoked credentials of revoked users.

Theorem 3.3. There exists a probabilistic polynomial-time adversary that can break the selective IND-RABE-CPA security of Xu et al.’s RABE scheme.

Proof. The main idea of our attack is for an inside adversary, which is a cloud server, to derive an outdated ciphertext \( CT_t \) of past time \( t \) from the original challenge ciphertext \( CT^* \) of challenge time \( t^* \) such that \( t < t^* \). If the adversary gathers revoked credentials of revoked users from the Internet, then it can decrypt the outdated ciphertext by combining the revoked credential with publicly available key updates.

A detailed adversary algorithm \( \mathcal{A} \) breaking the RABE scheme of Xu et al. is described as follows:

1. Initially \( \mathcal{A} \) submits a challenge attribute set \( S^* \) and receives public parameters \( PP \).

2. After that, \( \mathcal{A} \) gathers revoked credentials (private keys) of revoked users from the Internet. Note that revoked credentials of users can be available in Internet since they are safely revoked by the system although these credentials are intentionally revealed by an hacker or accidentally revealed by a honest user. Let \( SK_{id^*} \) be one of the obtained revoked private keys with an identity \( id^* \) and an access policy \( A \) that satisfies \( A(S^*) = 1 \). Because this private key \( SK_{id^*} \) is revoked by a trusted center, we have that it is revoked on time \( t^* \) in a key update \( KU_{t^*} \), but not yet revoked in a key update \( KU_t \) such that \( t < t^* \). \( \mathcal{A} \) sets past time \( t \) and challenge time \( t^* \) that satisfies the condition described in Lemma\[3.1]

3. Next \( \mathcal{A} \) also gathers key updates from the Internet since each key update \( KU_t \) is publicly broadcasted by the trusted center per each time period \( t \). It now derive a decryption key \( DK_{id^*,j} \) by combining the gathered private key \( SK_{id^*} \) and the gathered key update \( KU_t \) since \( SK_{id^*} \) was not yet revoked on past time \( t < t^* \). Note that it cannot derive \( DK_{id^*,j} \) for the challenge time \( t^* \) because \( SK_{id^*} \) was already revoked on time \( t^* \).

4. In the challenge step, \( \mathcal{A} \) submits the challenge time \( t^* \), randomly chosen two challenge messages \( m^*_0, m^*_1 \), and receives a challenge original ciphertext \( CT^* \). Let \( CT^* = \langle C, C_1, \{C_{2,i}\}, E_1, \{E_{2,j}\}_{j \in V_{et^*}} \rangle \) be the original ciphertext on the time \( t^* \) where \( et^* \) is obtained from CTEncode\( (t^*, T) \). Consider the set \( V_{bt} \) for the fixed past time \( t \). The ciphertext element \( E_i \) can be derived from the original ciphertext \( CT^* \) by Lemma\[3.2] because \( V_{bt} \subseteq V_{et^*} \) is satisfied. Thus \( \mathcal{A} \) can easily derive an outdated ciphertext \( CT_t = \langle C, C_1, C_{2,i}, E_i \rangle \) associated with the challenge attribute set \( S^* \) and the past time \( t \) by performing re-randomization.

5. Finally, \( \mathcal{A} \) obtains the message \( m^* \) by decrypting \( CT_t \) using \( DK_{id^*,j} \) and outputs a bit \( b' \) by comparing \( m^* \) with the challenge messages.

Now we analyze the success probability of the adversary \( \mathcal{A} \) described above. As shown above, the decryption succeeds because the behavior of \( \mathcal{A} \) satisfy the constraints of the security model and the outdated
ciphertext is also a valid ciphertext with the correct distribution. Therefore, $A$ wins the RABE security game since the advantage of $A$ is $1/2$.

3.1 Discussions

In this section, we consider two possible defences against the ciphertext outdate attacks.

**Weaker Security Model.** The reason for the above attack is that a cloud server can decrypt a stored original ciphertext by deriving an outdated ciphertext from the original ciphertext and using gathering revoked credentials of revoked users. A naive approach to prevent this devastating attack is to weaken the security model of RABE for cloud storage. In other words, if an inside attacker such as a cloud server performs an attack, then the inside attacker should be prohibited to gathering revoked credentials of revoked users from the Internet. However, this weaker security model only provides a limited security since it is very hard to forbid the inside attacker to (passively) gathering some useful information from the Internet [4].

**Self-Updatable Encryption.** Sahai et al. [6] first devised a new encryption scheme that supports ciphertext updates in cloud storage by using the delegation feature of ABE. After that, Lee et al. [3] proposed a self-updatable encryption scheme by combining a binary tree with the key delegation feature of hierarchical identity-based encryption. A self-updatable encryption scheme provides the most efficient ciphertext update and it is proven to be secure against collusion attacks. In order to enhance the security of Xu et al.’s RABE scheme, their RABE scheme can be modified to use a self-updatable encryption scheme instead of using time encoding functions. In addition, if a self-updatable encryption scheme is used, a data owner does not need to use a secure channel when he sends a ciphertext to cloud storage.

4 Conclusion

In this paper, we showed that a cloud server can perform the ciphertext outdate attack to the RABE scheme of Xu et al. This attack was possible because the cloud server is not a fully trusted entity and it can derive another ciphertext associated with past time from the original ciphertext stored by a user. Although Xu et al. showed that their RABE scheme is secure against outside attackers, they didn’t consider the cloud server can be an inside attacker in the security proof. One naive defence against this attack is to consider a weaker security model where an insider attacker who has access to the original ciphertext cannot obtain revoked credentials of users by preventing the inside attacker from accessing to the Internet. A better defence for this attack is to use the self-updatable encryption scheme of Lee et al. [3] for efficient ciphertext updating since an RABE scheme that uses a self-updatable encryption scheme can be secure against collusion attacks.

References

[1] Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. Identity-based encryption with efficient revocation. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM Conference on Computer and Communications Security - CCS 2008*, pages 417–426. ACM, 2008.

[2] Seny Kamara and Kristin E. Lauter. Cryptographic cloud storage. In Radu Sion, Reza Curtmola, Sven Dietrich, Aggelos Kiayias, Josep M. Miret, Kazue Sako, and Francesc Sebé, editors, *Financial Cryptography and Data Security - FC 2010*, volume 6054 of *Lecture Notes in Computer Science*, pages 136–149. Springer, 2010.
[3] Kwangsu Lee, Seung Geol Choi, Dong Hoon Lee, Jong Hwan Park, and Moti Yung. Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, volume 8269 of *Lecture Notes in Computer Science*, pages 235–254. Springer, 2013.

[4] Raluca A. Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. CryptDB: protecting confidentiality with encrypted query processing. In Ted Wobber and Peter Druschel, editors, *SOSP 2011*, pages 85–100. ACM, 2011.

[5] Mark Dermot Ryan. Cloud computing security: The scientific challenge, and a survey of solutions. *Journal of Systems and Software*, 86(9):2263–2268, 2013.

[6] Amit Sahai, Hakan Seyalioglu, and Brent Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 199–217. Springer, 2012.

[7] Shengmin Xu, Guomin Yang, Yi Mu, and Robert H. Deng. Secure fine-grained access control and data sharing for dynamic groups in the cloud. *IEEE Trans. Inf. Forensic Secur.*, 13(8):2101–2113, 2018.