Efficient Attribute Based Encryption for Boolean Circuits

Alexandru Ioniţă[1,2][0000−0002−9876−6121]

[1] Simion Stoilow Institute of Mathematics of the Romanian Academy,
Bucharest, Romania
[2] Department of Computer Science,
Alexandru Ioan Cuza University of Iaşi, Iaşi, Romania
alexandru.p.ionita@gmail.com
http://www.students.info.uaic.ro/~alexandru.ionita

Abstract. We provide a new technique for secret sharing and reconstruction for Boolean circuits, applicable in ABE systems.
We show that our construction holds for Key-policy ABE and can be adapted also to Ciphertext-policy ABE. This is the most efficient solution for Attribute Based Encryption for circuits access structures using bilinear maps. Our KP-ABE system has decryption key of linear size in the number of attributes, and public parameters linear in the circuit size (Two public values for each FO-gate). We prove that our scheme is secure under the decisional bilinear Diffie-Hellman Assumption in the Selective Set Model.

Keywords: Attribute Based Encryption · Boolean Circuits · Access Control · Bilinear Maps · Decisional Bilinear Diffie-Hellman · Ciphertext Policy · Key policy.

1 Introduction

In Cloud Computing and IoT systems, access control over the shared data is one of the most important issues studied. As shown in various works ([18, 17]), attribute-based access control (ABAC) proves to be more suitable than other access control models, such as role-based access control for such large-scale systems. ABAC offers more flexibility, allowing the creation of expressive access policies, based on existing attributes in the system, in opposition to a manual assignment of roles by some administrator. One of the solutions for ABAC relies on attribute-based encryption (ABE). Introduced in [24] as a special case of identity-based encryption, ABE has presented a high interest in the last decade, improving the expressiveness and efficiency of such cryptographic schemes. European Telecommunications Standards Institute (ETSI) has published in 2018 technical specifications for implementing ABE in the cloud, IoT, and other Internet services (ETSI TS 103 458 [9]), specifying high-level requirements and recommendations for different use-cases.

Motivation. Suppose we need a system for storing personal documents, medical records and other sensitive data, from which you can be able to download you data on demand, at any time. Using conventional techniques, this would require a third party to store and control the access to data, which will have full access to all information in the system.

ABE offers an alternative, more viable solution: All documents will be shared in the cloud in a common encrypted database and each user controls, via an access policy, who can decrypt its data and when. Beside a strong security, two other important features are requires:

- expressive policy: In order to assure fine-grained access over the encrypted data
- efficiency: The data should be accessible from different devices, like personal computers or mobile phones, thus the encryption/decryption algorithms should run in decent time even on low resourced equipment.

The problem of achieving expressive access structure, while maintaining the cryptosystem fast enough to be used in practice has been widely studied, researchers trying to find the best trade-off between efficiency and expressiveness: [13, 5, 28, 15, 22, 14]

1.1 Related Work

Attribute-based encryption was introduced in [24] as a type of Identity Based Encryption[25]. The first ABE system was later introduced in [13], having a Boolean tree access policy associated with its key (hence the name *Key-Policy* ABE). Their construction is proven to be secure in the Selective Set Model under the DBDH Assumption.

The first Ciphertext-Policy ABE system was proposed by Bethencourt et al in [7], which also used an access tree as access structure. However, their security was proven only in the generic group model, rather than reducing it to a known theoretical problem. Later there were proposed CP-ABE systems proven to be secure under cryptographic assumptions in the standard model. [12, **?**],

ABE and Boolean Circuits The first approach which targeted Boolean circuits as access structure was presented in [10], where it was also shown why the usual sharing approach from Boolean formulas doesn't work. They have proposed both KP-ABE and CP-ABE schemes. However, their system is unusable in practice since it relies on multi-linear maps, for which there is no knows secure construction [2, 33]. Another similar work, which uses multi-linear maps, but increases the efficiency of the scheme is [8].

Another important work in this area is [28], where there is presented a scheme that relies solely on bilinear maps. The main drawback of this system is that the key size and decryption time can grow exponentially for some Boolean circuits. They also emit some public parameters for the FO-gates during the key generation procedure. As resulting from the comparisons in [29], it seems that

[28] has the best results in terms of Boolean circuits access structure applied to KP-ABE up to that date.

Meanwhile, a slightly more efficient scheme has been proposed: [15] offer a CP-ABE scheme for Boolean circuits by refining the work from [28], more precisely removing the public parameters of the FO-gates. [16] offers a similar performance for KP-ABE, by expanding the circuit, resulting in an equivalent on with fan-out one. The drawback is that the new circuit could grow exponentially in size.

Another notable progress in the Boolean circuits is the system recently proposed in [19], which offers efficient solutions for NC^1 circuits for both CP and KP ABE systems.

Other approaches to solve the circuit problem in ABE systems are based on the Learning With Error assumption. The first such system was constructed by Gorbunov, Vaikuntanathan and Wee in [11] [1] offers an efficient CP-ABE system in the symmetric Key setting for Circuits from LWE.

Although ABE schemes based on LWE have strong security guarantees, they are impractical due to the high computational cost determined by the large matrices involved [27].

Since cryptosystems relying on multilinear maps are not yet safe, and current approaches for Boolean circuits using bilinear maps are not efficient enough to be used in practice, leading to an exponential increase in the key size, at the moment, there are no practical ABE schemes for Boolean circuits.

Considering the observations above, we think that the best current trade-off between expressivity and efficiency, such that the resulting scheme could be used in practice, is achieved using Linear Secret Sharing Scheme (LSSS) matrices as access structures. Various works uses them as starting point for their constructions. [30, 20, 4, 23, 32]

Our Contribution We propose a new construction for Attribute Based Encryption for monotone Boolean circuits, applicable to both *key-policy and* ciphertext-policy. Our scheme is the most efficient construction proposed so far: It has decryption keys of linear size in the size of the circuit (more precisely, $2(r + n)$, where r is the number of the FO-gates and n the number of attributes). We offer concrete construction and security proof for the KP version of our construction. We show that our system is secure under Selective Set Model, under the decisional bilinear Diffie-Hellman (DBDH) Assumption. Using this approach, we extend the possible access policies that can be used in practice with ABE to Boolean circuits.

2 Preliminaries

Access Structures [6] Let $p_1, \ldots, p_n$ be a set of parties. A collection $A \subseteq 2^{\{p_1,\ldots,p_n\}}$ is monotone if $B \in A$ and $B \subseteq C$ imply that $C \in A$. An access structure is a monotone collection $A \subseteq 2^{\{p_1,\ldots,p_n\}}$ of non-empty subsets of $\{p_1, \ldots, p_n\}$. Sets in A are called authorized, and sets not in A are called unauthorized.

Boolean Circuits As presented in [28], a Boolean circuit is an access structure which has a number of input wires (which are not gate output wires), a number of output wires (which are not gate input wires), and a number of OR-, AND-, and NOT-gates. The OR- and AND-gates have two input wires, while NOT-gates have one input wire. All of them may have more than one output wire. That is, the fan-in of the circuit is at most two, while the fan-out may be arbitrarily large but at least one. A Boolean circuit is *monotone* if it does not have NOT-gates, and it is of fan-out one if all gates have fan-out one. In this paper all Boolean circuits have exactly one output wire. Boolean circuits of fan-out one correspond to Boolean formulas.

Bilinear maps Given G_1 and G_2 two multiplicative cyclic groups of prime order p, a map $e : G_1 \times G_1 \rightarrow G_2$ is called *bilinear* if it satisfies:

- $e(x^a, y^b) = e(x, y)^{ab}$, for any $x, y \in G_1$ and $a, b \in \mathbb{Z}_p$;
- $e(g, g)$ is a generator of G_2, 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.

KP-ABE General Model A Key-Policy Attribute Based Encryption scheme, as first described in [13], consists of four algorithms:

setup(λ) A randomized algorithm that takes as input the implicit security parameter λ and return the public and secret keys (MPK and MSK).

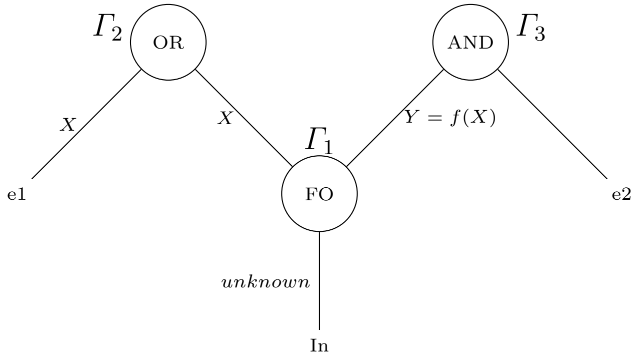
encrypt(m, A, MPK) A probabilistic algorithm that encrypts a message m under a set of attributes A with the public key MPK, and outputs the ciphertext E.

keygen($\mathcal{C}, MPK, MSK$) This algorithm generates receives an access structure, public and master keys, 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.

The Backtracking Attack The backtracking attack, as described in [28, 16], can occur in a Boolean circuit during the reconstruction phase of the secret, where an unauthorized set could gain access to the secret.

The main idea of such an attack is illustrated in Figure 1, where the values which comes as input (X) in the left wire is forwarded to the OR's gate right input wire. Then, using this value, we can compute the value from the AND's gate, using the value from the OR gate. The exact details of the attack depend on the sharing procedure used in the scheme.



(a) share

Fig. 1. The backtracking attack.

Selective-Set Model for ABE Goyal et al. propose in [13] a Selective-Set Model for ABE:

Init The adversary declares the set of attributes, γ, that he wishes to be challenged upon.

Setup The challenger runs the Setup algorithm of ABE and gives the public parameters to the adversary. Phase 1 The adversary is allowed to issue queries for private keys for many access structures A_j , where $\gamma / \in A_j$ for all j.

Challenge The adversary submits two equal length messages M0 and M1. The challenger flips a random coin b, and encrypts Mb with γ. The ciphertext is passed to the adversary.

Phase 2 Phase 1 is repeated.

Guess The adversary outputs a guess b' of b. The advantage of an adversary A in this game is defined as $Pr[b' = b] - \frac{1}{2}$.

Definition 1 ([13]). *An attribute-based encryption scheme is secure in the Selective-Set model of security if all polynomial time adversaries have at most a negligible advantage in the Selective-Set game.*

2.1 Notations

For a better understanding, we list some notations we made use of in our construction.

- As in [28], we make use of special Fan-Out (FO) gates that multiply the output of a node to simplify the explanation of our scheme.
- We will denote with Γ gates inside a circuit, and with Ψ, terminal nodes.
- $In_i(\Gamma)$ will be the i-th input wires associated with it.
- $Out_i(\Gamma)$ will be the i-th output wires associated with it.
- Some gates have a single input/output wire. We will refer it simply as $In(\Gamma)/Out(\Gamma)$.

- Since every wire could have more than one value associated to it in our system, we will denote $Out_i(\Gamma, j)$ The j-th value from the i-th output wire of the gate Γ.
- $L_1|L_2$ means list concatenation
- For some wire connecting gates Γ_1 and Γ_2, we always consider that both values on this wires are equal: $In(\Gamma_1) = Out(\Gamma_2)$ (that is using two notations for the same value attached to the wire)

3 Our Construction

Theorem 1. *An FO-gate with k output wires can be simulated with $k-1$ FO-gates limited at 2 output wires.*

Proof. By a simple construction of chained FO-gates.

Following Theorem 1, for simplicity, we will describe our construction only with FO-gates limited to 2 output wires.

Starting from [28]

We will first provide the *share* and *recon* procedures: For simplicity, we will consider that if on a wire there is a single value X, we actually have a pair consisting of $\langle X, X \rangle$. On actual implementations, we can discard this change, for efficiency reasons.

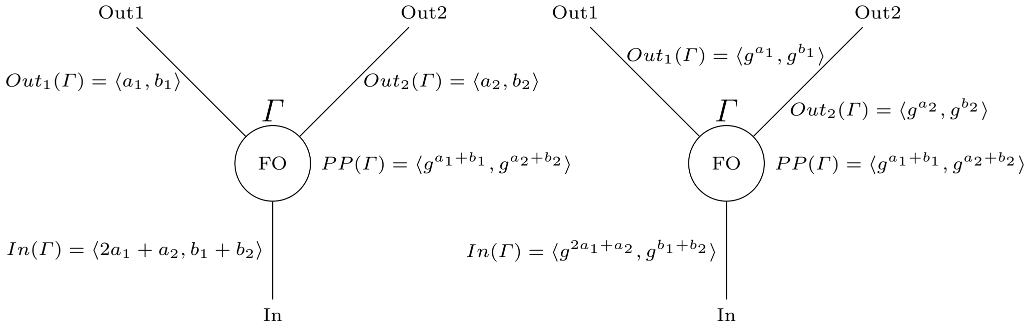
share$(s, \mathcal{C})$*:*

1. Initially, all gates of $\mathcal{C}$ are unmarked;
2. Assign s to the output wire of the circuit: $Out(\mathcal{C}) = \langle s, s \rangle$
3. Choose an unmarked gate Γ with all input wires defined
4. If Γ is an OR-gate, mark Γ and assign:
$$In_1(\Gamma) = In_2(\Gamma) = Out(\Gamma)$$
5. If Γ is an AND-gate, mark Γ and for each entry i in $Out(\Gamma)$, randomly generate $x_1(i)$ and then set $x_2(i)$ such that:
$$Out(\Gamma, i) = x_1(i) + x_2(i)$$
then assign:
$$In_1(\Gamma, i) = x_1(i)$$
$$In_2(\Gamma, i) = x_2(i)$$
for $i = 1, 2$
6. If Γ is an FO-gate, mark Γ and then, Compute values for the input wire of the gate $In(\Gamma)$, and for the public parameters of the gate $PP(\Gamma)$:

$$\begin{aligned} In(\mathit{\Gamma},1) &= 2Out_1(\mathit{\Gamma},1) + Out_2(\mathit{\Gamma},1) \\ In(\mathit{\Gamma},2) &= Out_1(\mathit{\Gamma},2) + Out_2(\mathit{\Gamma},2) \\ PP(\mathit{\Gamma},1) &= g^{Out_1(\mathit{\Gamma},1)+Out_1(\mathit{\Gamma},2)} \\ PP(\mathit{\Gamma},2) &= g^{Out_2(\mathit{\Gamma},1)+Out_2(\mathit{\Gamma},2)} \end{aligned}$$

Note that the gate has a single input wire $In(\mathit{\Gamma})$ with two values associated to it, and two public parameres, saved under $PP(\mathit{\Gamma})$.

7. Repeat steps [3, 4, 5, 6] until all gates are marked.
8. Return $\langle S, PP\rangle$, where $S(\mathit{\Psi}) = Out(\mathit{\Psi})$ for all terminal nodes $\mathit{\Psi}$.


Out1
Out2
$Out_1(\mathit{\Gamma}) = \langle a_1, b_1\rangle$
$Out_2(\mathit{\Gamma}) = \langle a_2, b_2\rangle$
$\mathit{\Gamma}$
FO
$PP(\mathit{\Gamma}) = \langle g^{a_1+b_1}, g^{a_2+b_2}\rangle$
$In(\mathit{\Gamma}) = \langle 2a_1 + a_2, b_1 + b_2\rangle$
In
Out1
Out2
$Out_1(\mathit{\Gamma}) = \langle g^{a_1}, g^{b_1}\rangle$
$\mathit{\Gamma}$
$Out_2(\mathit{\Gamma}) = \langle g^{a_2}, g^{b_2}\rangle$
FO
$PP(\mathit{\Gamma}) = \langle g^{a_1+b_1}, g^{a_2+b_2}\rangle$
$In(\mathit{\Gamma}) = \langle g^{2a_1+a_2}, g^{b_1+b_2}\rangle$
In

(a) share (b) recon

Fig. 2. sharing and reconstruction for FO gates.

Recon$(\mathcal{C}, P, V)$

1. Initially, all gates of $\mathcal{C}$ are unmarked;
2. $Out(\mathit{\Psi}) = V(attr(\mathit{\Psi}))$, for all leaf (starting) nodes $\mathit{\Psi}$. Mark these nodes.
3. Choose an unmarked gate $\mathit{\Gamma}$ with all input wires defined
4. If $\mathit{\Gamma}$ is an AND-gate, mark $\mathit{\Gamma}$ and assign:

$$Out(\mathit{\Gamma}, i) = In_1(\mathit{\Gamma}, i) \cdot In_2(\mathit{\Gamma}, i)$$

5. If $\mathit{\Gamma}$ is an OR-gate, mark $\mathit{\Gamma}$ and assign:

$$Out(\mathit{\Gamma}) = sup(In_1(\mathit{\Gamma}), In_2(\mathit{\Gamma}))$$

remark that $|In_1(\mathit{\Gamma})| = |In_2(\mathit{\Gamma})| = 2$ and, for any i, if $In_1(\mathit{\Gamma}) \neq In_2(\mathit{\Gamma})$ then either $In_1(\mathit{\Gamma}) =\perp$ or $In_2(\mathit{\Gamma}) =\perp$);

6. If Γ is an FO-gate, mark Γ and if all values from input wires are different from $\perp$, assign to the output wire of the circuits the values:

$$\begin{aligned} Out_1(\Gamma,1) &= \frac{In(\Gamma,1)\cdot In(\Gamma,2)}{PP(\Gamma,1)\cdot PP(\Gamma,2)} \\ Out_1(\Gamma,2) &= \frac{PP(\Gamma,1)}{Out_1(\Gamma,1)} \\ Out_2(\Gamma,1) &= \frac{In(\Gamma,1)}{Out_1(\Gamma,1)^2} \\ Out_2(\Gamma,2) &= \frac{In(\Gamma,2)}{Out_1(\Gamma,2)} \end{aligned}$$

   otherwise, assign $\perp$ to all output wires.
7. Repeat steps [3, 4, 5, 6] until all gates are marked.
8. return the value from the output wire of the circuit: $Out(\mathcal{C})$.

KP-ABE for BC:

setup(λ) This algorithm receives a security parameter λ, which is used to choose two multiplicative groups G_1 and G_2 of prime order p, g a generator of G_1, and a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. The set of attributes is defined by $\mathcal{U} = \{1, 2 \dots n\}$
It chooses random $y \in \mathbb{Z}_p$, and then generates random t_i, and sets the public key:

$$MPK = \langle p, G_1, G_2, e, g, n, Y = e(g,g)^y, T_i = g^{t_i} \rangle$$

and the master key:

$$MSK = \langle \alpha, (t_i, 1 \leq i \leq n) \rangle$$

encrypt(m, A, PK) The encryption algorithm receives a message m, and encrypts it under the set of attributes $A \subseteq \mathcal{U}$, with the public key *mpk*. Generate a random element s, and se the ciphertext as:

$$E = \langle A, E' = mY^s, T_i^s = g^{t_i s}, g^s \rangle$$

keygen($MPK, \mathcal{C}$) First, it generates a random s, and shares it through the circuit using the sharing procedure:

$$\langle S, PP \rangle = share(y, \mathcal{C})$$

Then, for every $i \in \mathcal{U}$ and $j \in \{1, 2\}$, output the decryption ket as:

$$DK = \langle D(i,j) = g^{S(i,j)/t_i}, PP \rangle$$

decrypt(E, DK) This algorithm receives a valid ciphertext and a decryption key, and returns the original message.

$$\text{Let } V(i,j) = \begin{cases} e(T_i^s, D(i,j) = e(g^{t_i s}, g^{S(i,j)/t_i}) = e(g,g)^{S(i,j)s}, & \text{if } i \in A \\ \perp & \text{otherwise} \end{cases}$$

for all $i \in A$, $j \in \{1,2\}$.

$$R = recon(\mathcal{C}, PP, V)$$

Then compute the message as:

$$m = E'/R = m \cdot e(g,g)^{ys}/$$

3.1 Extensions

CP-ABE We stress that our system is also appliable to CP-ABE. For example, we can simply replace the *share* and *recon* functions from [15], and with some other minimal adaptations (with regard to public parameters of the Boolean circuits) we will obtain a CP-ABE system with the same security properties as in [15], but more efficient.

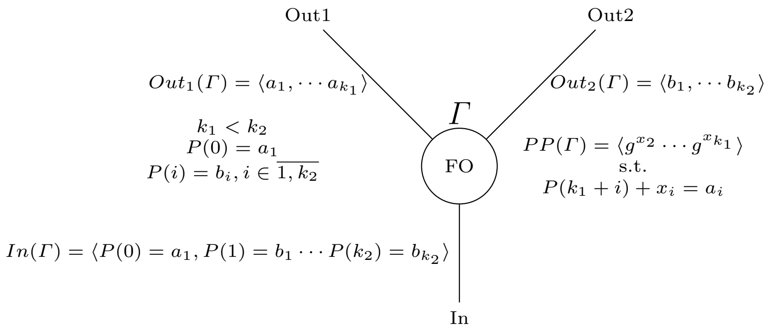

Out1
Out2
$Out_1(\Gamma) = \langle a_1, \cdots a_{k_1} \rangle$
$Out_2(\Gamma) = \langle b_1, \cdots b_{k_2} \rangle$
Γ
$k_1 < k_2$
$P(0) = a_1$
$P(i) = b_i, i \in \overline{1, k_2}$
FO
$PP(\Gamma) = \langle g^{x_2} \cdots g^{x_{k_1}} \rangle$
s.t.
$P(k_1 + i) + x_i = a_i$
$In(\Gamma) = \langle P(0) = a_1, P(1) = b_1 \cdots P(k_2) = b_{k_2} \rangle$
In

Fig. 3. sharing and reconstruction for FO gates.

FO-gate optimization. The idea of our construction started with the aim of optimizing the sharing procedure from [15] through the FO-gates. We will also discuss our initial idea, since it still may be used as a standalone optimization to [15] for certain Boolean circuits. We recall that in [28, 15] FO gates receive at least two lists of elements. Consider the FO-gate Γ from Figure 3: let $Out_1(\Gamma) = \langle a_1, \cdots a_{k_1} \rangle$ and $Out_2(\Gamma) = \langle b_1, \cdots b_{k_2} \rangle$ be the two lists. By symmetry, suppose $k_1 < k_2$. We construct a polynomial P of degree k_2 by choosing the following $k_2 + 1$ points:

$$P(0) = a_1 \text{ and } P(i) = b_i, i \in \overline{1, k_2}$$

Then, choose $x_2, x_3 \cdots x_{k_1}$ such that:

$$P(k_1 + i) + x_i = a_i, i \in \overline{2, k_1}$$

Finally, set the values from the input wire of the gate as:

$$In(\mathit{\Gamma}) = \langle P(0) = a_1, P(1) = b_1 \cdots P(k_2) = b_{k_2} \rangle$$

Using this optimization on FO-gates, the number of shares that are which are forwarded to child nodes is reduced to $max(|L_1|, |L_2|) + 1$. (Table 3.1)

Scheme	Input Size	Public Params
HuGao_CP[15]	$\lvert L_1\rvert + \lvert L_2\rvert$	0
Ours	$max(\lvert L_1\rvert, \lvert L_2\rvert) + 1$	$min(\lvert L_1\rvert, \lvert L_2\rvert) - 1$

Table 1. FO gate optimization

Threshold gates. As in [13, 7], our system could be extended to support threshold gates. The sharing procedure will be similar to the two proposed systems, with the mention that both shares that come as input to the threshold gate must be shared independently, as if there were two different gates.

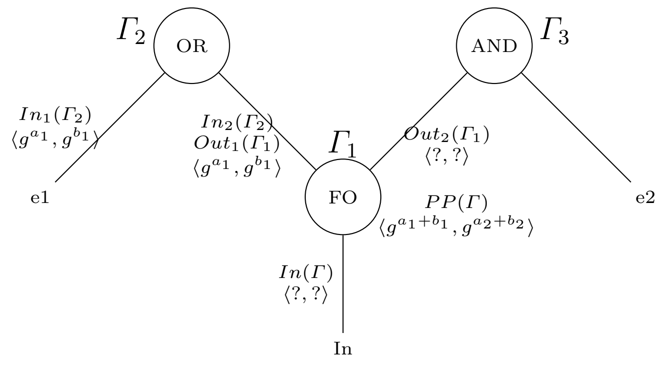
4 Security

Proposition 1. *For any input/output wire of the circuit, the values associated can be either both defined ($\neq\perp$), either both undefined ($\perp$).*

At first sight, our scheme could leak information about pairs of type $\langle X, \perp \rangle$, where one value is defined and the other one is undefined. If such value could enter an FO-gate, then, using the public parameters, we could compute the value which was initially undefined. However, due to Proposition 1 we see that such case will never happen.

Backtracking attack. We show that the backtracking attack does not occur in our construction. As it can be seen in Figure 4, we suppose that an attacker knows the values from the $\mathit{\Gamma}_2$'s right wire, $In_1(\mathit{\Gamma}_2) = \langle g^{a_1}, g^{b_1} \rangle$. Because this is an OR gate, he can also compute the values from the other wire: $In_2(\mathit{\Gamma}_2) = In_1(\mathit{\Gamma}_2) = \langle g^{a_1}, g^{b_1} \rangle$. (This is also possible for any threshold gates which are satisfied).

In the case of a backtracking attack, the attacker should be able to recover information about $\mathit{\Gamma}_3$'s right input wire using the information available on the FO-gate: $In_1(\mathit{\Gamma}_3) = Out_2(\mathit{\Gamma}_1)$. However, the only information available at the FO-gate is: $Out_1(\mathit{\Gamma}_1) = \langle g^{a_1}, g^{b_1} \rangle$ and $PP(\mathit{\Gamma}_1) = \langle g^{a_1+b_1}, g^{a_1+b_2} \rangle$, which clearly is not sufficient for computing $In_1(\mathit{\Gamma}_3) = \langle g^{a_2}, g^{b_2} \rangle$.



(a) share

Fig. 4. sharing and reconstruction for FO gates.

Theorem 2. *Our scheme is secure in the selective model under the decisional bilinear Diffie-Hellman assumption.*

Proof. Suppose that there exists a polynomial-time adversary $\mathcal{A}$ that has an advantage ϵ for our scheme in the Selective-Set model. We build a simulator $\mathcal{B}$ that can play the decisional BDH with advantage $\epsilon/2$, as follows:

Let G_1 and G_2 be two groups, g a generator of G_1 and e an efficient bilinear map, and the tuples $(A = g^a, B = g^b, C = g^c, Z_1 = g^{abc})$ and $(A = g^a, B = g^b, C = g^c, Z_0 = g^z)$. The challanger flips a coin $p \in 0, 1$ and chosses Z_p. The adversary has to guess Z_p between Z_0 and Z_1.

Init The simulator $\mathcal{B}$ runs the algorithm $\mathcal{A}$, which chooses the set of attributes A for encryption.

Setup $\mathcal{B}$ simulates Setup algorithm of ABE and sets $Y = e(A, B) = e(g, g)^{ab}$. Then, it generates random r_i and sets

$$T_i = \begin{cases} g_i^r, & \text{if } i \in A \\ (g^b)^{r_i}, & \text{otherwise} \end{cases}$$

Then, it outputs the public parameters as:

$$\langle p, G_1, G_2, e, g, nY, T_i\rangle$$

Phase 1 The adversary $\mathcal{A}$ is allowed to issue queries for private keys for many access structures C_j ,such that $C_j(A) = 0$ for all j. $\mathcal{B}$ will use in this scope a procedure called $Fake\_share$, which will simulate theese queries for $\mathcal{A}$.

The definition of $Fake\_share(g^a, \mathcal{C})$ is the following:

1. Initially, all gates of $\mathcal{C}$ are unmarked;

2. Assign g^a to the output wire of the circuit: $Out(\mathcal{C}) = \langle g^a, g^a \rangle$
3. Choose an unmarked gate $\varGamma$ with all input wires defined
4. If $\varGamma$ is an OR-gate, mark $\varGamma$ and assign:
   (a) If $\mathcal{C}(In_1(\varGamma)) = \mathcal{C}(In_2(\varGamma)) = \mathcal{C}(Out(\varGamma))$, then $In_1(\varGamma) = In_2(\varGamma) = Out(\varGamma)$
   (b) If $\mathcal{C}(In_1(\varGamma)) = 1$ and $\mathcal{C}(In_2(\varGamma)) = 0$, then $In_1(\varGamma) = Out(\varGamma)$ and $In_2(\varGamma) = \langle g^{Out(\varGamma,1)}, g^{Out(\varGamma,2)} \rangle$
   (c) If $\mathcal{C}(In_2(\varGamma)) = 1$ and $\mathcal{C}(In_1(\varGamma)) = 0$, then do similar to case (b).
5. If $\varGamma$ is an AND-gate, mark $\varGamma$ and do the following:
   (a) If $\mathcal{C}(In_1(\varGamma)) = \mathcal{C}(In_2(\varGamma)) = \mathcal{C}(Out(\varGamma)) = 1$, then randomly generate $x_1(i)$ and then set $x_2(i)$ such that $Out(\varGamma, i) = x_1(i) + x_2(i)$.
   Then set $In_j(i) = x_j(i)$ for all $i, j \in \{1, 2\}$
   (b) If $\mathcal{C}(In_1) = 1$ and $\mathcal{C}(In_2) = 0$, then:
      i. choose randomly $x_1(i)$ for $i \in \{1, 2\}$.
      ii. compute $g^{x_2(i)} = Out(\varGamma, i)/g^{x_1(i)}$ for $i \in \{1, 2\}$.
      iii. Set $In_1(\varGamma, i) = x_1(i)$ and $In_2(\varGamma, i) = g^{x_2(i)}$.
   (c) If $\mathcal{C}(In_2) = 1$ and $\mathcal{C}(In_1) = 0$: similar to (b) but swaping the two output wires.
   (d) If $\mathcal{C}(In_1) = \mathcal{C}(In_2) = 0$, then:
      i. choose randomly $x_1(i)$ for $i \in \{1, 2\}$.
      ii. compute $g^{x_2(i)} = Out(\varGamma, i)/g^{x_1(i)}$ for $i \in \{1, 2\}$.
      iii. Set $In_1(\varGamma, i) = g^{x_1(i)}$ and $In_2(\varGamma, i) = g^{x_2(i)}$.
6. If $\varGamma$ is an FO-gate, mark $\varGamma$ and then, based on the input wire's evaluation in the Boolean circuit we have two cases, :
   (a) If $\mathcal{C}(In(\varGamma)) = 1$ then:

$$\begin{aligned} In(\varGamma, 1) &= 2Out_1(\varGamma, 1) + Out_2(\varGamma, 1) \\ In(\varGamma, 2) &= Out_1(\varGamma, 2) + Out_2(\varGamma, 2) \\ PP(\varGamma, 1) &= g^{Out_1(\varGamma,1)+Out_1(\varGamma,2)} \\ PP(\varGamma, 2) &= g^{Out_2(\varGamma,1)+Out_2(\varGamma,2)} \end{aligned}$$

   (b) If $\mathcal{C}(In(\varGamma)) = 0$ then:

$$\begin{aligned} In(\varGamma, 1) &= Out_1(\varGamma, 1)^2 \cdot Out_2(\varGamma, 1) \\ In(\varGamma, 2) &= Out_1(\varGamma, 2) \cdot Out_2(\varGamma, 2) \\ PP(\varGamma, 1) &= Out_1(\varGamma, 1) \cdot Out_1(\varGamma, 2) \\ PP(\varGamma, 2) &= Out_2(\varGamma, 1) \cdot Out_2(\varGamma, 2) \end{aligned}$$

7. Repeat steps [3, 4, 5, 6] until all gates are marked.
8. Return $\langle S, PP \rangle$, where $S(\varPsi) = Out(\varPsi)$ for all terminal nodes $\varPsi$.

$\mathcal{B}$ will run $S, PP \rightarrow \mathit{fake\_share}(g^a, \mathcal{C})$ and compute:

$$D(i) = \begin{cases} (g^b)^{S(i,j)/r_i} & \text{if } i \in A \\ S(i,j)^{1/r_i}, & \text{otherwise} \end{cases}$$

Then forward to $\mathcal{A}$:

$$DK = \langle D, PP \rangle$$

From $\mathcal{A}$'s point of view, the shares look as if they were shared using the normal sharing procedure. By using the reconstruct procedure with an approved set of attributes, the *Recon* procedure will return $e(g,g)^{abc}$ if applied to $V(i,j) = e(g,g)^{S(i,j)bc}$ for $i \in A$.

Challenge $\mathcal{A}$ selects two equal length messages m_0 and m_1. The challenger $\mathcal{B}$ flips a random coin b, and encrypts m_b under the set of attributes A and by using Z_p, $p \in \{0,1\}$.

$$E = \langle A, Y = m_b \cdot Z_p, C^{r_i} = g^{r_i c} = T_i^c \rangle$$

If $p = 0$, then $Z_p = e(g,g)^{abc}$ and E is a valid encryption for m_b. Otherwise, Y is a random element from G_2.

Phase 2 Phase 1 is repeated.

Guess The adversary $\mathcal{A}$ outputs a guess b' of b. If $b' = b$, then $\mathcal{B}$ outputs $p = 0$. Otherwise, it outputs $p' = 1$

The advantage of $\mathcal{B}$ is:

$$Adv(\mathcal{B}) = Pr[p' = p] - \frac{1}{2} = Pr[p' = p|p = 0] \cdot Pr[p = 0] + Pr[p' = p|p = 1] \cdot Pr[p = 1] - \frac{1}{2}$$

Both $Pr[p = 0] = \frac{1}{2}$ and $Pr[p = 1] = \frac{1}{2}$

Next, we analyze the two cases:

- If $p = 0$, then $\mathcal{A}$ sees a valid encryption of the ciphertext, thus it's advantage is $Pr[p' = p|p = 0] = \frac{1}{2} + \epsilon$.
- If $p = 1$, then the ciphertext offers no information to $\mathcal{A}$ about the original message, thus in this case $Pr[p' = p|p = 1] = \frac{1}{2}$.

Putting all toghether we obtain:

$$\begin{aligned} Adv(\mathcal{B}) &= Pr[p' = p|p = 0] \cdot Pr[p = 0] + Pr[p' = p|p = 1] \cdot Pr[p = 1] - \frac{1}{2} = \\ &= \frac{1}{2}\left(\frac{1}{2} + \frac{1}{2} + \epsilon\right) - \frac{1}{2} = \\ &= \frac{1}{2}\epsilon \end{aligned}$$

5 Efficiency Analysis

Our KP-ABE scheme results in at most 2 shares for each attribute in the decryption key, regardless of the circuit's structure. However, it has 2 public parameters for each FO-gate in the circuit (Recall that the FO-gates are limited to 2 output wires). The ciphertext has a single value associated to each attribute. For general FO-gates (j-output wires), we can consider having $j-1$ 2-output wire FO-gates, which result in a total number of $2(j-1)$ public parameters.

We have made a comparison between the most efficient schemes for Boolean circuits, for both KP and CP-ABE: [15, 16, 28]. We have omitted in this comparison schemes such as [19], since they target particular cases of Boolean circuits. In the table below, we consider n the maximal number of attributes, r the number of FO-gates, j the number of inputs of the gates.

Another lower bound for our scheme key size could be the number of wires in the circuit: our construction cannot have more public parameters than the total number of wires in the circuit.

Scheme	ABE Type	Key Size	CT size
HuGao_KP[16]	KP	$n + j^r$	n
Tiplea[28]	KP	$nj + n + j^r$	n
Ours	KP	$2n + 2(j-1)r$	n
HuGao_CP[15]	CP	n	$n + j^r$
Ours [3]	CP	n	$2n + 2(j-1)r$

6 Conclusions

We stress that the scheme we have proposed is the best ABE scheme for Boolean circuits using bilinear maps proposed so far, and the first one with polynomial key size (Or linear in the number of wires). This is the first construction that is efficient enough to be used in practice.

Future Directions. Possible improvements could be further made to this system by providing more efficient construction. One possible way of achieving this could be offering a construction for general (unlimited output wires) FO-gates, with less public parameters. Other possible directions include testing compatibility with existing ABE functionalities, such as attribute revocation [3, 26, 31], outsourcing decryption [14, 21].

References

1. Agrawal, S., Yamada, S.: Cp-abe for circuits (and more) in the symmetric key setting. In: Theory of Cryptography Conference. pp. 117–148. Springer (2020)
2. Albrecht, M., Davidson, A.: Are graded encoding scheme broken yet (2017)
3. Attrapadung, N., Imai, H.: Attribute-based encryption supporting direct/indirect revocation modes. In: IMA international conference on cryptography and coding. pp. 278–300. Springer (2009)
4. Attrapadung, N., Imai, H.: Dual-policy attribute based encryption. In: International Conference on Applied Cryptography and Network Security. pp. 168–185. Springer (2009)

[3] Alteration of Hu-Gao's system from [15] using our new *share* and *recon* procedures

5. Attrapadung, N., Libert, B., De Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: International Workshop on Public Key Cryptography. pp. 90–108. Springer (2011)
6. Beimel, A.: Secret-sharing schemes: a survey. In: International conference on coding and cryptology. pp. 11–46. Springer (2011)
7. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE symposium on security and privacy (SP'07). pp. 321–334. IEEE (2007)
8. Drăgan, C.C., Ţiplea, F.L.: 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. pp. 112–133. Springer (2015)
9. ETSI: Cyber; application of attribute based encryption (abe) for pii and personal data protection on iot devices, wlan, cloud and mobile services - high level requirements (2018), https://www.etsi.org/deliver/etsi_ts/103400_103499/103458/01.01.01_60/ts_103458v010101p.pdf
10. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Annual Cryptology Conference. pp. 479–499. Springer (2013)
11. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. Journal of the ACM (JACM) **62**(6), 1–33 (2015)
12. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: International Colloquium on Automata, Languages, and Programming. pp. 579–591. Springer (2008)
13. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on Computer and communications security. pp. 89–98 (2006)
14. Green, M., Hohenberger, S., Waters, B., et al.: Outsourcing the decryption of abe ciphertexts. In: USENIX security symposium. vol. 2011 (2011)
15. Hu, P., Gao, H.: Ciphertext-policy attribute-based encryption for general circuits from bilinear maps. Wuhan University Journal of Natural Sciences **22**(2), 171–177 (2017)
16. Hu, P., Gao, H.: A key-policy attribute-based encryption scheme for general circuit from bilinear maps. IJ Network Security **19**(5), 704–710 (2017)
17. Hu, V.C., Kuhn, D.R., Ferraiolo, D.F., Voas, J.: Attribute-based access control. Computer **48**(2), 85–88 (2015)
18. Huang, D., Dong, Q., Zhu, Y.: Attribute-based Encryption and Access Control. CRC Press (2020)
19. Kowalczyk, L., Wee, H.: Compact adaptively secure abe for nc^1 from k-lin. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 3–33. Springer (2019)
20. Lewko, A., Waters, B.: Unbounded hibe and attribute-based encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 547–567. Springer (2011)
21. Li, J., Sha, F., Zhang, Y., Huang, X., Shen, J.: Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length. Security and Communication Networks **2017** (2017)
22. Li, J., Chen, X., Li, J., Jia, C., Ma, J., Lou, W.: Fine-grained access control system based on outsourced attribute-based encryption. In: European Symposium on Research in Computer Security. pp. 592–609. Springer (2013)

23. Liang, X., Lu, R., Lin, X., Shen, X.S.: Ciphertext policy attribute based encryption with efficient revocation. TechnicalReport, University of Waterloo **2**, 8 (2010)
24. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 457–473. Springer (2005)
25. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Workshop on the theory and application of cryptographic techniques. pp. 47–53. Springer (1984)
26. Shi, Y., Zheng, Q., Liu, J., Han, Z.: Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation. Information Sciences **295**, 221–231 (2015)
27. Steinfeld, R., Sakzad, A., Zhao, R.K.: Practical mp-lwe-based encryption balancing security-risk vs. efficiency. Cryptology ePrint Archive, Report 2019/1179 (2019), https://eprint.iacr.org/2019/1179
28. Ţiplea, F.L., Drăgan, C.C.: Key-policy attribute-based encryption for boolean circuits from bilinear maps. In: International Conference on Cryptography and Information Security in the Balkans. pp. 175–193. Springer (2014)
29. Tiplea, F.L., Dragan, C.C., Nica, A.: Key-policy attribute-based encryption from bilinear maps. In: Farshim, P., Simion, E. (eds.) Innovative Security Solutions for Information Technology and Communications - 10th International Conference, SecITC 2017, Bucharest, Romania, June 8-9, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10543, pp. 28–42. Springer (2017). https://doi.org/10.1007/978-3-319-69284-5_3, https://doi.org/10.1007/978-3-319-69284-5_3
30. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: International Workshop on Public Key Cryptography. pp. 53–70. Springer (2011)
31. Xue, L., Yu, Y., Li, Y., Au, M.H., Du, X., Yang, B.: Efficient attribute-based encryption with attribute revocation for assured data deletion. Information Sciences **479**, 640–650 (2019)
32. Zhong, H., Zhu, W., Xu, Y., Cui, J.: Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage. Soft Computing **22**(1), 243–251 (2018)
33. Ţiplea, F.L.: Multi-linear maps in cryptography. In: Conference on Mathematical Foundations of Informatics. pp. 241–258 (2018)