
ROMANIAN ACADEMY

School of Advanced Studies of the Romanian Academy
"Simion Stoilow" Institute of Mathematics

# PhD Thesis Summary

## Attribute-based Encryption for Boolean Circuits

Supervisor
CS1, Dr. Habil. Răzvan Diaconescu

PhD Student
Alexandru Ioniţă

2025

# Abstract

Attribute-Based Encryption (ABE) is a modern cryptographic framework that addresses the problem of fine-grained access control granting in secure data sharing. The attribute-based access policies enforced by these systems are suitable for scenarios involving complex access requirements, such as cloud computing and distributed systems.

This thesis explores the expressiveness of Attribute-Based Encryption (ABE) schemes from bilinear maps, with a focus on developing new constructions, with improvements in efficiency. Despite significant advancements, there is still consistent progress to be made: for example, it is unknown if there could be constructed efficient ABE schemes supporting arbitrary Boolean circuits.

The contributions of this thesis include on one side new ABE schemes using bilinear maps for multiple access structure: We provide two construction for two different weighted access structures, and a construction for Compartmented trees. We show that the latter is providing more expressiveness compared to existing schemes, while maintaining efficiency. All our constructions are accompanied by theoretical analysis and security proofs. Where it was necessary, we also provided practical tests to support our claims. Moreover, we have provided the first method of rewriting Boolean circuits into an equivalent form, such that the existing secret sharing schemes will produce fewer shares.

This work advances the state-of-the-art in ABE by addressing both theoretical and practical aspects of expressiveness. Our security proofs and implementations offer valuable insights and tools for future research and applications in cryptographic access control.

# Contents

# Chapter 1

# Introducere

With the constant increase in demand of Cloud hosting, the requirements for securely sharing data on these systems also increased. On such use-cases, modern fine-grained access granting encryption schemes provide the versatility much needed. Attribute-based encryption is one of the most used techniques to grant access to multiple parties, based on expressive access structures.

Sahai and Waters [22] proposed the first ABE scheme as a refinement of Identity-based Encryption. It was further refined into two flavors: *key-policy* and *ciphertext-policy*, based on how the access granting mechanism works: by linking the access policy to a document (ciphertext) or to a person (through the decryption key).

A central research question is determining which access structures can support efficient ABE schemes. [7] conjectures that there is no construction for Boolean circuits from bilinear maps, since the pairing operation is used in the first step of the decryption. However, until now, efficient ABE constructions were known only for Boolean formulae (access trees).

The primary scope of this work is to search for new schemes of ABE from bilinear maps. Also we try to find what are the most expressive access structure that can be used in these systems, resulting in at most a constant number of shares per attribute.

## 1.1 Our Contribution

First, we have proposed two ABE constructions for two different weighted access structures. The first construction aims to improve efficiency of traditional weighted ABE

schemes, and is backed by practical efficiency tests and theoretical analysis. The second construction provides is the first approach in ABE context for an access structure which we refer to as *fully weighted* access structure. Another important part of our contribution is the develpment of an efficient ABE scheme for CAS, which was then used to create a scheme for CAS-circuits. While exploring lower bounds on LSSS and MSP, we have proved that ABE schemes for Boolean circuits from linear secre sharing will require an exponential expansion on key or ciphertext size. Also, we have proved that *fully weighted* access structures cannot have ideal linear secret sharing schemes. We have also used heuristic algorithms in order to improve the efficiency of existing ABE schemes for Boolean circuits.

# Chapter 2

# Background

ABE is a complex cryptographic mechanism which often is based on in a combination of mathematical theories and cryptographic principles that enable attribute-based access policies. This chapter will present a detail overview over the background needed to fully understand how ABE is constructed, and the contribution that we have brought to the literature in this thesis. We will present basic secret sharing notions, since most ABE schemes rely on secret sharing over some descriptive access structure for fine-grained access granting. Moreover, we will describe general KP-ABE and CP-ABE structure followed up by security models and mathematical primitives used to construct such schemes.

## 2.1 Secret sharing

**Access Structure** The first step in order to ensure secure and efficient data management is to have a set of formal access control requirements over the data. Informally, an access structure represents these sets of authorized participants.

A *monotone* access structure is an access structure where if some set of participants is authorized, then every superset of that set is also authorized. We continue by explaining some important access structures used in ABE. A *Boolean circuit* is a directed acyclic graph over a set of input wires, concluding to a single output wire, with internal nodes representing logical gates of type $AND$, $OR$, or $NOT$. These gates may have fan-out greater than 1. A *monotone* Boolean circuit is a circuit without negation gates. A *Boolean formulae* or a Boolean access tree is a Boolean circuit where each node is limited to a fan-out of 1.

**Monotone Span Program.** Monotone span programs (in short, MSP) are another mathematical representation used to define access structures in secret sharing. They enable the construction of access policies by associating vectors to each party, and defining a span condition for authorized sets. A *monotone* span program is a span program which only contains positive literals ($\{x_1, x_2, ..., x_n\}$ but not $\bar{x_1}\bar{x_2}, ..., \bar{x_n}$). We say that a span program $\hat{M}$ computes a Boolean function $f$ if $\hat{M}$ accepts $\delta \Leftrightarrow f(\delta) = 1$.

**Secret sharing schemes** Based on an access structure, we can construct mathematical mechanisms to share a secret upon a set of participants, such that only a set of authorized participants is able to reconstruct it. In order to fully ensure the security of the secret sharing scheme, the scheme must ensure that no unauthorized set of participants is able to reconstruct the secret.

A *linear* secret sharing scheme is a secret sharing scheme for which the reconstruction of the secret is done as a linear combination of its parts. We say that a secret sharing scheme is *ideal* if the total parts received by the participants are the same size (number of bits) of the secret that was shared.

## 2.2 Attribute-based Encryption

Any KP-ABE or CP-ABE should follow the general structure of such a scheme, composed out of four algorithms (setup, encryption, key-generation and decryption). We will present in the next part the definition of KP-ABE and CP-ABE models.

**KP-ABE Model** A Key-Policy Attribute-based encryption scheme, as first described in [10], consists of four algorithms:

**setup($\lambda$)** A randomized algorithm that takes as input the implicit security parameter $\lambda$ and returns the public and secret keys ($MPK$ and $MSK$).

**encrypt**$(m, A, MPK)$ A probabilistic algorithm that encrypts a message $m$ under a set of attributes $A$ with the public key $MPK$, and outputs the ciphertext $E$.

**keygen**$(\mathcal{C}, MPK, MSK)$ This algorithm receives an access structure $\mathcal{C}$, public and master keys $MPK$ and $MSK$, and outputs corresponding decryption keys $DK$.

**decrypt**$(E, DK, MPK)$ Given the ciphertext $E$ and the decryption keys $DK$, the algorithm decrypts the ciphertext and outputs the original message.

In ABE schemes, security models play a crucial role in evaluating the robustness of the proposed system. The Selective-Set Model [10] is a widely used security model for Key-Policy Attribute-Based Encryption (KP-ABE).

## 2.3 Bilinear maps

Bilinear maps are a modern mathematical tool with a pivotal role in cryptography, especially in constructing advanced cryptographic primitives. A formal definition of bilinear maps, as it was given in [10] is:

**Bilinear maps** Given $G_1$ and $G_T$ two multiplicative cyclic groups of prime order $p$, a map $e : G_1 \times G_1 \to G_T$ is called *bilinear* if it satisfies:

- $e(x^a, y^b) = e(x, y)^{ab}$, for any $x, y \in G_1$ and $a, b \in \mathbb{Z}_p$;
- $g_T = e(g, g)$ is a generator of $G_T$, for any generator $g$ of $G_1$.

$G_1$ is called a *bilinear group* if the operation in $G_1$ and $e$ are both efficiently computable.

The Bilinear Decisional Diffie-Hellman Assumption (BDDH) is a hard problem in the bilinear map setting. Informally, it states that, having four random values $g^a, g^b, g^c$ and $g^r$ it is hard to distinguish between $e(g, g)^{abc}$ and $e(g, g)^r$. Most cryptographic schemes relying on bilinear maps use security proofs based on the BDDH assumption.

# Chapter 3

# Attribute-based Encryption Overview

After the first KP-ABE scheme [10] and the first CP-ABE scheme [4] many other schemes were proposed based on these two. Many extensions and functionalities were added to ABE to enhance its practicability. We will further iterate through the most important ABE directions and analyze the existing schemes.

## 3.1 Mathematical assumptions

One of the ways of categorizing the ABE constructions is based on the mathematical primitives they are using. Some of the most popular primitives are: bilinear maps, multi-linear maps and lattices. We will briefly discuss here some of the particularities for each one of them, alongside with important ABE schemes, focused on efficiency and expressiveness. Also, we will discuss various limitations for each of them, with possible solutions or open problems.

The first ABE scheme ever proposed [10] was constructed using *bilinear maps*, modeling a KP-ABE scheme with a Boolean tree access structure. Their model was efficient, robust, and proven to be secure unde the BDDH assumption. While [7] conjectured that it is impossible to construct ABE for circuits from bilinear maps, they have also proposed the first ABE for circuits from multilinear maps. The scheme proposed is proven to be selectively secure under the natural generalization of the BDDH Assumption - the *multilinear* DDH. [6] proposed a more refined system for ABE for circuits. The approach is similar, by using a *chained* multi-linear maps.

[1, 8] are other examples of ABE for circuits from multi-linear maps.

| Scheme | Primitive | Assumption | Efficiency | Access structure |
|---|---|---|---|---|
| [6] | multi-linear | MDDH | linear | BC |
| [7] | multi-linear | MDDH | linear | BC |
| [24] | bilinear | BDDH | exp. | BC |
| [12] | bilinear | BDDH | exp. | BC |
| Ours [25] | bilinear | BDDH | linear | CAS |
| Ours [13] | bilinear | BDDH | linear | CAS-circuit |

TABLE 3.1: Key-policy ABE schemes for Boolean circuits

While multi-linear map cryptography has only a theoretical importance at the moment due to the lack of secure constructions, lattice-based cryptography is another alternative to pairing-based cryptography.

## 3.2 Policy expressiveness in ABE

This work explores the policy expressiveness of ABE schemes, particularly focusing on Boolean circuits access structures. While the original access tree proposed by Goyal et al. [10], only provides limited expressiveness, an efficient implementation of Boolean circuits in ABE from bilinear maps remains an open challenge.

Garg et al. [7] proposed the first ABE for circuits, by relying on multi-linear maps. Ţiplea and Drăgan [6] refined this by introducing a leveled multi-linear map system, organizing circuit nodes into levels where multi-linear map operations occur during secret reconstruction.

For bilinear maps, Tiplea and Drăgan [24] extended the Goyal et al. scheme to support monotone Boolean circuits, by introducing a new gat - the FO gate - in order to mitigate the *backtracking attack*[7, 24].

In Table 3.1 we have a short overview of the *key-policy* ABE schemes for Boolean circuits (BC) based on bilinear or multi-linear maps.

**Range Attributes and Weighted ABE** Even starting with the first CP-ABE scheme[4] there was also proposed a method of realizing comparisons on attributes using a logarithmic-sized comparison tree at the bottom of the access tree. Shi et al. back in 2007 [23] developed one of the first ABE systems with support for range attributes. He uses a Segment Tree in order to share the keys, resulting in *logarithmic* number of encryptions, and also *logarithmic* decryption key size.

One of the first schemes *marketed* as *weighted* was [18]. They proposed a KP-ABE scheme where they modeled comparisons using chained components. However, their scheme was inefficient, the chain lengths being linear in the weight of the attribute.

In 2016, Wang et al. [26] proposed a CP-ABE scheme with two main features: support for weighted access structures and improved key issuing protocol, which results in resolving the key escrow problem. In the next years, more efficient ABE schemes have been proposed. [27] proposed in 2017 a CP-ABE where each weighted attribute only adds a logarithmic overhead to the ciphertext. Their construction is based on 0- and 1-Encodings of the weights. In 2021, [17] a new Weighted CP-ABE is proposed using the same technique. While tested against similar weighted CP-ABE schemes, the latter provided the best performance results.

## 3.3 ABE Extensions

In many use-cases, simple ABE schemes come with drawbacks and limitations, making them unusable. Therefore, the research community has invested a lot of effort in ABE, proposing numerous extensions and add-ons to increase the usability of ABE in practical systems.

**Outsourced decryption** Green et al. [11] was the first to introduce an ABE scheme with outsourced decryption, motivated by ABE's use cases in low-powered devices. The idea is to construct a mechanism to work out most of the decryption in cloud, and send

to the client a partially decrypted result. Then, the client should be able to decrypt it with less effort.

**Access Revocation** [2] proposes an ABE scheme which supports both direct and indirect revocation. The user is able to select between these two mechanisms at the encryption phase. The decryption keys are of a single type, and they can decrypt ciphertexts constructed in any method. [28] proposed a KP-ABE scheme which uses a simplified form of attribute revocation to ensure data deletion.

**Multi-authority ABE** In Cloud systems, users are often reluctant to trust a single entity for managing they decryption keys. Therefore the multi-authority setting is very important in this context, and numerous other ABE schemes with this property appeared, starting with the one proposed by Chase in [5].

**Predicate Encryption and Hidden-policy ABE** For some use cases, the fact that attributes or access policy are usually stored in plain in the ciphertext in KP-ABE, and CP-ABE could represent a problem. Therefore, *predicate encryption*[14] comes with two security notions, namely *payload hiding* and the stronger one - *attribute hiding* to solve this problem in KP-ABE. Moreover, in CP-ABE we have *policy-hiding* schemes, such as [16, 19].

# Chapter 4

# Weighted Attribute-based Encryption

## 4.1 Weighted Access Structures in ABE

The problem of constructing ABE schemes with expressive access structures is continuously explored in order to find better and better ones, suited for various purposes. Beside access trees and Boolean circuits, we can also define weighted access structures. We consider two different versions of weighted access structures, for which we propose two different construction. First, the *simple weighted* versions, where each attribute has a weight associated, and the access policy has some thrshold on the leaf nodes. If an attribute has it's weight greater than the threshold of the lead node, then it can be used in the decryption process. The *fully weighted* access structure assumes that on every wire of an access tree we have some weight. A threhold gate is satisfied if the sum of the weights of the satisfied input wires is greater than the threshold.

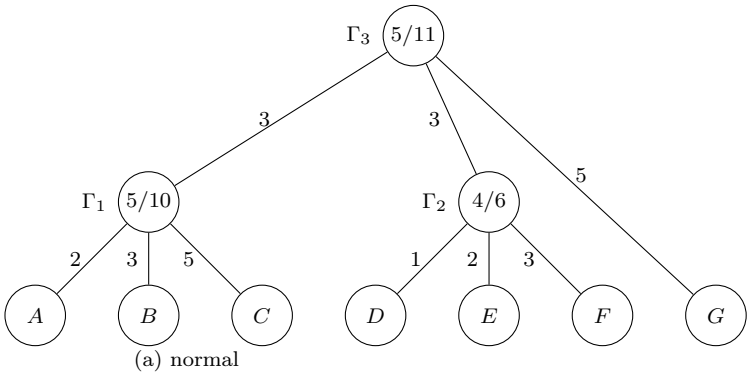


FIGURE 4.1: Example of fully weighted access tree

**Algorithm 1:** transform($\mathcal{T}$)

$\ell_N \leftarrow log_2(N)$;
**for** *every leaf node* $\Gamma$ *in* $\mathcal{T}$ *corresponding to a weighted attribute* **do**
  Let $\omega_\Gamma = (\overline{b_\ell \cdots b_1 b_0})_2$ the minimum required weight ;
  Find $i$ such that $b_i = 1$ and $b_{i-1} = \cdots = b_0 = 0$ ;
  `// Lest significant bit from` $\omega_\Gamma$ `set to 1`
  $Parent \leftarrow \Gamma$ ;
  `// This is a temporary variable to store the last gate created`
  **for** *every* $j$ *in* $\{\ell, \cdots i+2, i+1\}$ **do**
    $\Gamma_j \leftarrow$ new leaf node ;
    **if** $b_j = 1$ **then**
      **if** $b_j = b_{j+1}$ **then**
        $k_{Parent} \leftarrow k_{Parent} + 1$ `// increases the threshold, as we will add another child to this node, but we want it to remain an AND node.`
      **else**
        $Tmp \leftarrow$ new (2/2)-gate (simple AND gate). ;
        $parent(Tmp) \leftarrow Parent$ ;
        $Parent \leftarrow Tmp$ ;
    **else**
      **if** $b_j = b_{j+1}$ **then**
        continue ;
      **else**
        $Tmp \leftarrow$ new (1/2)-gate (simple OR gate). ;
        $parent(Tmp) \leftarrow Parent$ ;
        $Parent \leftarrow Tmp$ ;
    $parent(\Gamma_j) = Parent$ `// Link the leaf node to the last node created`
  $parent(\Gamma_i) = Parent$ `// Link the last leaf, corresponding to bit` $i$`, to the last node created`

## 4.2 Simple Weighted KP-ABE scheme

For the simple weighted access structure, our strategy is to provide a transformation algorithm for each access tree into an weighted access structure. We do this using the Algorithm 1. The algorithm simply converts a threshold into a small tree equivalent to the "greater than" comparison, while also adding two optimizations:

- any $k$ consecutive $OR$ gates can be compressed in one "1 out of $k+1$" threshold gate.
- any $k$ consecutive $AND$ gates can be compressed in one "$k+1$ out of $k+1$" threshold gate.

Out ABE scheme is an adaptation of Goyal et al.'s scheme[10]. First, it generates multiple attributed in the setup phase, and then, every access structure is converted to a threshold access structure using the Algorithm 1.

**Theorem 4.1.** *The Weighted KP-ABE system is secure in the Key-Policy Attribute-based Selective-Set Model under the bilinear Decisional Diffie-Hellman problem.*

This theorem is proved by showing that if there exists a non-negligible advantage adversary for $W-KP-ABE$, then we can also construct an adversary with non-negligible advantage for $GPSW$.

**Efficiency** We have identified a single KP-ABE scheme with support for weighted access structures, namely the one in [18]. For each attribute in the access structure, the key size grows linearly in the value of the attribute. Also, the encryption and decryption times are affected by this expansion, growing also linearly in the attribute's weight.

Using the theoretical and practical analysis from [17], we can conclude that LYL+ [17] and CABE [27] outperform the other weighted CP-ABE schemes by a considerable margin, being the only ones with logarithmic expansion for each weighted attribute. We have a computational overhead of $hw(N)$ per weighted attribute, which is slightly better than the $log(N)$ provided by LYL+ and CABE.

## 4.3 Fully Weighted ABE scheme

We say that $(\Gamma, w, t)$ is a $(t, w)$-weighted thrshold gate, where $w$ represents a vector of weights for the input wires, and $t$ is the threshold. A weighted gate is satisfied if and only if the sum of weights of the satisfied input nodes is greater or equal to the thrshold. We can see an example of a $(t = 5, w = (2, 3, 5))$-weighted threshold gate in Figure 4.2.

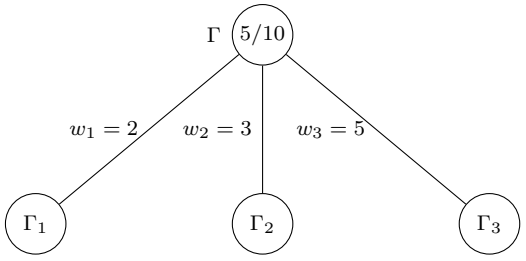


FIGURE 4.2: Example of $(t = 5, w - (2, 3, 5))$-weighted threshold gate

In our KP-ABE scheme, we will use two procedures, called $Share\_FW$ and $Recon\_FW$ which will share and reconstruct a secret over a fully weighted access tree. These procedures work as follows:

$Share\_FW(\Gamma, Out(\Gamma))$ This procedure shares the list of parts $Out(\Gamma)$ through the access structure rooted in the $(t, w)$-weighted threshold node $\Gamma$. Recall that the threshold of the gate is $t_\Gamma$ and the number of input wires is $n_\Gamma$. Then, share the values $Out(\Gamma)$ as follows:

1. For each $Out(\Gamma, i)$ in $Out(\Gamma)$, choose a random polynomial $P_{i,\Gamma}(0)$ of degree $t_\Gamma - 1$, such that $P_{i,\Gamma}(0) = Out(\Gamma, i)$

2. For each input wire $j$, denote with $\omega_j$ the sum of weights of the wires $1, 2, ..., j-i$. Therefore, when constructing a polynomial, wire $j$ should receive its evaluations in points $\omega_j + 1, \omega_j + 2, ..., \omega_j + w_j$. For each value $Out(\Gamma, i)$ we will add $w_i$ values to $In_j(\Gamma)$, computed as follows: assign as the $i$-th input value of that wire, the

coordinate $j$ of the polynomial $P_i$. More formally,

$$In_j(\Gamma) = In_j(\Gamma)||\langle P_{i,\Gamma}(\omega_j + 1), P_{i,\Gamma}(\omega_j + 2), ...P_{i,\Gamma}(\omega_j + w_j)\rangle$$

This will result in a total number of $w_j \cdot |Out(\Gamma)|$ values in the $j$-th wire.

3. recursively apply $Share\_FW(\Gamma_i, In_i(\Gamma))$ for each child $\Gamma_i$ of $\Gamma$.

This procedure can be applied on top of the weighted access tree, using the root node $\Gamma_0$, and a list consisting only of the secret to be shared, $y$.

The reconstruction phase works as follows:

$\underline{Recon\_FW(\Gamma, D, E)}$:

For the terminal nodes $\Psi$ in the access tree, use the ciphertext $E$ and decryption key $D$, re-compute the input values as follows:

$$In(\Psi, i) = \begin{cases} e(D_\Psi, T_x^s) = e(g,g)^{q_\Psi(0)\cdot s}, & \text{if } x = attr(\Gamma) \in \mathcal{A} \\ \perp, & \text{otherwise} \end{cases}$$

For an internal $(w, t)$-weighted threshold node $\Gamma$, if this node is not satisfied (sum of weights of satisfied children is smaller than the threshold), then output $\neq\perp$) Otherwise, considering $S(\Gamma)$ a list of the satisfied wires of $\Gamma$, and $\Gamma_z$ to be the $z$-th child of $\Gamma$. $S'(\Gamma)$ will be the list of points where the polynomials of the shares from $S(\Gamma)$ have been evaluated to. More concrete:

$$S'(\Gamma) = \cup_{z \in S(\Gamma)}\{\omega_z + 1, \omega_z + 2, ..., \omega_z + w_z\}$$

Denote with $\omega_j$ the sum of weights of the wires $1, 2, ..., j - i$. Thus, wire $j$ contains evaluations of some polynomials in points $\omega_j + 1, \omega_j + 2, ..., \omega_j + w_j$.

$$
\begin{aligned}
Out(\Gamma, i) &= Recon(\Gamma, i, D, E) \\
&= \prod_{z \in S(\Gamma)} \prod_{j=1}^{w_z} Recon(\Gamma_z, (i-1) \cdot w_z + j, D, E)^{\Delta_{\omega_z + j, S'(\Gamma)}(0)} \\
&= \prod_{z \in S(\Gamma)} \prod_{j=1}^{w_z} (e(g,g)^{s \cdot P_{i,\Gamma}(0)})^{\Delta_{\omega_z + j, S'(\Gamma)}(0)} \\
&= e(g,g)^{s \cdot P_{i,\Gamma(0)}}
\end{aligned}
$$

Note that in order to reconstruct the value $Out(\Gamma, i)$ we need to iterate over the polynomial evaluations of the respective share. For some child node $z$, these values are the values from $In_z(\Gamma) = Out(\Gamma_z)$ from indices $(i-1) \cdot w_z$ to $i \cdot w_z$. These values represent the evaluation of the polynomial in the points $\omega_z + 1, \omega_z + 2$

The full ABE scheme is an adaptation of [10] to use the above mentioned $share\_FW$ and $recon\_FW$ procedures.

**Theorem 4.2.** *The Weighted KP-ABE system is secure in the Key-Policy Attribute-based Selective-Set Model under the bilinear Decisional Diffie-Hellman problem.*

*Proof.* The proof will follow the same outline as the security proof of [10] or [24], with the required modifications on the *fake_ share* and *fake_ recon*.

□

# Chapter 5

# Towards Attribute-based Encryption for Boolean circuits

While searching for access structures more expressive then access trees for which we can construct efficient ABE schemes, we have looked into Compartmented access strucutres (CAS), since [25] proved that it is impossible to express a CAS as an access tree.

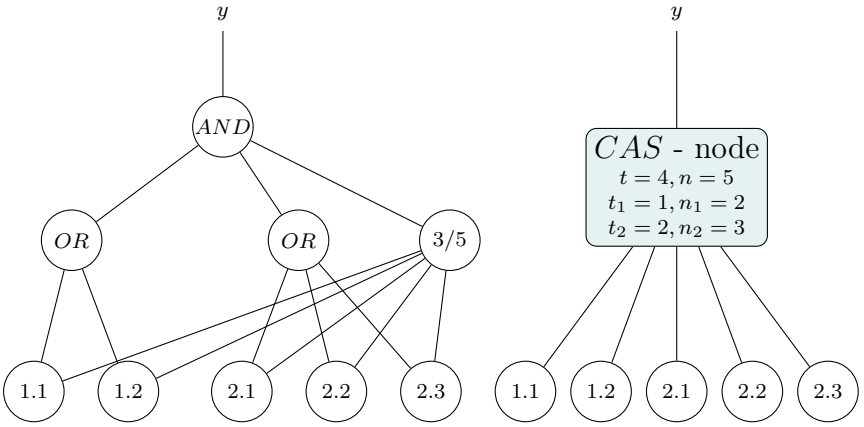


FIGURE 5.1: A sub-circuit and an equivalent CAS-node

We have provided two constructions for CASs, incrementally rising the efficiency of our schemes. The first construction is mostly based on the ABE scheme from [24], with some optimizations specific to CASs applies to it, while the second scheme is based on the Ghodosi et al.'s [9] secret sharing for CAS. We have further showed for the second construction that is can be used for more complex access structures built on top of CAS.

**The first construction** The first scheme [25] is an adaptation of the ABE scheme for Boolean circuits from [24] to the special case of Compartmented access structures

(CAS). More precisely, a CAS can be represented as a Boolean circuit consisting of threshold gates, with three levels. However, this approach generates two shares for each participant in the secret sharing pahse, and is far from optimum, as we can see in the next part.

## 5.1 ABE for CAS-circuits

Ghodosi et al. [9] scheme can actually be adapted for the ABE system, with only a small trick being required for the security proof. Moreover, the secret sharing technique can be applied recursively without blowing up exponentially the size of the secrets.

**Compartmented Nodes** Since we will use CASs as parts of larger and more complex access structures, we will define a $CAS$-node $\Gamma$ as a special gate modelling a CAS. The gate has a single output wire, and a number of input wires equal to the total number of participants. Each gate is also defined by a general thrshold $t$ and a threshold for each compartment $t_i$. Then, we define an access structure built upon these types of gate.

**Definition 5.1.** [13] A $CAS$-circuit is a tree formed out of $CAS$-nodes

**The ABE-CAS construction**

We describe the share_CAS procedure, which shares a value through a CAS node. This procedure can be applied recursively, resulting in a secret sharing over a CAS-circuit.

*share_CAS*$(\Gamma, y)$**:**

1. Let $T = t - \sum_{i=1}^{k} t_i$.

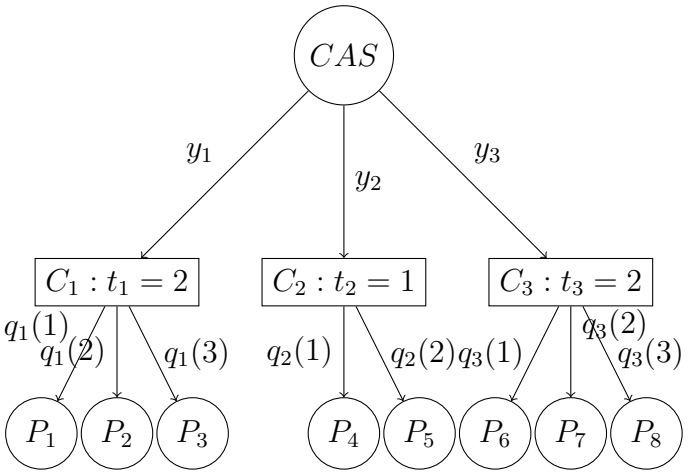


FIGURE 5.2: Secret sharing though a $CAS$

2. For each compartment, choose randomly the partial secret $y_i$ and a public parameter $p_\Gamma$ from $\mathbb{Z}_p$ such that $y_1 + y_2 + \ldots y_k + p_\Gamma = y$.

3. Then, for each compartment $i = 1, \ldots, k$: select randomly and uniformly $t_i - 1$ values $a_{i.1}, ..., a_{i.t_i-1}$ from $\mathbb{Z}_p$ corresponding to each compartment $i$, $i = 1, ..., k$.

4. Choose randomly and uniformly $T$ values $\beta_1, \ldots \beta_T$ in $\mathbb{Z}_p$.

5. Determine a sequence of $k$ polynomials, $q_i(x) = y_i + a_{i.1}x + \ldots + a_{i.t_i-1}x^{t_i-1} + \beta_1 x^{t_i} + \ldots + \beta_T x^{t_i+T-1}$ for every level $i$.

6. Assign the shares for each input node: $In_{i.j} = q_i(j)$, and publish $P(\Gamma) = g^{p_\Gamma}$ as the gate's public parameter.

In the reconstruction phase, for each value $\alpha$ associated to some wire at the sharing phase, we will have a corresponding value $g_T^{\alpha s}$ attached to the same wire during the reconstruction phase.

*recon_CAS*$(\Gamma, P(\Gamma) = g^{p_\Gamma}, S = g^s, \langle e(g,g)^{q_i(j)s}, ..., \rangle)$**:**

During the reconstruction phase in out ABE system, for each satisfied input wire $i.j$ of the $CAS$-node $\Gamma$, we will have some value $e(g,g)^{q_i(j)s}$, which represents the

result of an equation of form:

$$e(g,g)^{y_i s} \cdot e(g,g)^{a_{i.1} j s} \cdot \ldots \cdot e(g,g)^{a_{i.t_i-1} j^{t_i-1} s} \cdot$$
$$e(g,g)^{\beta_1 j^{t_i} s} \cdot \ldots \cdot e(g,g)^{\beta_T j^{T+t_i-1} s} = e(g,g)^{q_i(j)s}$$

which is equivalent with

$$e(g,g)^{s(y_i + a_{i.1} j + a_{i.2} j^2 + \ldots + a_{i.t_i-1} j^{t_i-1} + \beta_1 j^{t_i} + \ldots + \beta_T j^{T+t_i-1})} = e(g,g)^{q_i(j)s}$$

We need to select from each compartment $\ell_i$ wires, namely $j_{i.1}, j_{i.2}, \ldots j_{i.\ell_i}$, such that $\ell_1 \geq t_1, \ell_2 \geq t_2, \ldots \ell_k \geq t_k$ and $\sum_{i=1}^{k} \ell_i = t$.

Putting all such equations together from all input wires, we obtain an equation system which can be solved only if the the CAS node is satisfied.

**KP-ABE for *CAS*-circuits** Our KP-ABE scheme for *CAS*-circuits[13] is built over [10], by using our *share_CAS* procedure in the key generation step, alongside with *recon_CAS* on the decryption phase. We use two intermediate procedures *share_CAS_circuit* and *recon_CAS_circuit*, which recursively apply *share_CAS* and *recon_CAS* to the nodes in the CAS-circuit.

**Theorem 5.2.** *Our scheme is secure in the selective model under the decisional bilinear Diffie-Hellman assumption. [13]*

*Proof.* In the full version of this thesis □

## 5.2 Efficiency and Improvements

Beside the obvious benefit of creating an efficient ABE construction for CAS-circuits, the CAS sharing technique allows to optimize existing schemes by replacing a subcircuit which can be modelled as a CAS. While it may not seem a great benefit at the

TABLE 5.1: Worst case decryption key size

| ABE system | Bolean Formulae | CAS-circuit | Boolean circuits |
|---|---|---|---|
| Goyal et al. [10] | $n$ | Unsupported | Unsupported |
| Tiplea-Dragan [24] | $n$ | $nj + n + j^r$ | $nj + n + j^r$ |
| Hu-Gao [12] | $n$ | $n + j^r$ | $n + j^r$ |
| Ours-1 ($CAS$-circuit) | $n + q$ | $n + q$ | Unsupported |
| Ours-2 (general circuit) | $n$ | $n + q$ | $n + j^r$ |

first sight, we can actually observe that a lot of circuits can be expressed as CASs, by creating virtual compartments with threshold equal to zero.

**Comparison with other ABE systems** When compared to other schemes, ours clearly extends the usable access structures in the context of ABE. Regarding the notations used, we denoted with $n$ the number of input nodes in an access structure, and with $r$ the number of FO-gates. The FO-gates are considered to have $j$ input wires each, and the total number of internal gates to be $q$. Using these notation, we have compiled a complete analysis in Table 5.1. "Ours-1" represents the vanilla scheme proposed by us, using $CAS$-circuit access structures. The second one, "Ours-2", features, beside $CAS$-nodes, also threshold gates from [10] and $FO$-gates from [24].

## 5.3 Boolean Circuits and MSP

The open problem of constructing efficient secret sharing for general (unrestricted) monotone Boolean circuits has been widely studied, especially in the context of ABE. Using Beimel's Theorem of equivalence between LSSS and MSP [3], we can study focus on MSP lower bounds in order to prove lower bound for LSSS.

Going along this line, we provide a simple proof which states that there can not be constructed a general *ideal* LSSS for unrestricted monotone Boolean circuits.

**Theorem 5.3.** *[20] There is no ideal linear secret sharing scheme for the class of access structures represented by monotone Boolean circuits.*

*Proof.* Omitted due to space limitation. □

**Fully Weighted ABE and Boolean circuits** We noticed that the special circuit (refferd to as U-gate) which we discussed in section 5.3 can also be expressed as a fully weighted access structure. An important consequence of this is the following theorem:

**Theorem 5.4.** *There are no ideal LSSS representing fully weighted access structures.*

*Proof.* Suppose there exists an ideal LSSS for representing fully weighted access structures. Then, we can construct an ideal LSSS for a U-node, which is impossible due to Theorem 5.3 □

**Exponential Lower Bound for LSSS for Boolean circuits** [21] proved that a monotone access structure which is in the *monotone P* circuit class can only be expressed by exponential MSPs. This implies that we cannot construct sub-exponential ABE schemes for Boolean circuits from LSSS.

# Chapter 6

# Heuristic Optimizations in Attribute-based Encryption

While accepting the possibility that it is impossible to build efficient ABE schemes for Boolean circuits, we have switched our attention to optimizations which may be made to existing schemes. One possible approach would be to rewrite the Boolean circuit into an equivalent form, for which the secret sharing produces less shares, as we can see in Figure 6.1 As a standard for comparison we will use the state of the art scheme in ABE for Boolean circuits schemes using bilinear maps from [24]. In the secret sharing algorithm from this scheme, the number of shares that will be associated to each input wire of the circuit is equal to the number of paths from the respective input node to the output node of the circuit. When writing the circuit in the *multiplicative equation* form, we can observe that every path in the circuit corresponds to a literal in the formula. We will consider the cost function $c(\mathcal{C})$ = the number of shares the secret sharing technique in ABE is producing on $\mathcal{C}$ Moreover, to simplyfy notation we use addition to represent OR operations, and multiplication for AND .

## 6.1 Our Approach

We have made two different attempts for the optimization of Boolean circuits for ABE systems. The first one was based on replacing parts of a circuit with sub-circuit which have a better secret sharing technique. This did not lead to great results, due to the high computational power required and lack of flexibility. The second method was based on the Abstract Syntex Tree (AST), and tried to directly optimize the formula using three operations on the tree:

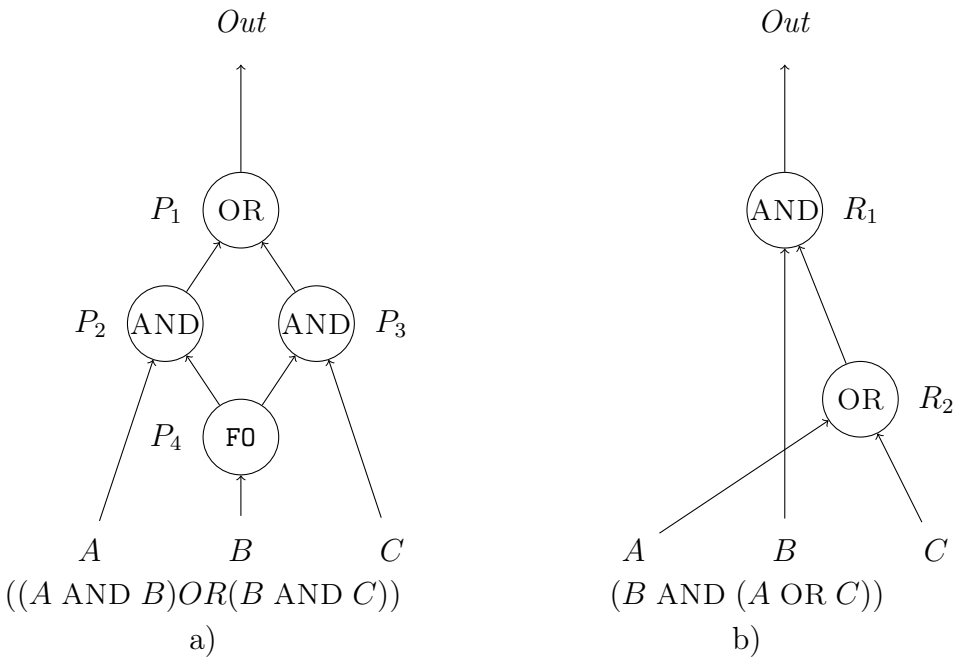


FIGURE 6.1: Two equivalent Boolean circuits, alongside their equivalent logical formulas

1. *factorization* – We make use of the fact that the `OR` is distributive under `AND` and viceversa. Therefore, a formula such as $AB + AC$ can be re-written to $A(B + C)$. This obviously reduces the total number of literals, leading to a lower cost.

2. *defactorization* – This is the inverse operation of *factorization*. It works by making the cross-product between the terms of two parenthesis from a conjunction. For example, $(A + B)(C + D)$ would convert into $AC + AD + BC + BD$. Note that the resulting formula after the *defactorization* will have a strictly higher cost.

3. *absorption* is the operation of eliminating 1s after the *factorization* and *defactorization*. For example, $A + AB$ can be factorized into $A(1 + B)$, the latter which is equivalent to $A$.

Using these operations, we have constructed various heuristic algorithms.

**Hill Climbing** Our first approach in using these techniques in some heuristics was a Hill Climbing algorithm. Since the factorization will always reduce the cost of the circuit, we used

to search a *local* minimum. Therefore, our approach was straight-forward: while it is possible, choose two nodes that can be factorized and apply the operation. Obviously, depending on the choices we could end up with different results.

**Simulated Annealing** Our second attempt was using the probabilistic method described in [15], known as *Simulated Annealing*. The basic idea is to consider at each iteration of the algorithm a solution of the problem. This solution will be assigned a probability to be accepted, based on its score and temperature: The higher the score and the temperature, the higher the probability for it to be accepted.

**Custom Heuristic** Since we felt that we can obtain better results with a different approach, we also constructed a custom heuristic algorithm, which was meant to combine the *factorization* and *defactorization* algorithms in a simpler manner than the one used in *simulated annealing*. The pseudocode for our algorithm is presented below, in Algorithm 2

**Algorithm 2:** Custom Heuristic for circuit optimization.

```
for k ∈ {1, 2, ..., k_max} do
    if random(0, 1) < (k_max − k)/(5k_max) then
        operation ← defactorization
    else
        operation ← factorization
    u ← choose neighbor of u using operation ;
if cost(u) < cost(u_min) then
    u_min ← u
apply factorization to u_min until formula is not improved anymore
```

**Iterated Versions** We have also provided iterated versions for our algorithms, in the hope of finding the global optimum, or, at least, a better local optimum.

## 6.2 Practical Tests

We have tested our heuristics agains four datasets, with variable count between 15 and 25, and literal count between 20 and 200. The last dataset contains circuits modelling comparisons over numeric attributes, represented in binary.

Our runs have been summarized in the Table 6.1: For each dataset we have three result observed for each of the six algorithms: *mean optimization* (M. Opt.), *best over iterations* (BOI), and *average running time* (Avg. RunTime). On the first column we have the six algorithms we have tested, identified by their initials. Their iterated versions can be easily identified by the letter "I" in front of them.

| | Dataset 1 | | | Dataset 2 | | |
|---|---|---|---|---|---|---|
| | M. Opt. | BOI | Avg. RunTime | M. Opt. | BOI | Avg. RunTime |
| HC | 15.0 % | 16.3 % | 0.00 s | 35.1 % | 41.9 % | 0.00 s |
| IHC | 16.3 % | 16.3 % | 0.08 s | 42.0 % | 42.1 % | 0.34 s |
| SA | 26.9 % | 43.1 % | 0.16 s | 41.0 % | 59.1 % | 0.86 s |
| ISA | 40.1 % | 46.9 % | 2.39 s | 57.8 % | 66.0 % | 13.1 s |
| CH | 24.8 % | 44.0 % | 0.13 s | 35.8 % | 61.8 % | 0.38 s |
| ICH | 43.0 % | 47.8 % | 2.74 s | 61.3 % | 68.6 % | 7.50 s |
| | Dataset 3 | | | Dataset 4 | | |
| | M. Opt. | BOI | Avg. RunTime | M. Opt. | BOI | Avg. RunTime |
| HC | 42.6 % | 56.5 % | 0.01 s | 4.8 % | 7.2 % | 0.00 s |
| IHC | 56.5 % | 56.5 % | 2.03 s | 7.2 % | 7.2 % | 0.11 s |
| SA | 43.0 % | 58.6 % | 0.76 s | 32.3 % | 50.0 % | 0.10 s |
| ISA | 59.3 % | 63.1 % | 13.6 s | 48.5 % | 50.4 % | 1.44 s |
| CH | 39.8 % | 59.4 % | 0.42 s | 20.8 % | 47.8 % | 0.14 s |
| ICH | 60.6 % | 64.9 % | 7.82 s | 48.5 % | 50.4 % | 2.94 s |

TABLE 6.1: (Iterated) Hill Climbing/ Simulated Annealing/ Custom Heuristic results on each dataset

# Chapter 7

# Conclusions and Future Work

This work focused on finding expressive, yet efficient access structures for ABE schemes developed with bilinear maps. We have pointed out that current results already prove that LSSS for Boolean circuits imply an exponential expansion of shares. Also, while the previous efficient ABE were limited to Boolean formulae or threshold trees, after our work, we extend the class of access structures for which we know efficient implementations of ABE to the access structure which we have introduced as CAS-tree.

In addition, we have proposed and implemented several heuristics which transform a Boolean circuits into an equivalent circuit, for which the secret sharing schemes will produce fewer shares.

Also regarding ABE expressiveness, we developed two new weighted ABE scheme for bilinear maps. The first scheme offers a slight advantage in implementing "greater than" comparisons over numerical attributes. The second scheme is the first *fully weighted* ABE scheme.

**Open Problems and Future Work** From our work, various open problems stand out, requiring more study and exploration. One of the most interesting open problems is to build sub-exponential schemes for *fully weighted* ABE. Another important open problem is whether is it possible to construct polynomial multi-linear secret sharing schemes for Boolean circuits. We believe the answer is negative to this question, and we also believe that a potential proof could be based on U-trees or similar access structures.

Also, we believe that the gap between CAS-trees and Boolean circuits which represent the limits of access structures for which we can build efficient ABE constructions, could be tightened even more. There could be many more access structure which are not yet addressed.

# Bibliography


[1] N. Attrapadung. Fully secure and succinct attribute based encryption for circuits from multi-linear maps. *Cryptology ePrint Archive*, 2014.

[2] N. Attrapadung and H. Imai. Attribute-based encryption supporting direct/indirect revocation modes. In *IMA international conference on cryptography and coding*, pages 278–300. Springer, 2009.

[3] A. Beimel et al. Secure schemes for secret sharing and key distribution. 1996.

[4] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*, pages 321–334. IEEE, 2007.

[5] M. Chase. Multi-authority attribute based encryption. In *Theory of Cryptography Conference*, pages 515–534. Springer, 2007.

[6] C. C. Drăgan and F. L. Ţiplea. Key-policy attribute-based encryption for general boolean circuits from secret sharing and multi-linear maps. In *International Conference on Cryptography and Information Security in the Balkans*, pages 112–133. Springer, 2015.

[7] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In *Annual Cryptology Conference*, pages 479–499. Springer, 2013.

[8] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure attribute based encryption from multilinear maps. *Cryptology ePrint Archive*, 2014.

[9] H. Ghodosi, J. Pieprzyk, and R. Safavi-Naini. Secret sharing in multilevel and compartmented groups. In *Australasian Conference on Information Security and Privacy*, pages 367–378. Springer, 1998.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.

[11] M. Green, S. Hohenberger, B. Waters, et al. Outsourcing the decryption of abe ciphertexts. In *USENIX security symposium*, volume 2011, 2011.

[12] P. Hu and H. Gao. A key-policy attribute-based encryption scheme for general circuit from bilinear maps. *IJ Network Security*, 19(5):704–710, 2017.

[13] A. Ionita. Optimizing attribute-based encryption for circuits using compartmented access structures. *Cryptology ePrint Archive*, 2023.

[14] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Advances in Cryptology–EUROCRYPT 2008: 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings 27*, pages 146–162. Springer, 2008.

[15] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[16] J. Lai, R. H. Deng, and Y. Li. Fully secure cipertext-policy hiding cp-abe. In *Information Security Practice and Experience: 7th International Conference, ISPEC 2011, Guangzhou, China, May 30–June 1, 2011. Proceedings 7*, pages 24–39. Springer, 2011.

[17] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava. An efficient ciphertext-policy weighted attribute-based encryption for the internet of health things. *IEEE Journal of Biomedical and Health Informatics*, 2021.

[18] X. Liu, H. Zhu, J. Ma, J. Ma, and S. Ma. Key-policy weighted attribute based encryption for fine-grained access control. In *2014 IEEE International Conference on Communications Workshops (ICC)*, pages 694–699. IEEE, 2014.

[19] T. Nishide, K. Yoneyama, and K. Ohta. Attribute-based encryption with partially hidden encryptor-specified access structures. In *Applied Cryptography and Network Security: 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings 6*, pages 111–129. Springer, 2008.

[20] I. Oleniuc and A. Ionita. Secret sharing limitations over boolean circuits. *Computer Science Journal of Moldova*, to appear, 2025.

[21] R. Robere, T. Pitassi, B. Rossman, and S. A. Cook. Exponential lower bounds for monotone span programs. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 406–415. IEEE, 2016.

[22] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 457–473. Springer, 2005.

[23] E. Shi, J. Bethencourt, T. H. Chan, D. Song, and A. Perrig. Multi-dimensional range query over encrypted data. In *2007 IEEE Symposium on Security and Privacy (SP'07)*, pages 350–364. IEEE, 2007.

[24] F. L. Ţiplea and C. C. Drăgan. Key-policy attribute-based encryption for boolean circuits from bilinear maps. In *International Conference on Cryptography and Information Security in the Balkans*, pages 175–193. Springer, 2014.

[25] F. L. Tiplea, A. Ionita, and A. Nica. Practically efficient attribute-based encryption for compartmented access structures. In P. Samarati, S. D. C. di Vimercati, M. S. Obaidat, and J. Ben-Othman, editors, *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications, ICETE 2020 - Volume 2: SECRYPT, Lieusaint, Paris, France, July 8-10, 2020*, pages 201–212. ScitePress, 2020.

[26] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie. Attribute-based data sharing scheme revisited in cloud computing. *IEEE Transactions on Information Forensics and Security*, 11(8):1661–1673, 2016.

[27] K. Xue, J. Hong, Y. Xue, D. S. Wei, N. Yu, and P. Hong. Cabe: A new comparable attribute-based encryption construction with 0-encoding and 1-encoding. *IEEE Transactions on Computers*, 66(9):1491–1503, 2017.

[28] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang. Efficient attribute-based encryption with attribute revocation for assured data deletion. *Information Sciences*, 479:640–650, 2019.