Constant-size Threshold Attribute based SignCryption for Cloud Applications

Sana Belguith[1], Nesrine Kaaniche[2], Maryline Laurent[2], Abderrazak Jemai[3] and Rabah Attia[1]
[1]*Laboratory of Electronic Systems and Communication Network, Tunisia Polytechnic School, Tunis, Tunisia*
[2]*SAMOVAR, CNRS, Telecom SudParis, University Paris-Saclay, Member of the Chair Values and Policies of Personal Information, Paris, France*
[3]*Laboratory LIP2, University of Sciences of Tunis, Tunis, Tunisia*

Keywords: Attribute based Signcryption, Public Clouds, Privacy, Confidentiality, Access Control, Anonymous Data Origin Authentication.

Abstract: In this paper, we propose a novel constant-size threshold attribute-based signcryption scheme for securely sharing data through public clouds. Our proposal has several advantages. First, it provides flexible cryptographic access control, while preserving users' privacy as the identifying information for satisfying the access control policy are not revealed. Second, the proposed scheme guarantees both data origin authentication and anonymity thanks to the novel use of attribute based signcryption mechanism, while ensuring the unlinkability between the different access sessions. Third, the proposed signcryption scheme has efficient computation cost and constant communication overhead whatever the number of involved attributes. Finally, our scheme satisfies strong security properties in the random oracle model, namely Indistinguishability against the Adaptive Chosen Ciphertext Attacks (IND-CCA2), Existential Unforgeability against Chosen Message Attacks (EUF-CMA) and privacy preservation of the attributes involved in the signcryption process, based on the assumption that the augmented Multi-Sequence of Exponents Decisional Diffie-Hellman (aMSE-DDH) problem and the Computational Diffie Hellman Assumption (CDH) are hard.

1 INTRODUCTION

Nowadays, technological advances relieve an important growth of digital contents which rises the demand for new storage and network capacities and increases the need for more cost-effective use of storage and network bandwidth for data transfer. Thus, cloud storage systems are gaining an expanding interest thanks to their profitable architecture which supports transmission, storage in a multi-tenant environment, and intensive computation of outsourced data in a pay per use business model. However, cloud benefits are spoiled with significant and persistent security concerns that come primarily from the loss of data control (Carlin et al., 2015). That is, users are called to store their sensitive information on remote servers.

These security concerns lead us to the necessity of designing comprehensive access control mechanism for outsourced data while ensuring data confidentiality, data authenticity and protecting users' privacy. Traditional cryptosystems were designed to confidentially encrypt data to a target recipient, however, in a cloud environment, this may restrict the range of opportunities and flexibility offered by cloud systems. Hence, several access control mechanisms were introduced to ensure data security in outsourced systems (Bacis et al., 2016; di Vimercati et al., 2016; di Vimercati et al., 2014; Kaaniche et al., 2014; Belguith et al., 2016; Belguith et al., 2015).

Signcryption, first introduced by Zheng (Zheng, 1997), is an important cryptographic primitive to ensure data confidentiality and authenticity. Signcryption is a logical combination of encryption, providing confidentiality, and signature, supporting data authenticity and integrity, in a single primitive. This novel concept offers a reduced cost, in terms of computation and communication complexity, compared to the cumulative cost of encryption and signature. Attribute Based Signcryption (ABSC) which combines the functions of Attribute Based Encryption (ABE) (Bethencourt et al., 2007) and Attribute Based Signature (ABS) (Maji et al., 2011) is appropriate to ensure fine-grained data access control in large-scale distributed environments mainly cloud storage envi-

Belguith, S., Kaaniche, N., Laurent, M., Jemai, A. and Attia, R.
Constant-size Threshold Attribute based SignCryption for Cloud Applications.
DOI: 10.5220/0006469202120225
In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECRYPT*, pages 212-225
ISBN: 978-989-758-259-2

ronments. Cloud users are able to outsource their data to cloud servers and achieve efficient data sharing with other users. In this scenario, ensuring confidentiality of outsourced data in remote servers may not be enough considering that cloud clients should also be able to anonymously authenticate with the cloud server, in order to avoid unauthorized storage and modification of data, while preserving their privacy.

In this paper, we propose a novel constant-size attribute based signcryption scheme that ensures flexible access control and anonymous data access to outsourced data, with respect to a threshold access policy. Our proposed mechanism has several advantages. First, the proposed scheme ensures flexible access control to data which preserves the users' privacy while authenticating with the remote server. In addition, it provides anonymous data origin authentication in order to ensure that outsourced data are uploaded and modified by an authorised user. Second, the size of the signcrypted message does not depend on the number of attributes involved in the access policy, which makes our scheme more suitable for bandwidth-limited applications. Contrary to most-existing schemes, the size of ciphertext/signature is linear to the number of signing and encrypting attributes used in the signcryption process. Thus, our proposal is highly scalable and offers interesting performances, at both the client and the cloud provider sides. Third, our scheme satisfies strong security properties in the random oracle model, namely Indistinguishability against the Adaptive Chosen Ciphertext Attacks (IND-CCA2), Existential Unforgeability against Chosen Message Attacks (EUF-CMA) and privacy preservation of the attributes involved in the signcryption process.

Paper Organisation – The remainder of this work is as follows: Section 2 highlights security considerations and design goals. Then, Section 3 reviews related work and introduces attribute based signcryption mechanisms. In Section 4, we describe the system model, as well as the security model. Afterwards, we detail the framework design and present our concrete construction in Section 5. Section 6 presents a rigorous security discussion. Finally, a theoretical analysis of computational performances is presented in Section 7, before concluding in Section 8.

2 PROBLEM STATEMENT

As cloud computing is achieving increased popularity, several organisations are using the cloud architecture in a collaborative way, for their employees and/or collaborators to share and collaboratively update a document in progress. In case of a multinational company having different sites all over the world, employees and collaborators can be organized into different dynamic working groups where each member of the working group can access documents, modify their contents and propose suggestions. Hence, users access rights have to be specified, for defining which users can access which data item(s).

Beyond the need for a flexible access control to data, it is necessary to provide data origin authentication as documents have to be checked as coming from a reliable source. The idea is to counteract illegal data storage by permitting only the authorized users to upload or update the data files. As such, members of the working group have to be able to check that the document originates from one of the members. Also, privacy preservation of the members is of importance, especially in the following scenario. A Human Resources (HR) department wants to publish a survey about the employees satisfaction of the company's strategy. The HR manager then uploads a file on the cloud storage platform and shares the file with the employees. For the employees to feel free to post opinions and suggestions on the document, there is a high interest in that case to support anonymity of the members, to avoid the members, the HR manager or the cloud provider to identify the origin of a comment. Finally, as documents are outsourced to a cloud storage service, and the cloud provider can not be fully trusted, it is of high importance to support data confidentiality against curious storage cloud providers. This collaborative scenario can be used in different fields, namely, industrial sectors (such as multinational companies), medical organisations (e.g. medical research groups), social networks$\cdots$.

To support all these features with efficiency, we propose to design a threshold attribute-based signcryption (t-ABSC) to be run at the client side. Indeed, in collaborative use cases, threshold access structures are much more appropriate than monotone access policies. Indeed, they are much more malleable, such that, an authorized user has to only satisfy t attributes from n . The idea is first to use the attribute-based concept to support the flexible sharing of outsourced data among a dynamic group of users, and second to benefit from the efficiency of the signcryption concept as the data secrecy and authenticity mechanisms are partly merged, contrary to the heavier traditional way considering separate signature and encryption operations.

Thus, the design of solution is motivated by providing the support of both robustness and efficiency while fulfilling the following properties:

- ***Data Confidentiality –*** our t-ABSC scheme has to protect the secrecy of outsourced and encrypted data contents against curious users.
- ***Flexible Access Control –*** our proposal should ensure flexible security policies among dynamic groups of users.
- ***Data Origin Authenticity –*** the proposed scheme has to guarantee the data origin anonymous authentication property, while ensuring the unlinkability between the different access sessions.
- ***Privacy –*** our solution must protect group members' access patterns privacy, while creating, uploading and requesting access to outsourced data. That is, the cloud server must be able to grant access with no need to additional identifying information of the requesting users.
- ***Low Computation Overhead and Storage Cost –*** the proposed algorithms should also have low processing complexity and acceptable storage cost, at both cloud provider side and client side.

3 ATTRIBUTE BASED SIGNCRYPTION SCHEMES

In 1997, Zheng et al. (Zheng, 1997) proposed the signcryption concept as a primitive that combines the functions of digital signature and encryption in a single logical step. Afterwards, several signcryption schemes have been proposed either in public key infrastructure or for Identity Based Encryption settings (Libert and Quisquater, 2004; Dent et al., 2010).

Haber et al. (Haber and Pinkas, 2001) have first proposed the idea of combining a public encryption scheme and a signature scheme to have the common public parameters and the key. However, in this concept, the Encrypt and Decrypt (resp. Sign and Verify) of the encryption (resp. signature) schemes were kept unchanged. Later, Vasco et al. (Vasco et al., 2008) proposed a combination of the Identity based Scheme and the Identity based Signature in a joint security model. In 2010, Gagné et al. proposed Attribute based SignCryption (ABSC) with a threshold structure (Gagne et al., 2010). In fact, this ABSC scheme is efficient compared with the encrypt-then-sign paradigm. As in CP-ABE, an encrypting entity can specify the access structure for decrypting entities, and as in ABS, each decryption entity can verify the encrypting entity's attributes. Note that the Gagné et al. definition does not consider the signer's attribute privacy. This permits the decryption entity to verify the encryption entity's attribute explicitly. However, this scheme is based on the use of two different sets of signing attributes and sets of decryption attributes as well as using two access structures related to signature and encryption. As such, the size of the generated signcrypted message increases linearly with the number of attributes, associated to the signing and encryption policies In (Emura et al., 2012), Keita et al. proposed an ABSC scheme with dynamic property, where the access structures of encrypting entity can be changed without reissuing secret keys of users. This signcryption scheme generates a signcrypted message whose size increases linearly with the number of attributes used in the access structures. Recently, Liu et al. (Liu et al., 2015) introduced an attribute based signcryption inspired from the ciphertext-policy attribute based encryption introduced by Waters (Waters, 2011) and the attribute based signature proposed by Maji et al. (Maji et al., 2011). Their proposal is used to secure personal health records in cloud storage. This scheme is based on the use of two different sets of signing attributes and sets of decryption attributes as well as using two access structures related to signature and encryption. As such, the size of the generated signcrypted message increases linearly with the number of attributes, associated to the signing and encryption policies.

The communication and computation overhead as well as the bandwidth consumption in the existing ciphertext policy attribute based signcryption (Gagne et al., 2010; Emura et al., 2012; Liu et al., 2015) schemes increase linearly with the number of attributes required in the access policies. In 2014, Rao et al. (Rao and Dutta, 2014) proposed the first key-policy attribute based signcryption scheme with constant ciphertext size (Rao and Dutta, 2016). However, in the key-policy attribute based schemes, the data owners cannot decide on who has access to their encrypted data, except by their choice of descriptive attributes for the data, since the access policy is embedded in the user private keys. As a result, the data owners have to trust the key issuer. Ciphertext-policy attribute based schemes remove such inconvenience by directly embedding the access policy on the ciphertext. Thus, Ciphertext-policy attribute based schemes are much more appropriate to data outsourcing. This motivates us to address the problem of constructing a ciphertext policy attribute based signcryption scheme which introduces a signcrypted message with a constant size and constant computation cost for flexible access control, data confidentiality and anonymous data origin authenticity.

Table 1 details the computation and storage costs of different attribute based signcryption schemes. We denote by $n_s = |A_s|$ and $n_e = |A_e|$ where A_s and A_e

present the set of attributes of the user obtained from the key extraction procedure, for signing and encrypting respectively. For common setup process, we denote by $n = |A|$ the cardinal of the set of user's attributes. Let l_s and l_e denote the size of the signing policy and the encryption policy, respectively. Let t, $O(M)$ and m respectively define the threshold value, the size of the plaintext message M and the cardinal of the attributes' universe $\mathcal{U}$.

4 MODEL DESCRIPTION

In this section, we first present the system model of our constant-size (t,S)-attribute based signcryption scheme. Then, we detail our security model.

4.1 System Model

We suppose that the signcrypting entity chooses a subset S from the universe of attributes $\mathcal{U}$ and a threshold t such that $1 \leq t \leq |S|$ to define his (t,S) access policy. Then, he signcrypts the message $M \in \mathcal{M}$ (i.e; $\mathcal{M}$ is the message space) with respect to the policy (t,S).
Our constant-size threshold attribute based signcryption consists of four randomized algorithms: `Setup`, `KeyGen`, `SignCrypt` and `UnsignCrypt`, defined as follows:

$\texttt{Setup}(\xi) \rightarrow (PP, msk)$ – the setup algorithm is performed by the central trusted authority. It takes as input a security parameter ξ. The `Setup` algorithm outputs the public parameters PP and the secret master key msk.

$\texttt{KeyGen}(\texttt{PP}, msk, E_i, A_i) \rightarrow sk_{E_i}$ – this randomized algorithm is executed to derive the secret keys of the user E_i. Given the public parameters PP, an attribute set $A_i \subset \mathcal{U}$ (i.e; $\mathcal{U}$ is the attribute universe) of the user E_i and the secret master key msk. The algorithm outputs the user's secret key sk_{E_i} associated to the attribute set A_i.

$\texttt{SignCrypt}(\texttt{PP}, sk_{E_S}, A_S, (t,S), M) \rightarrow \Sigma$ – the signcryption algorithm is performed by a signcrypting entity E_S. It takes as inputs the public parameters PP, the user secret key sk_{E_S}, the set of signing attributes A_S, the (t,S) threshold access policy and the message M. This algorithm outputs the signcrypted message M referred to as Σ.

$\texttt{UnsignCrypt}(\texttt{PP}, sk_{E_U}, A_U, (t,S), \Sigma) \rightarrow M$ – the unsigncryption algorithm is executed by the user E_U. It takes as inputs the public parameters PP, the set of attributes A_U of E_U, the signcrypted message Σ, the access policy (t,S) and the user's private key sk_{E_U}. The algorithm returns the message M if the user E_U has successfully verified the signature output by E_S and has obtained the secret key related to the t required attributes for deciphering the signcrypted message. Otherwise, the algorithm outputs a reject symbol $\perp$.

Our constant-size (t,S)-attribute based signcryption scheme has to satisfy the **correctness property**. The correctness property requires that for all security parameter ξ, all universe descriptions $\mathcal{U}$, all $(\texttt{PP}, msk) \in \texttt{Setup}(\xi)$, all $A_i \subseteq \mathcal{U}$, all domain entities E_i, all
$sk_{E_i} \in \texttt{KeyGen}(\texttt{PP}, msk, E_i, A_i)$, all $M \in \mathcal{M}$, all $(t,S) \in \mathcal{G}$ ($\mathcal{G}$ is the access policy space) and all $\Sigma \in \texttt{SignCrypt}(\texttt{PP}, sk_{E_S}, A_S,$
$(t,S), M)$ (i.e; E_S is the signcrypting entity), if the unsigncrypting user E_U has successfully verified the signature output by E_S and obtained the secret key related to the t required attributes for deciphering the signcrypted message such that $|A_S \cap S| \geq t$ and $|A_U \cap S| \geq t$, the $\texttt{UnsignCrypt}(\texttt{PP}, sk_{E_U}, A_U, (t,S), \Sigma)$ outputs M.

4.2 Security Model

In this section, we consider two threat models for proving security and privacy properties of our proposed constant-size (t,S)-attribute based signcryption scheme. We first consider the case of a *honest but curious* cloud provider. That is, the cloud server is honest as it provides proper inputs or outputs, at each step of the protocol, properly performing any calculations expected from it, but it is curious in the sense that it attempts to gain extra information from the protocol. As such, we consider the honest but curious threat model against the confidentiality and privacy requirements with respect to adaptive chosen-ciphertext attacks (IND-CCA2) and the computational privacy of involved attributes.
Second, we consider the case of malicious users trying to override their rights. That is, malicious users may attempt to deviate from the protocol or to provide invalid inputs. As such, we consider the malicious user security model mainly against the unforgeability requirement considering chosen message attacks (CMA).

4.2.1 Confidentiality

A threshold attribute based signcryption t-ABSC scheme is said to be indistinguishable against non-adaptive chosen ciphertext attacks if there is no probabilistic polynomial time (PPT) adversary that can win the Exp^{conf} security game with non-negligible advantage. The Exp^{conf} security game is formally defined, between an adversary $\mathcal{A}$ and a

Table 1: Features and Functionality Comparison of Attribute Based Signcryption Schemes.

Scheme	*Common Setup*	*Type*	Access Policy	*Key size*	*Signcryption size*
(Emura et al., 2012)	*No*	CP-ABE	Monotone	$2n_s, 2n_e+1+m$	$O(l_e)+O(n_s)$
(Liu et al., 2015)	*No*	CP-ABE	Monotone	n_s+2, n_e+2	$O(l_s)+O(l_e)$
(Gagne et al., 2010)	*No*	CP-ABE	Threshold	$3n_e, 2n_s$	$O(M)+O(n_e)+O(n_s)$
(Rao and Dutta, 2014)	*No*	KP-ABE	Monotone	$m+l_s, m+l_e$	$O(1)$
Our scheme	*Yes*	CP-ABE	Threshold	$n+m+1$	$O(1)$

challenger $\mathcal{C}$ as follows:

Initialisation – in this phase, the adversary $\mathcal{A}$ selects a set of signcryption attributes S^* (i.e; S^* corresponds to the set of attributes specified for the access policy) to be used to signcrypt the challenge ciphertext, and sends S^* to the challenger $\mathcal{C}$.

Setup – the challenger $\mathcal{C}$ runs the $\texttt{Setup}(\xi)$ algorithm of the signcryption scheme and sends the public parameters PP to the adversary $\mathcal{A}$.

Unsigncryption Query Phase 1 – the adversary $\mathcal{A}$ can request, as many times as he wants, the following queries:

- KeyGen – the adversary $\mathcal{A}$ queries, for each session i, a signcryption attribute set $A_{\mathcal{A},i}$ with respect to a threshold t_i where $|A_{\mathcal{A},i} \cap S^*| < t_i$. The challenger $\mathcal{C}$ answers by running the $\texttt{KeyGen}(\texttt{PP}, msk, \mathcal{A}, A_{\mathcal{A},i})$ algorithm and sends the resulting secret key to the adversary $\mathcal{A}$. The secret key is referred to as $sk_{\mathcal{A},i}$.
- UnsignCrypt – the adversary $\mathcal{A}$ requests the unisgncryption of Σ with respect to a threshold t_i, while considering the signcryption attribute set $A_{\mathcal{A},i}$. The challenger $\mathcal{C}$ executes the KeyGen algorithm to generate the secret key $sk_{\mathcal{C},i} = \texttt{KeyGen}(\texttt{PP}, msk, \mathcal{C}, A_{\mathcal{C},i})$, such that $|A_{\mathcal{C},i} \cap S^*| \geq t_i$. Finally, the challenger $\mathcal{C}$ answers the query by running the $\texttt{UnsignCrypt}(\texttt{PP}, sk_{\mathcal{C},i}, A_{\mathcal{C},i}, (t_i, S^*), \Sigma)$ algorithm that outputs a message m or a reject symbol $\perp$.

Challenge Phase – during the challenge phase, the adversary picks two equal length plaintexts ${M_0}^*$ and ${M_1}^*$ and a threshold signcrypting attribute set (t^*, S^*) (i.e; t^* has never been queried during the **Unsigncryption Query Phase 1**) and sends them to the challenger. The challenger $\mathcal{C}$ chooses a random bit b from $\{0,1\}$, extracts a signcrypting secret key $sk_{\mathcal{C}} = \texttt{KeyGen}(\texttt{PP}, msk, A_{\mathcal{C}})$ and computes the challenge signcrypted message ${\Sigma_b}^* = \texttt{SignCrypt}(\texttt{PP}, sk_{\mathcal{C}}, A_{\mathcal{C}}, (t^*, S^*), {M_b}^*)$. Then, the challenger sends ${\Sigma_b}^*$ to the adversary.

KeyGen Query Phase 2 – in this phase, the adversary $\mathcal{A}$ who has already received ${\Sigma_b}^*$, can query a polynomially bounded number of queries as in **Unsigncryption Query Phase 1**, except that the adversary $\mathcal{A}$ can not make an $\texttt{UnsignCrypt}(\texttt{PP}, , sk_{\mathcal{A},i}, A_{\mathcal{A},i}, (t_i, S^*), {\Sigma_b}^*)$ such that $|A_{\mathcal{A},i} \cap S^*| = t^*$, where $t_i < t^*$.

Guess – the adversary tries to guess which message M_i, where $i \in \{0,1\}$ corresponds to the enciphered data ${\Sigma_b}^*$. Thus, $\mathcal{A}$ outputs a bit b' of b and wins the game if $b = b'$. The advantage of the adversary $\mathcal{A}$ in the above game is defined as $Adv_{\mathcal{A}}[Exp^{Conf}(1^{\xi})] = |Pr[b = b'] - \frac{1}{2}|$.

Definition 1. *A t-ABSC scheme satisfies the IND-CCA2 confidentiality property, if the probability $Adv_{\mathcal{A}}[Exp^{unf}(1^{\xi})]$ is negligible for any PPT adversaries.*

4.2.2 Unforgeability

A threshold attribute based signcryption t-ABSC scheme is unforgeable against chosen message attack (EUF-CMA) if there is no probabilistic polynomial time (PPT) adversary that can win the Exp^{unf} security game with non-negligible advantage. The Exp^{unf} security game is formally defined, between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Initialisation – in this phase, the adversary $\mathcal{A}$ selects a set of signcryption attributes S^* (i.e; S^* corresponds to the set of attributes specified for the access policy) to be used to signcrypt the challenge ciphertext, and sends S^* to the challenger $\mathcal{C}$.

Setup – the challenger $\mathcal{C}$ runs the $\texttt{Setup}(\xi)$ algorithm of the scheme and sends the public parameters PP to the adversary $\mathcal{A}$.

Signcrypt Query Phase – the adversary $\mathcal{A}$ can request, as many times as he wants, the following queries:

- KeyGen – the adversary $\mathcal{A}$ queries, for each session i, a signcryption attribute set $A_{\mathcal{A},i}$ with

respect to a threshold t_i where $|A_{\mathcal{A},i} \cap S^*| < t_i$. The challenger $\mathcal{C}$ answers by running the $\texttt{KeyGen}(\texttt{PP}, msk, A_{\mathcal{A},i})$ algorithm and sends the resulting secret key to the adversary $\mathcal{A}$. The secret key is referred to as $sk_{\mathcal{A},i}$.

- `SignCrypt` – the adversary $\mathcal{A}$ requests the signcryption of a message M while considering for each session i, a signcryption attribute set $A_{\mathcal{C},i}$ and a threshold t_i. The challenger $\mathcal{C}$ executes the `KeyGen` to generate the secret key $sk_{\mathcal{C},i} = \texttt{KeyGen}(\texttt{PP}, msk, A_{\mathcal{C},i})$ and then, he executes `SignCrypt` algorithm and returns the signcrypted message $\Sigma_i = \texttt{SignCrypt}(\texttt{PP}, sk_{\mathcal{C},i}, A_{\mathcal{C},i}, (t_i, S^*), M)$ to the adversary $\mathcal{A}$.

Forgery Phase – after the **Signcrypt Query Phase**, the adversary $\mathcal{A}$ outputs a ciphertext Σ^* with respect to the threshold challenge policy (t^*, S^*) (i.e; t^* has never been queried during the **Signcrypt Query Phase** and where $t_i < t^*$). Then, the challenger unsigncrypts Σ^* using the secret key $sk_{\mathcal{C}}$ and obtains a message M^*. The adversary $\mathcal{A}$ wins the game if Σ^* is valid and was not obtained from a `SignCrypt` query. That is, $\texttt{UnsigncCrypt}(\texttt{PP}, sk_{\mathcal{C}}, A_{\mathcal{C}}, (t^*, S^*), \Sigma^*) = M$ and $\mathcal{A}$ did not issue a $\texttt{SignCrypt}(\texttt{PP}, sk_{\mathcal{C}}, A_{\mathcal{C}}, (t^*, S^*), M)$. The adversary's advantage is defined as $Adv_{\mathcal{A}}[Exp^{unf}(1^{\xi})] = |Pr[Exp^{unf}(1^{\xi})] = 1|$.

Definition 2. *A t-ABSC scheme is unforgeable against chosen-message attack (EUF-CMA), if the probability $Adv_{\mathcal{A}}[Exp^{unf}(1^{\xi})]$ is negligible for all PPT adversaries.*

This unforgeability property also includes the collusion among users trying to override their rights by combining their complementary attributes to generate a signcrypted message satisfying a given access policy (t, S).

4.2.3 Privacy

A threshold attribute-based signcryption scheme is said to be computationally private if any adevrsary $\mathcal{A}$ running in polynomial time can not win the $Exp_{\mathcal{A}}^{Priv}$ security game with non-negligible advantage. The $Exp_{\mathcal{A}}^{Priv}$ security game is formally defined, between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Setup – the adversary $\mathcal{A}$ chooses an attributes universe $\mathcal{U}$ where $|\mathcal{U}| = n$ and sends it to the challenger $\mathcal{C}$. Then, the challenger runs the `setup` algorithm and returns the public parameters to the adversary $\mathcal{A}$.

Challenge Phase – the adversary $\mathcal{A}$ chooses an access policy (t, S) such that $1 \leq t \leq |S| \leq n$, two attribute sets A_{S_1} and A_{S_2} satisfying the threshold access policy such that $|A_{S_1} \cap S| = |A_{S_2} \cap S| = t$, and a message M and sends them to the challenger $\mathcal{C}$. Afterwards, $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$ and executes the `KeyGen` algorithm such that $sk_{\mathcal{C}} = \texttt{KeyGen}(\texttt{PP}, msk, \mathcal{C}, A_{S_b})$ and then, outputs the signcryption $\Sigma_b = \texttt{SignCrypt}(\texttt{PP}, sk_{\mathcal{C}}, A_{S_b}, (t, S), M)$. The signcrypted message Σ_b is sent to the adversary $\mathcal{A}$ as a challenge message.

Guess – the adversary outputs a bit b' and wins the game if $b' = b$.
The advantage of the adversary $\mathcal{A}$ in the above game is defined as $Adv_{\mathcal{A}}[Exp^{Priv}(1^{\xi})] = |Pr[b = b'] - \frac{1}{2}|$.

Definition 3. *A threshold attribute-based signcryption scheme is computationally private if $Adv_{\mathcal{A}}[Exp^{Priv}(1^{\xi})]$ is negligible with respect to the security parameter λ, for any adversary $\mathcal{A}$ running in a polynomial time.*

5 A NOVEL CONSTANT-SIZE THRESHOLD ATTRIBUTE BASED SIGNCRYPTION FOR CLOUD APPLICATIONS

5.1 Overview

In this paper, we develop a new constant size threshold Attribute Based Signcryption Scheme (t-ABSC) as a novel security mechanism for the access control, data encryption, and message authentication in cloud storage environments. Our proposal is based on the use of the constant-size threshold identity based encryption scheme proposed by Delerablee et al. (Delerablée and Pointcheval, 2008) in 2008 and the constant size attribute based encryption proposed by Harranz et al. (Herranz et al., 2010) in 2010, which have been extended to provide all security features as presented in section 2. For instance, our scheme introduces a novel attribute based signature which is combined with the encryption scheme to design a signcryption scheme. Thus, beyond including the attribute based ciphertext, the signcrypted message involves the generated signatures. Moreover, our system supports the common setup procedures (i.e; public parameters and users' keys are same for both encryption and signature processes). In addition, it supports anonymous data-origin authentication and provides flexible control to data.

5.2 Complexity Assumptions

In, our constant size threshold attribute based signcryption construction, we rely on the Computational Diffie Hellman Assumption (CDH) and the augmented multi-sequence of exponents computational Diffie-Hellman ($(\tilde{l},\tilde{m},\tilde{t})$-aMSE-CDH), which was introduced by Delerablée et al. (Delerablée et al., 2007; Delerablée and Pointcheval, 2008) in 2007. These assumptions are defined as follows:

Definition 4. ***Computational Diffie Hellman Assumption (CDH) –*** *Let $\mathbb{G}$ be a group of a prime order p, and g is a generator of $\mathbb{G}$. The CDH problem is, given the tuple of elements (g, g^a, g^b), where $\{a,b\} \xleftarrow{R} \mathbb{Z}_p$, there is no efficient probabilistic algorithm $\mathcal{A}_{CDH}$ that computes g^{ab}.*

Definition 5. ***$(\tilde{l},\tilde{m},\tilde{t})$-augmented multi-sequence of exponents computational Diffie-Hellman ($(\tilde{l},\tilde{m},\tilde{t})$-aMSE-CDH) –*** *The $(\tilde{l},\tilde{m},\tilde{t})$-aMSE-CDH problem related to the group pair $(\mathbb{G},\mathbb{G}_\mathbb{T})$ is to compute $T = e(g_0,h_0)^{k\cdot f(\gamma)}$. It takes as **input**: the vector $\vec{x}_{\tilde{l}+\tilde{m}} = (x_1,\cdots,x_{\tilde{l}+\tilde{m}})^\top$ whose components are pairwise distinct elements of $\mathbb{Z}_p$ which define the polynomials f(X) and g(X) as follows:*

$$f(X) = \prod_{i=1}^{\tilde{l}}(X+x_i); \qquad g(X) = \prod_{\tilde{l}+1}^{\tilde{l}+\tilde{m}}(X+x_i) \quad (1)$$

where the values x_i are random and pairwise distinct of $\mathbb{Z}_p^*$, and the values:

$$\begin{cases} g_0, g_0^{\gamma}, \cdots, g_0^{\gamma^{\tilde{l}+\tilde{t}-2}}, g_0^{k\cdot\gamma\cdot f(\gamma)} \\ g_0^{\omega\gamma}, \cdots, g_0^{\omega\gamma^{\tilde{l}+\tilde{t}-2}} \\ g_0^{\alpha}, g_0^{\alpha\gamma}, \cdots, g_0^{\alpha\gamma^{\tilde{l}+\tilde{t}}} \\ h_0, h_0^{\gamma}, \cdots, h_0^{\gamma^{\tilde{m}-2}} \\ h_0^{\omega}, h_0^{\omega\gamma}, \cdots, h_0^{\omega\gamma^{\tilde{m}-1}} \\ h_0^{\alpha}, h_0^{\alpha\gamma}, \cdots, h_0^{\alpha\gamma^{2(\tilde{m}-\tilde{t})+3}} \end{cases}$$

*Where k,α,γ,ω are unknown random elements of $\mathbb{Z}_p$ and g_0 and h_0 are generators of $\mathbb{G}$. We can solve the problem if we get an **output** $b \in \{0,1\}$ where $b = 1$ if $T = e(g_0,h_0)^{k\cdot f(\gamma)}$ or $b = 0$ when T is a random value from $\mathbb{G}_T$.*

5.3 Aggregate Algorithm

Our scheme construction relies on the aggregate algorithm `Aggreg` introduced by Delerablee et al. (Delerablée et al., 2007; Delerablée and Pointcheval, 2008). This algorithm is explained in the following description:

Let us consider a list of values $\{g^{\frac{r}{\gamma+x_i}}, x_i\}_{1\le i\le n}$, where $r,\gamma \in \mathbb{Z}_p^*$ and $x_1,\cdots,x_n$ are pairwise different. Then , the algorithm proceeds as follows:

$$\texttt{Aggreg}(\{g^{\frac{r}{\gamma+x_i}}, x_i\}_{1\le i\le n}) = g^{\frac{r}{\prod_{i=1}^{n}(\gamma+x_i)}}$$

Concretely, the `Aggreg` algorithm defines $P_{0,m} = g^{\frac{r}{\gamma+x_m}}$ for each $m \in \{1,\cdots,n\}$. Afterwards, the algorithm computes sequentially $P_{i,m}$ for $i = 1\cdots n-1$ and $m = i+1,\cdots,n$ using the induction:

$$P_{i,m} = \left(\frac{P_{i-1,i}}{P_{i-1,m}}\right)^{\frac{1}{x_m - x_i}} \quad (2)$$

Then, we get $P_{i,m} = g^{\frac{r}{(\gamma+x_m)\prod_{k=1}^{i}(\gamma+x_k)}}$ where $1 \le i \le m \le n$.

Therefore, since the elements $x_1,\cdots,x_n$ are pairwise different (Herranz et al., 2010; Attrapadung et al., 2012) and using the equation 2, we can compute $P_{i,m}$ for $i = 1\cdots n-1$ and $m = i+1\cdots n$ such as $P_{n,n-1} = g^{\frac{r}{\prod_{i=1}^{n}(\gamma+x_i)}}$

5.4 Concrete Construction

Our construction is based on the four algorithms defined as follows:

- `Setup` – the trusted authority defines a bilinear group triple $(\mathbb{G}_1,\mathbb{G}_2,\mathbb{G})$ of prime order p, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}$ and a collusion resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$. In addition, It specifies an encoding function τ such that $\tau : \mathcal{U} \to (\mathbb{Z}/p\mathbb{Z})^*$, where $|\mathcal{U}| = n$ and $\mathcal{U}$ is an attribute universe supported by the trusted authority. For each attribute $a \in \mathcal{U}$, the encoded attribute values $\tau(a) = x$ are pairwise different. Then, the `Setup` algorithm selects two generators g and h of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. It also chooses a set of pairwise different elements of $(\mathbb{Z}/p\mathbb{Z})^*$, $\mathcal{D}_i = \{d_1,\cdots,d_i\}$ where $i \le n-1$. Finally, it outputs the global public parameters `PP` defined as follows:
  $\texttt{PP} = \{\mathbb{G}_1,\mathbb{G}_2,\mathbb{G},\hat{e},\{h^{\alpha\gamma^i}\}_{\{i=0,\cdots,2n-1\}},\mathcal{D}_i,\tau,u = g^{\alpha\cdot\gamma}, e(g^{\alpha},h),\mathcal{H}\}$
  We note that the master key of the trusted authority is referred to as $msk = (g,\alpha,\gamma)$ where α,γ are two random from values $(\mathbb{Z}/p\mathbb{Z})^*$.

- `KeyGen` – for any subset $A_i \subset \mathcal{U}$ of attributes associated with a user E_i, the algorithm chooses a random value $r_{E_i} \in (\mathbb{Z}/p\mathbb{Z})^*$ and computes the related secret key as follows:

$$\begin{aligned} sk_{E_i} &= (\{g^{\frac{r_{E_i}}{\gamma+\tau(a)}}\}_{a\in A_i}, \{h^{r_{E_i}\gamma^j}\}_{i=0,\cdots,m-2}, h^{\frac{r_{E_i}-1}{\gamma}}) \\ &= (sk_{E_{i_1}}, sk_{E_{i_2}}, sk_{E_{i_3}}) \end{aligned}$$

- `SignCrypt` – let (t,S) be the access policy where $S \subset U$ is an attribute set of size $s = |S|$ such that $1 \leq t \leq |S|$ and let A_S be the sub-set of attribute set related to the signcrypting entity E_S where $|A_S \cap S| = t$.
  To signcrypt the message M with respect to (t,S), E_S uses his secret key and the aggregate algorithm `Aggreg` introduced in the Section 5.3 and computes T_1 such as:

$$T_1 = \texttt{Aggreg}(\{g^{\frac{r_{E_S}}{\gamma+\tau(a)}}, \tau(a)\}_{a\in A_S}) = g^{\frac{r_{E_S}}{\prod_{a\in A_S}(\gamma+\tau(a))}}$$

  Then, E_S defines the polynomial $P_{(A_S,S)}(\gamma)$ such as:

$$P_{(A_S,S)}(\gamma) = \frac{1}{\gamma}\Big(\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}(\gamma+\tau(a)) - B_1\Big)$$

  Where $B_1 = \prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\tau(a)$

  Afterwards, using the element $sk_{E_{i_2}}$, the signcrypting entity derives B_2 as follows:

$$B_2 = h^{r_{E_S}P_{(A_S,S)}(\gamma)/B_1}$$

  In the sequel, E_S computes $\mathcal{H}(M)$ and generates the signature $\sigma = (\sigma_1,\sigma_2,\sigma_3)$ defined as:

$$\begin{cases} \sigma_1 = T_1 \cdot g^{\frac{\mathcal{H}(M)}{\prod_{a\in A_S}(\gamma+\tau(a))}} \\ \sigma_2 = sk_{E_{S_3}} \cdot B_2 \cdot h^{\mathcal{H}(M)P_{(A_S,S)}(\gamma)/B_1} \\ \sigma_3 = h^{\alpha\cdot\mathcal{H}(M)} \end{cases}$$

  Finally, E_S picks a random $\kappa \in (\mathbb{Z}/p\mathbb{Z})^*$ and computes C_1, C_2 and C_3. Note that the elements C_1, C_2 are computed using the public parameters u and $\hat{e}(g,h)^\alpha$. Moreover, the set $\{h^{\alpha\gamma^j}\}_{i=0,\cdots,2n-1}$ is used to compute C_3 with respect to the equation 1. C_1, C_2 and C_3 are detailed as follows:

$$\begin{cases} C_1 = (g^{\alpha\cdot\gamma})^{-\kappa} \\ C_2 = h^{\kappa\alpha\cdot\prod_{a\in S}(\gamma+\tau(a))\prod_{d\in D_{n+t-1-s}}(\gamma+d)} \\ C_3 = \hat{e}(g,h)^{\alpha\cdot\kappa}\cdot\hat{e}(g,h)^{\alpha\cdot\mathcal{H}(M)}\cdot M = K\cdot M \end{cases}$$

  Finally, the signcrypting entity outputs the signcryption of the message M, $\Sigma = (C_1,C_2,C_3,\sigma_1,\sigma_2,\sigma_3,P_{(A_S,S)}(\gamma),B_1)$.

- `UnsignCrypt` – any user E_U having a set of attributes A_U where $|A_U \cap S| = t$ can verify and decrypt the signcrypted message under the access policy (t,S).
  First, to verify that the received message has been correctly signed by E_S, E_U has to check the following equality:

$$\begin{aligned} \text{Ⓢ} &= \hat{e}(u^{-1},\sigma_2)\cdot\hat{e}(\sigma_1^{\frac{1}{B_1}}, h^{\alpha\cdot\prod_{a\in S\cup D_{n+t-1-s}}(\gamma+\tau(a))}) \\ &\quad\cdot\hat{e}(g^\alpha,h)^{\mathcal{H}(M)} = \hat{e}(g^\alpha,h) \end{aligned}$$

  where $U_1 = \prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\frac{\gamma+\tau(a)}{\tau(a)}$.

  Then, for all $a \in A_U$, E_U has to aggregate the required attributes such as:

$$A_2 = \texttt{Aggreg}(\{g^{\frac{r_{E_U}}{\gamma+\tau(a)}}, \tau(a)\}_{a\in A_U}) = g^{\frac{r_{E_U}}{\prod_{a\in A_U}(\gamma+\tau(a))}} \quad (3)$$

  Afterwards, E_U uses the aggregated secret key A_2 and the $sk_{E_{U_2}}$ key element to compute:

$$[\hat{e}(C_1, h^{r_{E_U}P_{(A_S,S)}(\gamma)})\cdot\hat{e}(A_2,C_2)]^{\frac{1}{B_1}} = e(g,h)^{(\kappa\cdot\alpha)\cdot r_{E_U}}$$

  Then, the unsigncrypting entity E_U deduces the deciphering key K such as:

$$\begin{aligned} K &= \frac{\hat{e}(C_1, sk_{E_{U_2}})\cdot\hat{e}(g,\sigma_3)}{\hat{e}(g,h)^{\kappa\cdot r_{E_U}\cdot\alpha}} = \hat{e}(g,h)^{\alpha\cdot(\mathcal{H}(M)+\kappa)} \\ &= \hat{e}(g,h)^{\alpha\cdot\kappa}\cdot\hat{e}(g,h)^{\alpha\cdot\mathcal{H}(M)} \end{aligned}$$

  Finally, E_U recovers the message by computing $M = \frac{C_3}{K}$.
  Then, to verify the authenticity of the signature of the signcrypting entity, the user E_U uses the retrieved message M to compute $\mathcal{H}(M)$. Afterwards, E_U verifies that $\sigma_3 = h^{\alpha\cdot\mathcal{H}(M)}$

6 SECURITY ANALYSIS

In this section, we prove the correctness and the security of our construction with respect to the security model as defined in section 4.2.

6.1 Correctness

Any user E_U having a set of attributes A_U where $|A_U \cap S| = t$ can verify and decrypt the signcrypted message under the policy (t,S).
In the following, we denote by Ⓢ, the

quantity $\hat{e}(u^{-1},\sigma_2) \cdot \hat{e}(g^{\alpha},h)^{\mathcal{H}(M)\cdot(1-U_1-\frac{1}{U_1})} \cdot \hat{e}(\sigma_1^{\frac{1}{B_1}}, h^{\alpha\cdot\prod_{a\in S\cup D_{n+t-1-s}}(\gamma+\tau(a))})$, where $U_1 = \prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\frac{\gamma+\tau(a)}{\tau(a)}$.
First, E_U has to verify that E_S has correctly signed the message M, such as:

$$
\begin{aligned}
\text{Ⓢ} &= \hat{e}(g^{-\alpha\cdot\gamma}, h^{\frac{r_{E_S}^{-1}}{\gamma}} \cdot h^{\frac{(r_{E_S}+\mathcal{H}(M))\cdot P_{(A_S,S)}(\gamma)}{\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}(\tau(a)+\gamma)}}) \\
&\quad \cdot\hat{e}(g^{\alpha},h)^{\mathcal{H}(M)\cdot(1-U_1-\frac{1}{U_1})} \cdot \\
&\quad \hat{e}(g^{\frac{(r_{E_S}+\mathcal{H}(M))}{\prod_{a\in A_S}(\gamma+\tau(a))\cdot\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\tau(a)}}, \\
&\quad h^{\alpha\cdot\prod_{a\in S\cup D_{n+t-1-s}}(\tau(a)+\gamma)}) \\
&= \hat{e}(g^{-\alpha\cdot\gamma}, h^{\frac{r_{E_S}^{-1}}{\gamma}} h^{r_{E_S}\cdot\frac{P_{(A_S,S)}(\gamma)}{\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}(\tau(a)+\gamma)}}) \\
&\quad \cdot\hat{e}(g^{-\alpha\cdot\gamma}, h^{\mathcal{H}(M)\cdot\frac{P_{(A_S,S)}(\gamma)}{\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\tau(a)}}) \\
&\quad \cdot\hat{e}(g^{\alpha},h)^{\mathcal{H}(M)} \cdot \hat{e}(g^{\alpha},h)^{-\mathcal{H}(M)\cdot U_1} \cdot \hat{e}(g^{\alpha},h)^{\frac{-\mathcal{H}(M)}{U_1}} \\
&\quad \cdot\hat{e}(g^{\frac{r_{E_S}}{\prod_{a\in A_S}(\gamma+\tau(a))\cdot\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\tau(a)}}, \\
&\quad h^{\alpha\cdot\prod_{a\in S\cup D_{n+t-1-s}}(\tau(a)+\gamma)}) \cdot \\
&\quad \hat{e}(g^{\frac{\mathcal{H}(M)}{\prod_{a\in A_S}(\gamma+\tau(a))\cdot\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\tau(a)}}, \\
&\quad h^{\alpha\cdot\prod_{a\in S\cup D_{n+t-1-s}}(\tau(a)+\gamma)}) \\
&= \hat{e}(g^{\alpha},h)^{\mathcal{H}(M)} \cdot \hat{e}(g^{\alpha},h)^{-\mathcal{H}(M)\cdot U_1} \cdot \hat{e}(g^{\alpha},h)^{\frac{-\mathcal{H}(M)}{U_1}} \\
&\quad \hat{e}(g^{\alpha},h)\cdot\hat{e}(g^{\alpha},h)^{\mathcal{H}(M)\cdot\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\frac{\gamma+\tau(a)}{\tau(a)}} \\
&\quad \cdot\hat{e}(g^{\alpha},h)^{-\mathcal{H}(M)\cdot\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\frac{\tau(a)}{\tau(a)}} \\
&\quad \cdot\hat{e}(g^{\alpha},h)^{\frac{\mathcal{H}(M)\cdot\prod_{a\in S\cup D_{n+t-1-s}}(\gamma+\tau(a))}{\prod_{a\in A_S}(\gamma+\tau(a))\cdot\prod_{a\in S\cup D_{n+t-1-s}\setminus A_S}\tau(a)}} \\
&= \hat{e}(g^{\alpha},h)
\end{aligned}
$$

In the sequel, E_U aggregates his secret keys to obtain A_2, as defined in Equation 3, uses the $sk_{E_{U_2}}$ key element and computes:

$$\hat{e}(C_1, h^{r_{E_U}P_{(A_U,S)}(\gamma)})\cdot\hat{e}(A_2,C_2) \quad (4)$$
$$= e(g,h)^{\kappa\cdot\alpha\cdot r_{E_U}\prod_{a\in S\cup D_{n+t-1-s}\setminus A_U}\tau(a)}$$

and

$$\hat{e}(C_1, h^{\frac{r_{E_U}^{-1}}{\gamma}}) = \hat{e}(g,h)^{-\kappa\cdot r_{E_U}\cdot\alpha}\cdot\hat{e}(g,h)^{\kappa\cdot\alpha} \quad (5)$$

Afterwards, to derive the deciphering key K, the user E_U executes Equations 6, 7 and 8 defined as follows:

$$\hat{e}(g,h)^{\kappa\cdot r_{E_U}\cdot\alpha} = (\hat{e}(C_1, h^{r_{E_U}P_{(A_U,S)}(\gamma)}))\cdot \quad (6)$$
$$\hat{e}(A_2,C_2))^{\frac{1}{\prod_{a\in S\cup D_{n+t-1-s}\setminus A_U}\tau(a)}}$$

$$\hat{e}(g,h)^{\kappa\cdot\alpha} = \hat{e}(C_1, h^{\frac{r_{E_U}^{-1}}{\gamma}})\hat{e}(g,h)^{\kappa\cdot r_{E_U}\cdot\alpha} \quad (7)$$

$$\hat{e}(g,\sigma_3) = \hat{e}(g,h^{\alpha\cdot\mathcal{H}(M)}) = \hat{e}(g,h)^{\alpha\cdot\mathcal{H}(M)} \quad (8)$$

Finally, the user can retrieve the message:

$$M = \frac{C_3}{\hat{e}(g,h)^{\alpha\cdot(\mathcal{H}(M)+\kappa)}} = \frac{C_3}{K} \quad (9)$$

Then, he computes $\mathcal{H}(M)$, using the retrieved message M, in order to verify that the received σ_3 is equal to $h^{\alpha\cdot\mathcal{H}(M)}$.

6.2 Confidentiality

For the security game against the confidentiality property introduced in Section 4.2.1, the adversary $\mathcal{A}$ tries to distinguish between two signcrypted messages generated using the signcryption scheme. In other words, the adversary $\mathcal{A}$ tries to solve the *aMSE − CDH* problem using a challenging algorithm $\mathcal{B}$.

In a first phase, the adversary $\mathcal{A}$ sets a set of signcryption attributes S^*. Then, the algorithm $\mathcal{B}$ generates and publishes the public parameters. In the second phase, the adversary $\mathcal{A}$ tries to gain knowledge about the unsigncryption process by requesting the execution of the `KeyGen` and `UnsignCrypt` algorithms. Thanks to the hardness of the *aMSE − CDH* problem, the adversary $\mathcal{A}$ cannot recognize the secret values such as r_i, γ used to generate the users' secret keys. The **Challenge Phase** consists of choosing two messages by the adversary and sending them to the algorithm $\mathcal{B}$. Then, $\mathcal{B}$ randomly chooses a message, signcrypts it and returns it to the adversary. $\mathcal{A}$ tries to guess the original message being signcrypted. However, with respect to the hardness of the *aMSE − CDH* problem (Definition 5), the adversary is not able to guess the message even if he might after seeing the challenge ciphertext. Note that the adversary may not request decryption of the challenge message itself.

In the following detailed proof, we prove that our scheme is IND-CCA2 secure assuming that the *aMSE − CDH* problem is hard to solve with respect to Theorem 1.

Theorem 1. *Let ξ be an integer. For any adversary $\mathcal{A}$ against the IND-CCA2 security of our attribute-based signcryption scheme, for a universe of m attributes $\mathcal{U}$, and a challenge pair (t^*, S^*) with $s = |S^*|$, there exists a solver $\mathcal{B}$ of the $(\tilde{l},\tilde{m},\tilde{t})$-aMSE-CDH problem, for $\tilde{l} = m - s$, $\tilde{m} = m + t - s$ and $\tilde{t} = t + 1$, such that $Pr[Exp_{\mathcal{B}}^{aMSE-CDH}(1^{\xi})] \geq \frac{1}{2}Pr[Exp_{\mathcal{A}}^{conf}(1^{\xi})]$.*

Proof. We consider an algorithm $\mathcal{B}$ that uses the adversary $\mathcal{A}$ as a black-box and that solves the $(\tilde{l},\tilde{m},\tilde{t})$-aMSE-CDH problem.

Let $I(x_{2m+\vec{t-1-s}}, \kappa, \mu, \alpha, \gamma, \omega, T)$ be the input of the algorithm $\mathcal{B}$ where the components of the vector $x_{2m+\vec{t-1-s}} = (x_1, \cdots, x_{2m+t-1-s})$ are pairwise distinct elements of $(\mathbb{Z}/p\mathbb{Z})^*$ and let define the polynomials $f(X)$ and $g(X)$ as detailed in Definition 5. First, $\mathcal{B}$ chooses a universe of attributes $\mathcal{U} = \{a_1, \cdots, a_m\}$. Then, the adversary $\mathcal{A}$ defines a set $S^* \subset \mathcal{U}$ where $|S| = s$ and a threshold t^* such that $1 \leq t^* \leq s$. We assume that $S^* = \{a_{m-s+1}, \cdots, a_m\} \subset \mathcal{U}$. We denote by $A_{\mathcal{A}}$ the subset $A \cap S^*$, for any subset of attributes A.

Setup – the algorithm $\mathcal{B}$ sets $\tau(a_i) = x_i$ for $i = 1, \cdots, m$ as the encoding function of the attributes. Notice that the encoding of the first $m-s$ elements are the opposite of the roots of $f(X)$, and the encoding of the attributes in S^* are the opposite of some roots of $g(X)$ (Delerablée and Pointcheval, 2008). Then, $\mathcal{B}$ defines the values corresponding to the *dummy* attributes $\mathcal{D} = \{d_1, \cdots, d_{m-1}\}$ as $d_j = x_{m+j}$ if $j = 1, \cdots, m+t-1-s$. For $j = m+t-s, \cdots, m-1$, where the d_j values are selected at random in $(\mathbb{Z}/p\mathbb{Z})^*$ so that the elements of the subset $\{x_1, \cdots, x_{2m+t-1-s}, d_{m+t-s}, \cdots, d_{j-1}\}$ are pairwise distinct. The algorithm $\mathcal{B}$ defines $g = g_0{}^{f(\gamma)}$ which can be computed using f. To complete the **Setup** phase, the algorithm $\mathcal{B}$ sets $h = h_0$ and computes:

- $u = g^{\alpha \cdot \gamma} = g_0{}^{\alpha \cdot \gamma \cdot f(\gamma)}$ can be computed since $Xf(X)$ is a polynomial of degree $\tilde{l}+1$. In fact, the quantity $\alpha\gamma f(\gamma)$ is a linear combination of $\alpha\gamma, \cdots, \alpha\gamma^{\tilde{l}+1}$ where its coefficients are known to $\mathcal{B}$.
- $v = \hat{e}(g,h)^{\alpha} = \hat{e}(g_0{}^{\alpha \cdot f(\gamma)}, h_0)$, where g^{α} is computed based on Equation 5.

The algorithm $\mathcal{B}$ derives the values $\{h^{\alpha\gamma^i}\}_{i=0,\cdots,2m-1}$ from the input set I. Finally, the algorithm $\mathcal{B}$ returns the public parameters to the adversary $\mathcal{A}$.

Unsigncryption Query Phase 1 – during this phase, $\mathcal{B}$ and $\mathcal{A}$ proceed as follows:

- `KeyGen` – for each `KeyGen` query i, $\mathcal{A}$ requests the key extraction for a subset $A_{\mathcal{A}}$ from $\mathcal{B}$, where $|A_{\mathcal{A}}| < t_i$. To compute the queried secret keys, $\mathcal{B}$ chooses $r_i = (\omega y_i \gamma + 1) Q_i(\gamma)$, where y_i is randomly picked in $(\mathbb{Z}/p\mathbb{Z})^*$, and $\mathcal{B}$ defines the polynomial $Q_i(X)$ as $Q_i(\gamma) = 1$ when $|A_{\mathcal{A}}| = 0$, or $Q_i(X) = \lambda_i \cdot \prod_{a \in A_{\mathcal{A}}} (X + \tau(a))$ where $\lambda_i = (\prod_{a \in A_{\mathcal{A}}} \tau(a))^{-1}$.
  Then, the algorithm $\mathcal{B}$ can compute the components of the secret key tuple $sk_{\mathcal{A}} = (\{g^{\frac{r_i}{\gamma+\tau(a)}}\}_{a \in A_{\mathcal{A}}}, \{h^{r_i \gamma^k}\}_{k=0,\cdots,m-2}, h^{\frac{r_i - 1}{\gamma}})$ for a set of attributes $A_{\mathcal{A}} = \{a_{i_1}, \cdots, a_{i_n}\} \subset \mathcal{U}$, queried at session i, as follows:
  - for each $a_k \in A_{\mathcal{A}}$, the algorithm $\mathcal{B}$ computes $Q_{a_k}(\gamma) = \frac{Q_i(\gamma)}{(\gamma+\tau(a_k))} = \lambda_i \cdot \prod_{\tilde{a_k} \in A_{\mathcal{A}}, \tilde{a_k} \neq a_k} (\gamma + \tau(\tilde{a_k}))$. Then, $\mathcal{B}$ calculates the first secret key element $sk_{\mathcal{A}_1} = \{g^{\frac{r_i}{\gamma+\tau(a_k)}}\}_{a_k \in A_{\mathcal{A}}}$ defined as $sk_{\mathcal{A}_1} = \{g_0{}^{f(\gamma)\omega y_i Q_{a_k}(\gamma)} \cdot g_0{}^{f(\gamma) Q_{a_k}(\gamma)}\}_{a_k \in A_{\mathcal{A}}}$.
  - for each $a_k \in A \setminus A_{\mathcal{A}}$, the algorithm $\mathcal{B}$ defines $f_{a_k}(X) = \frac{f(X)}{(X+\tau(a_k))}$. Then, $\mathcal{B}$ computes the first secret key element such as $sk_{\mathcal{A}_1} = \{g^{\frac{r_i}{\gamma+\tau(a_k)}}\}_{a_k \in A_{\mathcal{A}}}$
  $= \{g_0{}^{f_{a_k}(\gamma)\omega y_i Q_i(\gamma)} \cdot g_0{}^{f_{a_k}(\gamma) Q_i(\gamma)}\}_{a_k \in A_{\mathcal{A}}}$.
  - $\mathcal{B}$ calculates the second item of the secret key $sk_{\mathcal{A}_2} = \{h^{r_i \gamma^j}\}_{j=0,\cdots,m-2}$ such as $h^{r_i \gamma^j} = h^{\omega y_i \gamma^{j+1} Q_i(\gamma)} \cdot h^{\gamma^j Q_i(\gamma)}$.
  - finally, $\mathcal{B}$ computes the third secret key element such as $sk_{\mathcal{A}_3} = h^{\frac{r_i - 1}{\gamma}} = h^{\omega y_i Q_i(\gamma)} \cdot h^{\frac{Q_i(\gamma)-1}{\gamma}}$.
- `UnsignCrypt` – the adversary $\mathcal{A}$ can adaptively requests the unsigncryption queries while considering the signcryption attribute set A_U. The algorithm $\mathcal{B}$ executes the `KeyGen` to generate the secret key $sk_{\mathcal{B}} = \texttt{KeyGen}(\texttt{PP}, msk, A_U)$ and finally, $\mathcal{B}$ answers by running the $\texttt{Unsigncrypt}(\texttt{PP}, sk_{\mathcal{B}}, A_U, (t_i, S^*), \Sigma_i)$ algorithm and forwards M_i to the adversary.

Challenge Phase – the adversary $\mathcal{A}$ chooses the two equal length plaintext messages $M_0{}^*$ and $M_1{}^*$ and sends them to $\mathcal{B}$, who flips a coin $b \in \{0,1\}$. Then, $\mathcal{B}$ sets $C_3^* = T \cdot M_b$. Afterwards, $\mathcal{B}$ chooses a random value for the encryption as $\kappa' = \kappa/\alpha$ and computes $\mathcal{H}(M_b)$. Then, $\mathcal{B}$ sets $C_2^* = h_0^{\kappa' \cdot g(\gamma)}$, $C_1^* = (g_0^{\kappa' \cdot \gamma \cdot f(\gamma)})^{-1}$ and $\sigma_3^* = h_0{}^{\mathcal{H}(M_b)}$.

Unsigncryption Query Phase 2 – the queries are executed as in the **Unsigncryption Query Phase 1**.

Guess – the adversary $\mathcal{A}$ outputs a bit b'. If $b' = b$, $\mathcal{B}$ answers 1 as the solution to the given instance of the aMSE-CDH problem with respect to Definition 5, meaning that $T = \hat{e}(g_0, h_0)^{(\kappa + \mathcal{H}(M_b))} \cdot f(\gamma)$. Otherwise, $\mathcal{B}$ answers 0, meaning that T is a random element. The advantage of the algorithm $\mathcal{B}$ is as follows:

$$
\begin{aligned}
Pr[Exp_{\mathcal{B}}^{aMSE-CDH}(1^{\xi})] &= |Pr[\mathcal{B}(\mathtt{I}(x_{2m+\vec{t-1-s}}, \kappa, \mu, \alpha, \gamma, \omega, T) = 1 | real] \\
&\quad - Pr[\mathcal{B}(\mathtt{I}(x_{2m+\vec{t-1-s}}, \kappa, \mu, \alpha, \gamma, \omega, T) = 1 | random]| \\
&= |Pr[b = b' | real] - Pr[b = b' | random]|
\end{aligned}
$$

When the event real occurs, then $\mathcal{A}$ is playing a real attack and therefore $|Pr[b = b'|real] - \frac{1}{2}]| = \frac{1}{2}Exp_{\mathcal{A}}{}^{conf}(1^{\xi})$. During the random event, the view of $\mathcal{A}$ is completely independent of the bit b; in this case, the probability $Pr[b = b']$ is equal to $1/2$. Then, we obtain:

$$Pr[Exp_{\mathcal{B}}{}^{aMSE-CDH}(1^{\xi})] \geq \frac{1}{2}Pr[Exp_{\mathcal{A}}{}^{conf}(1^{\xi})]$$

Thus, for all PPT adversaries, the advantage of the adversary $\mathcal{A}$ is negligeable. Then, our proposed $t-ABSC$ scheme satisfies the confidentiality property. □

6.3 Unforgeability

In the security game defined in Section 4.2.2, the adversary $\mathcal{A}$ tries to forge a valid signature based on the use of an algorithm $\mathcal{B}$ who is able to solve the $aMSE-CDH$ problem.

Thus, the adversary $\mathcal{A}$ chooses a set of signcryption attributes S^* and a threshold t, in the **Setup** phase. Afterwards, the adversary $\mathcal{A}$ requests the execution of the `KeyGen` queries while changing the threshold t_i. Moreover, the adversary $\mathcal{A}$ asks for the signcryption of a message M under different signcryption attribute sets and threshold values. By executing this **Signcrypt Query Phase**, the adversary tries to get information about the secret values included in the `KeyGen` and `SignCrypt` algorithms.

Afterwards, the adversary $\mathcal{A}$ tries to generate a valid signcryption with respect to the challenge policy $(t^*, S*)$. Thus, the adversary $\mathcal{A}$ has to solve the $aMSE-CDH$ problem with respect to the Definition 5 for proving that he has the required attributes to satisfy the (t^*, S^*) access policy. Moreover, thanks to the random values added to the generated signature, the adversary $\mathcal{A}$ has to solve the CDH problem with respect to the Definition 4.

However, thanks to the hardness of to the $aMSE-CDH$ problem and the CDH problem, the adversary cannot learn information about the secret values used in the key generation and the signcryption process.

In the following, we prove that our construction is unforgeable against chosen message attack with respect to Theorem 2.

Theorem 2. *The scheme is adaptive-message unforgeable under chosen message attacks for a universe $\mathcal{U}$ of m attributes, and a challenge pair (t^*, S^*) with $s = |S^*|$, there exists a solver $\mathcal{B}$ of the $(\tilde{l}, \tilde{m}, \tilde{t})$-aMSE-CDH problem, for $\tilde{l} = m - s$, $\tilde{m} = m + t - s$ and $\tilde{t} = t + 1$, such that $Pr[Exp_{\mathcal{B}}{}^{aMSE-CDH}(1^{\xi})] \geq Pr[Exp_{\mathcal{A}}{}^{unf}(1^{\xi}) = 1]$ assuming that the $(\tilde{l}, \tilde{m}, \tilde{t})$-aMSE-CDH problem holds.*

Proof. We consider an algorithm $\mathcal{B}$ that uses the adversary $\mathcal{A}$ as a black-box and that solves the $(\tilde{l}, \tilde{m}, \tilde{t})$-aMSE-CDH problem.

Let $I(x_{2m+\vec{t}-1-s}, \kappa, \mu, \alpha, \gamma, \omega, T)$ be the input of the algorithm $\mathcal{B}$ where the components of the vector $x_{2m+\vec{t}-1-s} = (x_1, \cdots, x_{2m+t-1-s})$ are pairwise distinct elements of $(\mathbb{Z}/p\mathbb{Z})^*$ and define the polynomials $f(X)$ and $g(X)$ as detailed in Definition 5. First, $\mathcal{B}$ chooses a universe of attributes $\mathcal{U} = \{a_1, \cdots, a_m\}$. Then, the adversary $\mathcal{A}$ chooses a set $S^* \subset \mathcal{U}$ where $|S| = s$ and a threshold t^* such that $1 \leq t^* \leq s$. We assume that $S^* = \{a_{m-s+1}, \cdots, a_m\} \subset \mathcal{U}$. We denote by $A_{\mathcal{A}}$ the subset $A \cap S^*$, for any subset of attributes A.

We consider the same setting for the **Setup** phase as detailed in the Exp^{conf} security game. For the **Signcrypt Query Phase**, the adversary $\mathcal{A}$ requests, as many times as he wants, the key extraction oracle `KeyGen` and the signcryption oracle `SignCrypt`. For each `KeyGen` query i, $\mathcal{A}$ requests the key extraction for a subset $A_{\mathcal{A}}$ from $\mathcal{B}$, as detailed in the Exp^{conf} security game.

Afterwards, the adversary $\mathcal{A}$ can adaptively request a `SignCrypt` query i while considering the signcryption attribute set A_U and a message M_i. The algorithm $\mathcal{B}$ executes the `KeyGen` to generate the secret key $sk_{\mathcal{B}} = \texttt{KeyGen}(\texttt{PP}, msk, A_U)$ and finally, $\mathcal{B}$ answers by running the $\texttt{signCrypt}(\texttt{PP}, sk_{\mathcal{B}}, A_U, (t_i, S^*), M_i)$ algorithm and forwards Σ_i to the adversary.

The algorithm $\mathcal{B}$ sets $Q_i(\gamma)/\lambda_i = \prod_{a \in A_U}(\gamma + \tau(a))$ and computes:

- $\sigma_{1,i} = g_0{}^{(\omega y_i \gamma + 1) f(\gamma) \lambda_i} \cdot g_0{}^{f(\gamma)\frac{\lambda_i \mathcal{H}(M_i)}{Q_i(\gamma)}}$.
- $\sigma_{2,i} = h^{\omega y_i Q_i(\gamma)} \cdot h^{\frac{Q_i(\gamma)-1}{\gamma}} \cdot h^{\frac{(\omega y_i \gamma + 1) Q_i(\gamma) \cdot P_{(A_U, S^*)}(\gamma)}{\prod_{a \in S^* \cup D_{n+t-1-s} \setminus A_U} \tau(a)}} \cdot h^{\frac{\mathcal{H}(M_i) P_{(A_U, S^*)}(\gamma)}{\prod_{a \in S^* \cup D_{n+t-1-s} \setminus A_U} \tau(a)}}$.
- $\sigma_{3,i} = h^{\alpha \cdot \mathcal{H}(M_i)}$.
- finally, $\mathcal{B}$ computes $C_{1,i}$, $C_{2,i}$ and $C_{3,i}$, as detailed in the Exp^{conf} security game.

Forgery Phase – during this phase, $\mathcal{A}$ outputs a signcrypted message Σ^* with respect to the threshold challenge policy (t^*, S^*). For this purpose, $\mathcal{A}$ can compute C_1^*, C_2^*, C_3^* and σ_3^*, based on public parameters and the selected random κ^* and $\mathcal{H}(M_i)$. Finally, $\mathcal{A}$ has to compute valid σ_1^* and σ_2^* to win the unforgeability security game. Hence, $\mathcal{A}$ has to solve the aMSE-CDH problem with respect to Definition 5 for proving that he has the required attributes to satisfy the (t^*, S^*) access policy. Similarly, the adversary $\mathcal{A}$ has to solve the CDH problem with respect to Definition 4. Thus, $\mathcal{A}$ has the same advantage as the Exp^{conf} security game.

Thus, for all PPT adversaries, the advantage of the adversary $\mathcal{A}$ is negligible. Then, our proposed $t-ABSC$ is unforgeable against chosen-message attack. □

6.4 Privacy

The privacy game begins when the challenger $\mathcal{C}$ executes the `Setup` algorithm to generate the public parameters. Then, the challenger $\mathcal{C}$ sends the public parameters to the adversary $\mathcal{A}$. Afterwards, the adversary $\mathcal{A}$ chooses two attribute sets A_{S_1} and A_{S_1} satisfying the threshold access policy (t,S) and sends them to the challenger $\mathcal{C}$. The challenger generates the private keys related to the sets of attribute A_{S_1} and A_{S_2} as follows:

$$sk_{E_{C_1}} = (\{g^{\frac{r_{E_C}}{\gamma+\tau(a)}}\}_{a\in A_{S_1}}, \{h^{r_{E_C}\gamma^i}\}_{i=0,\cdots,m-2}, h^{\frac{r_{E_C}-1}{\gamma}})$$

$$sk_{E_{C_2}} = (\{g^{\frac{r_{E_C}}{\gamma+\tau(a)}}\}_{a\in A_{S_2}}, \{h^{r_{E_C}\gamma^i}\}_{i=0,\cdots,m-2}, h^{\frac{r_{E_C}-1}{\gamma}})$$

Moreover, the adversary $\mathcal{A}$ outputs a challenge message M and asks the challenger $\mathcal{C}$ to generate the signcryption of M using one of the private keys $sk_{E_{C_1}}$ or $sk_{E_{C_2}}$. Thus, the challenger chooses a random bit $b\in\{0,1\}$, and computes a signcryption Σ_b by running the algorithm $\texttt{SignCrypt}(\texttt{PP}, sk_{E_{C_b}}, sk_{E_{C_b}}, (t,S), M)$. Since $|A_{S_b}\cap S| = t$, the challenger can generate a valid signcrytpion on the message M. Thus, to prove that the scheme is private, we just have to prove that the signcrypted message created using $sk_{E_{C_1}}$ or $sk_{E_{C_2}}$ are identical.

Using the private key related to $sk_{E_{C_b}}$, the generated signcryption $\Sigma_b = (C_1, C_2, C_3, \sigma_1, \sigma_2, \sigma_3)$ is detailed as follows:

$$\begin{cases} C_1 = (g^{\alpha\cdot\gamma})^{-\kappa} \\ C_2 = h^{\kappa\alpha\cdot\prod_{a\in S}(\gamma+\tau(a))\prod_{d\in D_{n+t-1-s}}(\gamma+d)} \\ C_3 = \hat{e}(g,h)^{\alpha\cdot\kappa}\cdot\hat{e}(g,h)^{\alpha\cdot\mathcal{H}(M)}\cdot M = K\cdot M \end{cases}$$

and

$$\begin{cases} \sigma_1 = g^{\frac{r_{E_C}}{\prod_{a\in A_{S_b}}(\gamma+\tau(a))}}\cdot g^{\frac{\mathcal{H}(M)}{\prod_{a\in A_{S_b}}(\gamma+\tau(a))}} \\ \sigma_2 = h^{\frac{r_{E_C}-1}{\gamma}} h^{\frac{r_{E_C}\cdot P_{(A_{S_b},S)}(\gamma)}{\prod_{a\in S\cup D_{n+t-1-s}\setminus A_{S_b}}\tau(a)}}\cdot h^{\frac{\mathcal{H}(M)P_{(A_{S_b},S)}(\gamma)}{\prod_{a\in S\cup D_{n+t-1-s}\setminus A_{S_b}}\tau(a)}} \\ \sigma_3 = h^{\alpha\cdot\mathcal{H}(M)} \end{cases}$$

While receiving the signcrypted message, the adversary $\mathcal{A}$ verifies the signature, by computing:

$$\begin{aligned} \text{Ⓢ} &= \hat{e}(u^{-1},\sigma_2)\cdot\hat{e}(\sigma_1^{\frac{1}{B_1}}, h^{\alpha\cdot\prod_{a\in S\cup D_{n+t-1-s}}(\gamma+\tau(a))})\cdot \\ &\quad \hat{e}(g^{\alpha},h)^{\mathcal{H}(M)\cdot(1-U_1-\frac{1}{U_1})} = \hat{e}(g^{\alpha},h) \end{aligned}$$

Since $|A_{S_b}\cap S| = |A_{S_1}\cap S| = |A_{S_2}\cap S| = t$, we can prove that the signature generated using the set of attributes A_{S_1} (in other words, using the secret key $sk_{E_{C_1}}$) is similar to the signature generated using the set of attributes A_{S_2} (using the secret key $sk_{E_{C_2}}$). Hence, the challenger $\mathcal{A}$ cannot know the set of attributes used to generate the signature with respect to the hardness of the *CDH* problem (Definition 4).
Thus, our proposed attribute based signcryption scheme is computationally private.

7 PERFORMANCES ANALYSIS

In this section, we present the computation and the storage complexities of our proposed signcryption scheme. In our analysis, we are interested in the computations performed to execute the `SignCrypt` and the `UnsignCrypt` algorithms as well as the size of the generated signcrypted message (signcryption size) and the size of the secret keys as introduced in Table 2. In the following, we denote by E_1 the exponentiation cost in $\mathbb{G}_1$, E_T the exponentiation cost in $\mathbb{G}_T$ and τ_P the computation cost of a pairing function $\hat{e}$. Table 2 details the performance comparison of ABSC schemes.

7.1 Storage Complexities

Our signcryption scheme consists of using a common setup procedure to generate the public parameters and the secret keys used to sign and encrypt the message. Our proposal relies on the use of a single access structure for both signature and encryption phases on the signcryption procedure unlike other signcryption schemes. Thus, the key size is equal to $n+m$ where m and $n=|A|$ are the cardinals of the attributes' universe $\mathcal{U}$ and the set of the user's attributes, respectively.

The proposal of Emura et al. (Emura et al., 2012) consists of using two access structures. As such, it uses two different secret keys' sets related to the signature and the encryption procedures. The size of the first part of the keys is equal to $2n_s$ while the size of the second part is equal to $2n_e+1$ where n_s and n_e are the sizes of the set of the signing attributes and the size of the set of the encryption attributes of the user, respectively. Similarly, Liu et al. (Liu et al., 2015) use two access structures. So that, the sizes of the secret keys are equal to n_s+2 and n_e+2 for signing and encrypting, respectively. The proposal of Gagné et al. (Gagne et al., 2010) also uses different access structures for signature and encryption and consists of using secret keys which their sizes are equal to $2n_s$ and $3n_e$. Finally, Rao et al.(Rao and Dutta, 2014)

Table 2: Computation and Storage Costs of Attribute Based Signcryption Schemes.

Scheme	*Key size*	*Signcryption size*	*Signcrypt time*	*Unsigncrypt time*
(Emura et al., 2012)	$2n_s, 2n_e+1$	$2+5l_e+2n_s$	$E_1(10+4n_s)+E_T+\tau_p$	$\tau_p(3n_s+2+2n_e+2m)$
(Liu et al., 2015)	n_s+2, n_e+2	$4+2l_s+l_e$	$\tau_p+E_T+E_1(3+l_e+3l_s)$	$\tau_p(3n_s+5)+2n_sE_T+2n_sE_1$
(Gagne et al., 2010)	$3n_e, 2n_s$	$O(M)+4+n_e+n_s$	$E_1(2O(M)+mn_e+4n_e+3+n_s)$	$2E_1+\tau_p(2t+2+n_s)+tE_T+$
(Rao and Dutta, 2014)	$m+l_s, m+l_e$	8	$E_1(10+l_e+3l_s+n_s)$	$6\tau_p+E_1(n_e+3l_e)$
(Rao and Dutta, 2016)	$m+l_s, m+l_e$	8	$E_1(10+l_e+3l_s+n_s)$	$6\tau_p+E_1(n_e+3l_e)$
Our scheme	$n+m$	8	$E_1(6+t)+2E_T$	$E_1(5+t)+9\tau_p+4E_T$

proposal consists of a key-policy attribute based signcryption scheme which relies on the use of different access structures for signing and encrypting. Thus, the sizes of the secret keys are equal to $m+l_s$ and $m+l_e$ where l_s and l_e denote the sizes of the signing policy and the encryption policy, respectively.

Emura et al. (Emura et al., 2012) propose a CP-ABSC scheme where the size of the signcrypted message is equal to $2+5l_e+2n_s$. Thus, it increases linearly with both the number of attributes introduced in the encryption access policy l_e and the number of the signing attributes of the user n_s. Moreover, the proposal of Liu et al. (Liu et al., 2015) introduces a signcryption scheme where the size of the signcrypted message depends on the number of attributes used on both encryption and signature policies (l_s, l_e). For instance, this scheme outputs a signrypted message whose size is $4+2l_s+l_e$. The threshold attribute based signcryption scheme proposed by Gagné et al. outputs a signcrypted message's size depending on both the number of the user's attributes related to signing and decrypting, n_s and n_e, respectively. The size of the signrypted message is equal to $O(M)+4+n_e+n_s$. The proposal of Rao et al.(Rao and Dutta, 2014) is the first constant size key-policy attribute based signcryption scheme. Indeed, the size of the signcrypted message is equal to 8 and it is independent from the number of attributes involved in the access policy. Similarly, our t-ABSC proposal is the first threshold ciphertext-policy attribute based signcryption scheme that outputs a constant signcrypted message's size equal to 8, where the size of the signcrypted message does not depend on the size of the access policy.

Consequently, our proposal is the only threshold ciphertext-policy attribute based signcryption that uses a common setup based on a single access policy for encryption and signing. Although, our scheme presents less expressiveness compared to other schemes, it presents interesting performances especially related to the key size and the constant size of the encrypted message.

7.2 Computation Complexities

In (Emura et al., 2012), Emura et al. proposal consists of computing $(10+4n_s)$ exponentiations in $\mathbb{G}_1$, one exponentiation in $\mathbb{G}_T$ and one pairing function while performing the signcryption algorithm. Moreover, the unsigncryption algorithm computes $(3n_s+2+2n_e+2m)$ pairing function. Recently, Liu et al. (Liu et al., 2015) executes $(3+l_e+3l_s)$ exponentiations in $\mathbb{G}_1$, one exponentiation in $\mathbb{G}_T$ and one pairing function during the signcryption procedure. Both the proposed schemes by Rao et al. (Rao and Dutta, 2014; Rao and Dutta, 2016) introduce an attribute based signcryption scheme where the signcryption and unsigncryption processing overheads are equal to $E_1(10+l_e+3l_s+n_s)$ and $6\tau_p+E_1(n_e+3l_e)$, respectively. The threshold attribute based signcryption scheme proposed by Gagné et al. introduces a signcryption mechanism where the signcrypting consists of performing $(2O(M)+mn_e+4n_e+3+n_s)$ exponentiation in $\mathbb{G}_1$, where $O(M)$ is the size of the plaitext message. Moreover, the unsigncryption algorithm consists of $(2t+2+n_s)$ pairing function, 2 exponentiations in $\mathbb{G}_1$ and t exponentiations in $\mathbb{G}_T$. Finally, our proposed t-ABSC scheme consists of performing $(6+t)$ exponentiations in $\mathbb{G}_1$ and 2 exponentiations in $\mathbb{G}_T$ in the signcryption phase. In addition, it computes $(5+t)$ exponentiations in $\mathbb{G}_1$, 2 exponentiations in $\mathbb{G}_T$ and 9 pairing function to perform the unsigncryption algorithm. It is noteworthy that the computation overheads of the signcryption and unsigncryption procedures introduced in our threshold attribute based signcryption scheme only depends on the threshold value t.

Our proposed scheme presents interesting computation costs compared to other signcryption schemes thanks to the use of common setup phase to generate public parameters and the use of the same access policy for both signature and encryption phases. For instance, our proposal consists of performing a fixed number of pairing functions whose computation overhead are heavy compared to the exponentiation operations.

8 CONCLUSIONS

The widespread use of cloud data storage and the attractive properties offered by the attribute based mechanisms lead us to combine them in order to ensure data security and user's privacy. In this paper, we have proposed a novel threshold attribute based signcryption (t-ABSC) scheme ensuring flexible access control, data confidentiality, user's privacy as well as data origin authentication with anonymity. Moreover, the size of signcrypted messages does not depend on the number of attributes involved in the access policy, which makes our t-ABSC scheme more suitable for bandwidth-limited applications. Finally, our construction is proven to ensure strong security properties in the random oracle model, namely confidentiality against the chosen ciphertext attacks, unforgeability against chosen message attacks and privacy preservation, based on formal security games.

REFERENCES

Attrapadung, N., Herranz, J., Laguillaumie, F., Libert, B., De Panafieu, E., and Ràfols, C. (2012). Attribute-based encryption schemes with constant-size ciphertexts. *Theoretical Computer Science*, 422:15–38.

Bacis, E., di Vimercati, S. D. C., Foresti, S., Paraboschi, S., Rosa, M., and Samarati, P. (2016). Access control management for secure cloud storage. In *Proc. of the 12th International Conference on Security and Privacy in Communication Networks (SecureComm 2016), Guangzhou, China*.

Belguith, S., Jemai, A., and Attia, R. (2015). Enhancing data security in cloud computing using a lightweight cryptographic algorithm. In *ICAS 2015 : The Eleventh International Conference on Autonomic and Autonomous Systems*, pages 98–103. IARIA.

Belguith, S., Kaaniche, N., Jemai, A., Laurent, M., and Attia, R. (2016). Pabac: a privacy preserving attribute based framework for fine grained access control in clouds. In *13th IEEE International Conference on Security and Cryptography(Secrypt)*.

Bethencourt, J., Sahai, A., and Waters, B. (2007). Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy.*

Carlin, A., Hammoudeh, M., and Aldabbas, O. (2015). Intrusion detection and countermeasure of virtual cloud systems-state of the art and current challenges. *International Journal of Advanced Computer Science and Applications*, 6(6):1–15.

Delerablée, C., Paillier, P., and Pointcheval, D. (2007). Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In *International Conference on Pairing-Based Cryptography*, pages 39–59. Springer.

Delerablée, C. and Pointcheval, D. (2008). Dynamic threshold public-key encryption. In *Annual International Cryptology Conference*. Springer.

Dent, A. W., Fischlin, M., Manulis, M., Stam, M., and Schröder, D. (2010). Confidential signatures and deterministic signcryption. In *Public Key Cryptography–PKC*.

di Vimercati, S. D. C., Foresti, S., and Samarati, P. (2014). Selective and fine-grained access to data in the cloud. In *Secure Cloud Computing*, pages 123–148. Springer.

di Vimercati, S. D. C., Livraga, G., Piuri, V., Samarati, P., and Soares, G. A. (2016). Supporting application requirements in cloud-based iot information processing. In *Proc. of the International Conference on Internet of Things and Big Data (IoTBD 2016), Rome, Italy*.

Emura, K., Miyaji, A., and Rahman, M. S. (2012). Dynamic attribute-based signcryption without random oracles. *International Journal of Applied Cryptography*, 2(3).

Gagne, M., Narayan, S., and Safavi-Naini, R. (2010). Threshold attribute-based signcryption. In *Security and Cryptography for Networks*. Springer.

Haber, S. and Pinkas, B. (2001). Securely combining public-key cryptosystems. In *The 8th ACM conference on Computer and Communications Security*.

Herranz, J., Laguillaumie, F., and Ràfols, C. (2010). Constant size ciphertexts in threshold attribute-based encryption. In *International Workshop on Public Key Cryptography*, pages 19–34. Springer.

Kaaniche, N., Laurent, M., and El Barbori, M. (2014). Cloudasec: A novel publickey based framework to handle data sharing security in clouds. In *11th IEEE International Conference on Security and Cryptography(Secrypt)*.

Libert, B. and Quisquater, J.-J. (2004). Efficient signcryption with key privacy from gap diffie-hellman groups. In *Public Key Cryptography–PKC*.

Liu, J., Huang, X., and Liu, J. K. (2015). Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption. *Future Generation Computer Systems*, 52.

Maji, H. K., Prabhakaran, M., and Rosulek, M. (2011). Attribute-based signatures. In *Topics in Cryptology–CT-RSA 2011*. Springer.

Rao, Y. S. and Dutta, R. (2014). Expressive bandwidth-efficient attribute based signature and signcryption in standard model. In *Australasian Conference on Information Security and Privacy*. Springer.

Rao, Y. S. and Dutta, R. (2016). Efficient attribute-based signature and signcryption realizing expressive access structures. *International Journal of Information Security*, 15(1):81–109.

Vasco, M. I. G., Hess, F., and Steinwandt, R. (2008). Combined (identity-based) public key schemes. *IACR Cryptology ePrint Archive*, 2008.

Waters, B. (2011). Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography–PKC*.

Zheng, Y. (1997). Digital signcryption or how to achieve cost (signature & encryption) ¡ cost (signature)+ cost (encryption). In *Advances in Cryptology, Crypto97*. Springer.