Attribute-based Encryption for Circuits using Compartmented Access Structures

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.ionita@info.uaic.com
https://profs.info.uaic.ro/~alexandru.ionita

Abstract. Attribute-based encryption (ABE) is an asymmetric encryption method that allows expressive access granting mechanisms, with high applicability in modern IT infrastructure, such as Cloud or IoT systems. [11,25] One open problem regarding ABE is using Boolean circuits as access structures. While Boolean Formulae were supported since the first ABE scheme proposed, there is still no efficient construction that supports Boolean circuits.
We propose a new ABE scheme for a new access structure type, situated between Boolean formulae and Boolean circuits in terms of expressiveness. This key point in our construction is the usage of CAS-nodes, a structure modeling compartmented groups access structures. We also show that our CAS-nodes can be used to improve the efficiency of existing ABE schemes for Boolean circuits. Our construction is secure in the Selective Set Model under the bilinear Decisional Diffie-Hellman Assumption.

Keywords: Attribute-based Encryption · Boolean Circuits · Access Control · Bilinear Maps · Decisional Bilinear Diffie-Hellman.

1 Introduction

In our days, Cloud and IoT services are experiencing continuous growth, and a large amount of data ends up being stored in such systems. Hence, the Cloud service provider has access to sensitive data, such as personal documents or confidential information. For example, suppose we need an online platform for storing personal documents, medical records, and other sensitive data, such that you should be able to download your data on-demand, at any time. We do not wish the Cloud Service provider to have access to such information.

The natural approach to solving this problem is to encrypt all documents containing sensitive information. However, we then face the problem of granting access to these encrypted documents. Using conventional techniques, each user should keep a decryption key for each encrypted document for which he has the right to decrypt. This approach is impractical, due to the large number of decryption keys a user may have.

A modern solution for this problem could be Attribute-based encryption (ABE). This allows us to grant *content-based* or *role-based* access over the encrypted data, depending on which flavor of ABE we are using - key-policy (KP-ABE) or ciphertext-policy (CP-ABE).

Using a KP-ABE system, each user could have set up an instance of the scheme, having a public key and a secret key. With the public key, any doctor could encrypt the respective user's medical data, such as test results or medical examinations. Using the private key, the user is in full control of granting access to its data. Not even the third party which provides the system infrastructure will be able to decrypt these pieces of information, as it will not have access to a user's decryption keys.

Each user can control who has decryption privileges over his data using an access policy defined over some attributes. Thus, we can identify 2 vital features such a system should have:

- expressiveness: to have fine-grained access over the encrypted data, using an expressive access policy
- efficiency: the running time and decryption key size of our scheme should be as low as possible since our data should be accessible even from devices with limited computational power.

Due to the expressiveness it provides, ABE is a subject of high interest in network security. Many researchers are developing secure ABE systems with many functionalities, such as outsourced decryption [17], access revocation [3], or multi-authority ABE [8,27]. Recently, European Telecommunications Standards Institute (ETSI) has published technical specifications for implementing ABE in the cloud, IoT, and other Internet services (ETSI TS 103 458 [10])

One challenge is to construct ABE systems with more expressive access structures while maintaining the computational cost of the scheme in reasonable parameters. Finding the best trade-off between these two properties has been widely studied. There are efficient solutions for Boolean formulae access structures in ABE system [16], but for more complex ones, such as Boolean Circuits [23,18] we only have inefficient solutions.

Our Contribution Starting from the secret sharing for compartmented access structure (CAS) proposed in [13], we propose a new access structure - CAS-circuit. We build an efficient KP-ABE system for this structure, with decryption key linear in the access structure size. Since [24] showed that compartmented access structures cannot be expressed via access trees, our new access structure offers more expressiveness than Boolean formulae. Our scheme is secure in the Selective Set Model for ABE, under the decisional bilinear Diffie-Hellman Assumption.

Moreover, using the CAS-nodes, we can improve the state-of-the-art solutions in terms of ABE for Boolean circuits, namely, [23] for KP-ABE and [18] for

CP-ABE. The CAS-node can be used to re-write a Boolean circuit such that the decryption key will be smaller and the decryption algorithm will run faster. This brings us one step closer to solving the open problem of constructing efficient ABE schemes for Boolean circuits from bilinear maps.

1.1 Related Work

The general idea of Attribute-based encryption was introduced in [21], and the first ABE system was proposed one year later in [16]. In their system, the access policy is represented as an access tree and it is associated with the key (hence the name *Key-Policy* ABE). Their construction is efficient, the key size being linear in the access policy size, and it is proven to be secure in the Selective Set Model under the decisional bilinear Diffie-Hellman (DBDH) Assumption.

Later on, Bethencourt et al. in [6] proposed the first Ciphertext-Policy ABE system. They used the same access structure as in [16], an access tree, but their security was only proven in the generic group model. The first CP-ABE systems proven to be secure under cryptographic assumptions in the standard model were introduced a few years later in [15,26].

ABE and Boolean Circuits Garg et al. [12] proposed the first ABE system with access structures represented by Boolean circuits. They have shown that the sharing technique used for Boolean formulae does not work for Boolean circuits, and proposed a new ABE scheme, based on multi-linear maps. Later, [9] proposed a more efficient system, which relies on a simple form of multi-linear maps, called *chained multi-linear maps*. Since at the moment there is no secure cryptographic construction for any type of multi-linear maps [2,28], these two systems presented above have no practical applicability.

Ţiplea-Dragan [23] proposed the first ABE system for Boolean circuits that relies solely on bilinear maps, which proves to be secure under the bilinear Decisional Diffie-Hellman (BDDH) Assumption. They make use of special fan-out gates (FO-gates) in their Boolean circuit representation, and for each such gate they attach a group element to the decryption key. Hu-Gao [18] refined their result and proposed a similar system, which removes this element associated to the FO-gates. [19] creates a similar system for KP-ABE, by expanding the circuit, resulting in a Boolean access tree, equivalent to the initial circuit. However, all three systems presented above have exponential key size and decryption time in the worst-case scenario.

More progress regarding Boolean circuits ABE schemes was made in the system recently proposed in [20], which offers efficient solutions for NC^1 circuits for both CP and KP ABE systems from bilinear maps. Their system is proven to be secure under the k-Lin Assumption.

In [7] is proposed a solution for ABE for general circuits using bilinear maps. However, this solution's correctness is questionable, due to the secret sharing technique they provide for NAND gates. More precisely, an output wire of such a gate will be divided into a_1 and a_2 such that $a = -a_1 - a_2$. a_1 and a_2 are forwarded to the children nodes of the gate. However, we note that in this case, the reconstruction is possible only if both values from the children are available. This leads to an incorrect construction of the scheme.

Secure ABE schemes for Boolean circuits [14,1] were proposed from Learning With Errors (LWE) assumption. Although LWE offers a strong security guarantee, encryption schemes based on this assumption are often impractical due to the great computing power that they require [22].

2 Preliminaries

Bilinear maps [16] 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.

Decisional Bilinear Diffie-Hellman Assumption Let $a, b, c, z \in \mathbb{Z}_p$ chosen randomly, and g a generator of G_1.

The decisional BDH Assumption [21] is that no polynomial-time algorithm $\mathcal{B}$ can distinguish between $(A = g^a, B = g^b, C = g^c, e(g, g)^{abc})$ and $(A = g^a, B = g^b, C = g^c, e(g, g)^z)$ with a non-negligible advantage.

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

$$|Pr[\mathcal{B}(A, B, C, e(g, g)^{abc})] - Pr[\mathcal{B}(A, B, C, e(g, g)^z)]|$$

where the probability is taken over the random choice of the generator g, the random choice of $a, b, c, z \in \mathbb{Z}_p$, and the random bits consumed by $\mathcal{B}$.

Access Structures [4] 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 formulae and Boolean circuits 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 formula is a Boolean circuit where each node is limited to a fan-out of 1.

KP-ABE Model A Key-Policy Attribute-based encryption scheme, as first described in [16], consists of four algorithms:

setup(λ) A randomized algorithm that takes as input the implicit security parameter λ 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.

Selective-Set Model for KP-ABE [16]

Init The adversary declares the set of attributes $\mathcal{A}$, 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 $\mathcal{A} \notin A_j$ for all j.

Challenge The adversary submits two equal length messages M_0 and M_1. The challenger flips a random coin b, and encrypts M_b with $\mathcal{A}$. 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 $Pr[b' = b] - \frac{1}{2}$.

Notations and abbreviations

Notation	Meaning
$\mathit{\Gamma}$	A node in an access structure
$attr(\mathit{\Gamma})$	attribute corresponding to node $\mathit{\Gamma}$
$In_i(\mathit{\Gamma})$	value of i-th input wire of gate $\mathit{\Gamma}$
$Out(\mathit{\Gamma})$	value of the output wire of gate $\mathit{\Gamma}$

3 Our Construction

While access trees offer a decent level of expressiveness, it has been proven in [24] that they cannot represent compartmented or multi-level access structures. One way of achieving a more refined access control would be by using Boolean circuits instead of access trees. Unfortunately, at the moment there is no efficient and secure construction of ABE systems for them. We can divide the existing schemes into three categories, based on the cryptographic primitives which they are using, and show for each case why it cannot be used in practice:

- ABE for Boolean circuits from LWE [14]: although secure, current LWE-based ABE systems have large public keys, which makes them unpractical.
- ABE for Boolean circuits from multi-linear maps [12,9]: multi-linear maps are very powerful primitives, however, there is still no secure cryptographic solution for implementing them. [2,28]
- ABE for Boolean circuits from bilinear maps: For some Boolean circuits, the current approaches could lead to an exponential expansion of keys or ciphertexts.

The constructions from bilinear maps are the most promising ones at the moment. In order to develop more expressive ABE schemes which can be used in practice, we focus on optimizing existing schemes for Boolean circuits from bilinear maps.

One of the reasons for the great success of Goyal's et al. ABE scheme [16] is that their access tree policy could support more than just Boolean formulae with AND and OR nodes: their construction uses the more expressive threshold nodes (t out of n). Similarly, we propose extending these access structures by adding a new node type: a compartmented access structure - node (further referred to as $CAS-$node).

However, to be able to create even more expressive schemes, we provide construction for secret sharing through a general circuit, by adding sharing techniques for FO-gates. These are special gates introduced in [23], which multiply the output of a node in order to represent Boolean circuits more easily. More details on the efficiency and the improvements added by this new construction can be found in Section 4.

3.1 Compartmented Nodes

A CAS-node Γ will be a special node having 1 output wire (values associated to this wire are stored in $Out(\Gamma)$) and n_Γ input wires. These wires are divided into k disjoint compartments, compartment i having n_i nodes, respecting $n_\Gamma = n_1 + n_2 + \dots n_k$. The i-th compartment input wires are denoted with: $In_{i.1}(\Gamma), In_{i.2}(\Gamma), \dots In_{i.n_i}(\Gamma)$. Each compartment i will have a threshold t_i associated to it: $t_i \leq n_i$, and the CAS-gate will also have a general threshold t, such that $t_1 + t_2 \dots t_k \leq t \leq n_\Gamma$.

A CAS-node Γ is satisfied if and only if:

- each compartment is satisfied (the number of the satisfied input wires in the compartment i exceeds or equals the threshold value t_i)
- the general threshold value is satisfied: The sum of all satisfied input wires of the gate is greater or equal to the general threshold t

Since Ghodosi et al. [13] proposed an ideal secret sharing scheme for Compartmented Groups, we can apply a slightly modified technique to our CAS-node. Thus, the $share\_CAS(y)$ procedure receives a value from $\mathbb{Z}_p$ at the output wire - $Out(\Gamma)$, and assigns a single value to each input wire $In_{i.j}(\Gamma)$, while ensuring that the reconstruction of the initial value is possible if and only if the CAS policy is satisfied.

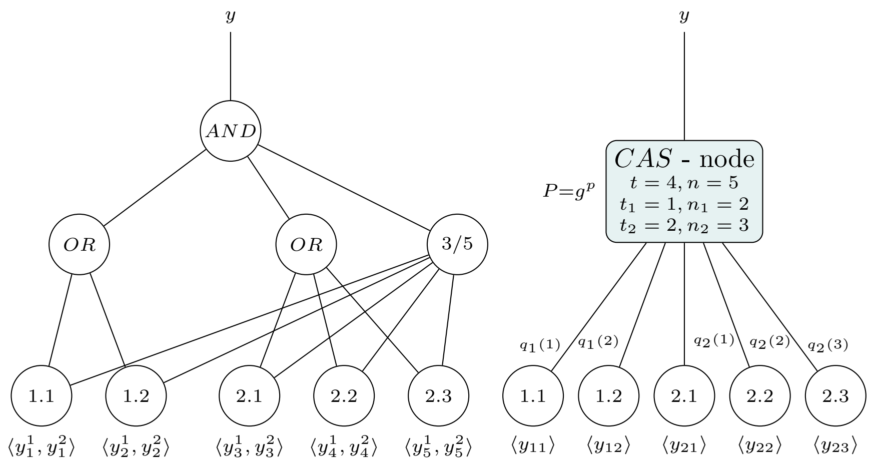

Fig. 1. Replacing a sub-circuit with an equivalent CAS-node. Shares obtained in [23] sharing compared to our scheme

share_CAS(Γ, y):

1. Let $T = t - \sum_{i=1}^{k} t_i$.
2. For each compartment, choose randomly the partial secret y_i and a public parameter p_Γ 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.

Similar to most ABE schemes, in the reconstruction phase, the secret must be reconstructed using elements from G_T. For each value α 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. We further present how the reconstruction should be done:

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 Γ, we will have some value $e(g,g)^{q_i(j)s}$, which represents the result of an equation of form:

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

We need to select from each compartment ℓ_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 the equation system:

$$\begin{cases} g_T^{s(y_1 + a_{1.1} j_{1.1} + a_{1.2} j_{1.1}^2 + \ldots + a_{1.t_1-1} j_{1.1}^{t_1-1} + \beta_1 j_{1.1}^{t_1} + \ldots + \beta_T j_{1.1}^{T+t_1-1})} & = e(g,g)^{q_1(j_{1.1})s} \\ \ldots & \\ g_T^{s(y_1 + a_{1.1} j_{1.\ell_1} + a_{1.2} j_{1.\ell_1}^2 + \ldots + a_{1.t_1-1} j_{1.\ell_1}^{t_1-1} + \beta_1 j_{1.\ell_1}^{t_1} + \ldots + \beta_T j_{1.\ell_1}^{T+t_1-1})} & = e(g,g)^{q_1(j_{1.\ell_1})s} \end{cases}$$

$$\vdots$$

$$\begin{cases} g_T^{s(y_k + a_{k.1} j_{k.1} + a_{k.2} j_{k.1}^2 + \ldots + a_{k.t_k-1} j^{t_k-1} + \beta_1 j_{k.1}^{t_k} + \ldots + \beta_T j_{k.1}^{T+t_k-1})} & = e(g,g)^{q_k(j_{k.1})s} \\ \ldots & \\ g_T^{s(y_k a_{k.1} j_{k.\ell_k} + a_{k.2} j_{k.\ell_k}^2 + \ldots + a_{k.t_k-1} j^{t_k-1} + \beta_1 j_{k.\ell_k}^{t_k} + \ldots + \beta_T j_{k.\ell_k}^{T+t_k-1})} & = e(g,g)^{q_k(j_{k.\ell_k})s} \end{cases}$$

Note that in this system the values y_i, $a_{i.j}$ and β_i are unknown, but we do know all j indices. We have a system of t equations with t unknowns. Therefore, we can compute the partial secrets $g_T^{y_1 s}, \ldots g_T^{y_k s}$. Using these partial secrets, and the gate's public parameter $P(\Gamma)$ we can compute

$$g_T^{ys} = e(g^s, P(\Gamma)) \cdot g_T^{y_1 s} \cdot \ldots \cdot g_T^{y_k s} = g_T^{s(p_\Gamma + y_1 + y_2 + \ldots y_k)}.$$

We can simply observe that the system can be solved if and only if the CAS-node is satisfied.

Sharing and Reconstruction Example. For a better understanding, we provide an example of how these procedures work in Figure 1. First, construct the partial secrets y_1 and y_2 and p such that $y = y_1 + y_2 + p$, and then the polynomials:

$$q_1(x) = y_1 + \beta_1 x$$
$$q_2(x) = y_2 + a_{2.1} x + \beta_1 x^2.$$

Since $T = t - t_1 - t_2 = 1$, we require a single value β_1 common across all compartments. Then, each terminal node will receive an evaluation of the polynomial of his compartment. (Node 1.1 will receive $q_1(1)$, 2.3 will receive $q_2(3)$, etc..)

If at the reconstruction phase we will receive at least one element from each compartment, with a total of at least three elements, we will be able to reconstruct the corresponding values from G_T of the partial secrets y_1 and y_2, namely $g_T^{y_1 s}$ and $g_T^{y_2 s}$, and then compute the corresponding value in G_T for the secret y: $g_T^{ys} = g_T^{y_1 s} g_T^{y_2 s} g_T^{p_\Gamma s}$.

Note that using the CAS-node, we will obtain one share for each input node. For comparison, using the first approach from [24] (a regular Boolean circuit) will result in two shares for each input node.

3.2 Access structure based on CAS-nodes

We can use our newly defined CAS-node to construct expressive access structures with efficient secret sharing. We extend the access structure proposed in [16], namely the access tree, to CAS-nodes, resulting in an access structure that we call a CAS-circuit.

Definition 1. *A CAS-circuit is a tree with CAS-nodes.*

Since threshold gate t/n can be modeled by a CAS-node (by setting the threshold for each compartment to 0, and the general threshold to t), then we can model a Boolean formulae access structure through a CAS-circuit. Although it has visually the form of a tree, we have named our access structure a *circuit*, because Ţiplea et al. proved in [24] that CAS cannot be represented as Boolean formulae, but as Boolean circuits. Therefore, we have that:

Proposition 1. *The class of access structures represented by CAS-circuits is a proper extension of the ones represented by Boolean formulae*

Our construction is very efficient for CAS-circuits, producing only one decryption key element in G_1 for each attribute and one element in G_1 for each access structure internal node. Therefore, we can say that our decryption key has linear size in the access structure size.

3.3 Full construction

We provide a full construction of our scheme for CAS-circuits. For a better understanding, we have separated the secret sharing part from our ABE construction, by defining two special procedures: *share_CAS_circuit* and *recon_CAS_circuit*. They share and reconstruct a secret through a CAS-circuit access structure. The sharing procedure starts with a single value y assigned to the CAS-circuit's output wire and assigns values to all leaf nodes (corresponding to attributes). The reconstruction procedure starts with values in G_T at the bottom of the CAS-circuit in order to reconstruct the value $e(g,g)^{ys}$ required for decryption. We will denote with $In_i(\Gamma)/Out(\Gamma)$ the values associated with the i-th input/output wire of gate Γ. Note that if a gate Γ_2 is the i-th children of another gate Γ_1, then $In_i(\Gamma_2) = Out(\Gamma_1)$.

We start by describing the sharing and reconstruction procedure, and then our full construction of the KP-ABE scheme.

share_CAS_circuit$(y, \mathcal{C})$:

1. Initially, all gates of $\mathcal{C}$ are unmarked;
2. Assign y to the output wire of the circuit: $Out(\mathcal{C}) = y$
3. Choose an unmarked CAS-gate Γ with all input wires defined, and run $\langle \alpha, P(\Gamma) \rangle = share\_CAS(Out(\Gamma))$. This returns a collection of values α, where $\alpha(x.y)$ represents the value of the x-th input node from the y-th compartment. Thus, we set $In_{x.y}(\Gamma) = \alpha(x.y)$. $P(\Gamma)$ is the public parameter which is associated with this gate.
4. Repeat step 3 until all gates are marked.
5. Return $S(\Psi) = Out(\Psi)$ for all terminal nodes Ψ and the public parameters P of CAS-gates.

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

1. Initially, all gates of $\mathcal{C}$ are unmarked;
2. $Out(\Psi) = V_\Psi)$, for all leaf (starting) nodes Ψ. Mark these nodes.
3. Choose an unmarked gate Γ with all input wires defined. We consider that Γ has k input wires, and we do the following: Mark Γ and set $Out(\Gamma) = recon\_CAS(\Gamma, P, g^s, In(\Gamma))$.
4. Repeat step 3 until all gates are marked.
5. return the value from the output wire of the circuit: $Out(\mathcal{C})$.

KP-ABE for *CAS*-circuits scheme :

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

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

and the master key:

$$MSK = \langle y, (t_i, 1 \le i \le n) \rangle$$

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 $s \in \mathbb{Z}_p$, and se the ciphertext as:

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

keygen$(MPK, \mathcal{C})$ The key generation algorithm shares the y component of the MSK through the circuit using the sharing procedure:

$$S, P = share\_CAS\_circuit(y, \mathcal{C}).$$

Then, for every $i \in \mathcal{U}$, output the decryption key as:

$$DK = \langle D_\Psi = g^{S_\Psi / t_i}, P \rangle, \text{ where } \Psi \text{ is a leaf node}$$

decrypt(E, DK) This algorithm receives a valid ciphertext and a decryption key, and returns the original message. For all leaf nodes Ψ, and considering $i = attr(\psi)$, compute:

$$V_\Psi = \begin{cases} e(T_i^s, D_\Psi) = e(g^{t_i s}, g^{S_i/t_i}) = e(g,g)^{S_i s}, & \text{if } i \in A \\ \perp & \text{otherwise} \end{cases}$$

$$R = recon\_CAS\_circuit(\mathcal{C}, V, P)$$

Then compute the message as:

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

4 Efficiency and Improvements

4.1 *CAS*-nodes in other ABE systems

Our CAS-node could be easily combined with other access structures from other KP-ABE schemes. For example, alongside AND and OR nodes, [23] or [19] can be modified to support also CAS-nodes, by simply adding to the scheme the secret sharing and reconstruction procedures provided in our construction.

Similarly, our construction could be applied also to CP-ABE schemes such as [6] or [18] to increase expressiveness or efficiency. We show in the following sections how this node can improve the efficiency of such systems.

4.2 Comparison with other ABE systems

We provide comparisons of our system with existing KP-ABE systems. The relevant schemes are Goyal et al.'s [16], Tiplea-Dragan [23] and Hu-Gao [19]. Regarding the scheme proposed by Goyal et al. [16], we have only compared our system with the access trees version. Their construction for Monotone Span Programs could offer a solution of similar efficiency. However, we do not know how to efficiently convert a CAS-circuit to a Monotone Span Program. Using the construction provided in [5], which transforms any Linear secret sharing scheme to a Monotone Span Program, we obtain an impractically large matrix, with p columns, where p represents the order of our group $\mathbb{Z}_p$.

Besides our basic construction for CAS-circuits, we also added to the comparison an altered version of our scheme, referred to as "Ours-2", as we said that we are able to do in Section 4.1. This scheme supports, besides CAS-nodes, also threshold (from [16]) and FO gates (as in [18] or [23]).

The Ciphertext size and the setup/encryption algorithms are the same in all these schemes. The decryption time in all these schemes is proportional to the decryption key size. Therefore, we will only analyze the key size in these systems.

In our comparison, we have considered that the access structure contains n input nodes, r fan-out gates, j input wires at each fan-out gate, and a total number of q internal gates. The concrete analysis can be seen in Table 4.2. The basic version of our system (Referred to as "Ours-1") can only support CAS-circuits. However, it is the most efficient scheme available at the moment for such access structures. With "Ours-2" we prove that our node is useful in providing efficient secret sharing for CAS-circuit without losing expressiveness or efficiency compared to existing KP-ABE schemes.

Although "Ours-2" does not have better results in the worst-case scenario than the other schemes, we show in Section 4.3 that CAS-gates actually can improve the existing ABE schemes for Boolean circuits in some cases.

Table 1. Worst case decryption key size

ABE system	Bolean Formulae	CAS-circuit	General Boolean circuits
Goyal et al. [16] (using Access trees)	n	Unsupported	Unsupported
Tiplea-Dragan [23]	n	$nj+n+j^r$	$nj+n+j^r$
Hu-Gao [19]	n	$n+j^r$	$n+j^r$
Ours-1 (CAS-circuit)	$n+q$	$n+q$	Unsupported
Ours-2 (general circuit)	n	$n+q$	$n+j^r$

4.3 CAS-nodes in Boolean circuits

While generating decryption keys, one would probably construct a Boolean circuit according to its requirements. Given such an already defined circuit, we explain how it could be optimized using CAS-nodes.

Many sub-circuits can be expressed as CAS-nodes, although at first sight they do not comply to the typical CAS structure. The only requirements that we have for a sub-circuit to be represented as a CAS-node are:

- An AND node Γ_0 at the top of the sub-circuit
- Γ_1 and Γ_2 two children of Γ_0 (Γ_0 may also have other children besides them)
- The set of children of Γ_1 is a proper subset of children of Γ_2.

Such circuits can be expressed as CAS-nodes by creating a *virtual* compartment of threshold 0 which will contain all children of Γ_2 which are not children of Γ_1. This can be seen as a CAS-node with two compartments: the *virtual* one and the one consisting of children of Γ_1.

For such a sub-circuit, the total number of shares are reduced by: $|Out(\Gamma_2)| - |Out(\Gamma_1)+1|$. Note that if this sub-circuit occurs somewhere at the top of a large circuit, this reduction in the number of shares propagates all over the circuit.

We provide such an example of a sub-circuit that can be modeled as a CAS-node in Figure 2. We can consider the children of the OR-node (Γ_1) as the first compartment. Node Z does not have a compartment associated, therefore we will consider it to be in a *virtual* compartment with threshold 0.

For the sub-circuit in Figure 2 (a) the best approach used until now in ABE system for secret sharing over Boolean circuit access policy would result in a

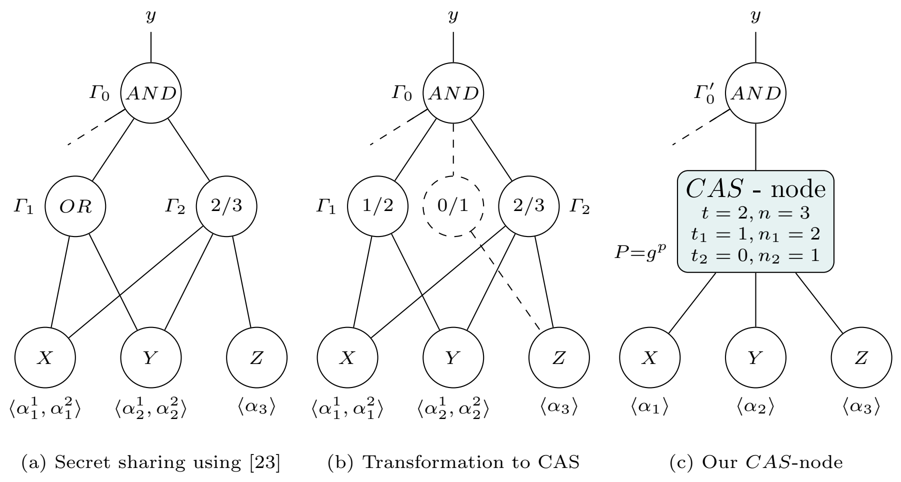

Fig. 2. Replacing a sub-circuit with an equivalent CAS-node. Shares obtained in [23] sharing compared to our scheme

total of 5 shares at the bottom (using the scheme proposed by [23]), while our approach reduces the number of shares to 3.

We observe that sub-circuits which can be expressed as CAS-nodes (as the one Figure 2(a)) are quite common and may occur often in some larger circuits.

Experimental results. Because we could not estimate the benefit of using CAS-nodes in general Boolean circuits, we have tested how much these nodes could reduce the decryption key size in practice. Thus, we have generated various random Boolean circuits with different parameters and then replaced as many sub-circuits as we were able to, with CAS-nodes.

While analyzing a Boolean circuit in order to replace sub-circuits with CAS-nodes, we can observe that some sub-circuits may overlap. Therefore, we must choose only some (non-overlapping) circuits to be replaced. However, the problem of optimally choosing which sub-circuits to use in such cases seems difficult. In our implementation, we have chosen randomly the sub-circuits that will be replaced with CAS-nodes.

Our results can be seen in Table 4.3. The first four columns represent parameters of generated access structure: Number of nodes, number of edges, number of leaf nodes and the height of the Boolean circuit. The last column (denoted with "% Optimized") represents how much the total number of the shares has been reduced. More exactly, it represents p from the formula $S_1 \cdot (1 - \frac{p}{100}) = S_2$, where S_1 denotes the number of share obtained with regular secret sharing, and S_2 represents the number of shares obtained after various sub-circuits being replaced with CAS-nodes.

Table 2. Key size

Nodes	Edges	Leaf nodes	Height	% Optimized
60	160	12	8	12.3%
50	100	12	7	10.24%
50	150	12	10	5.57%
70	200	12	10	21.20%
70	150	12	10	17.26%

5 Security

We provide the full construction of a KP-ABE system using our *share* and *recon* procedures. We stress that these procedures could also be used in CP-ABE systems, such as [6] or [18], to create more efficient CP-ABE systems for Boolean circuits.

Theorem 1. *Our scheme is secure in the selective model under the decisional bilinear Diffie-Hellman assumption.*

Proof. In our security demonstration we will make use of some special procedures, which we will describe beforehand: $PolySat\_CAS$, $PolyUnSat\_CAS$ and $fake\_share\_CAS$. Due to space limitations, we describe these procedures in detail in the Appendix.

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_T 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:

$$\langle p, G_1, G_T, e, g, n, Y, T_i \rangle$$

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: $Out(\mathcal{C}) = g^a$
3. Choose an unmarked gate Γ with all output wires defined, and then run the $fake\_share\_CAS$ algorithm to obtain the values for child nodes and the public values associated to the gate: $In(\Gamma), P(\Gamma) = fake\_share\_CAS(\Gamma)$
4. Repeat step 3 until all gates are marked.
5. Return $\langle S, PP\rangle$, where $S(attr(\Psi)) = Out(\Psi)$ for all terminal nodes Ψ.

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

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

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_T.

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 its 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:

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

5.1 *CAS*-node Public Parameter

Note that the public parameter of the CAS-nodes does not provide additional security in any way. However, it is needed in our security proof. We believe that this parameter is not necessary, but we do not know how to prove the security of our scheme without it.

6 Conclusions

The CAS-node adds a significant improvement to ABE systems for Boolean circuits from bilinear maps, extending the class of practically efficient access structures to a new sub-class of Boolean circuits, the CAS-circuit. Previous constructions of ABE schemes ([16,23]), when adapted to CAS-circuits, are considerably less efficient than our version. This new access structure proves to be a proper extension of the Boolean formulae, lowering the gap between efficient ABE schemes with limited expressiveness (such as [16]) and the expressive ones with high computational demands (such as [18,23]). Our construction is secure in the Selective Set Model under the bilinear decisional Diffie-Hellman Assumption.

Also, as we can see in our practical analysis, the CAS-node can be used to optimize pre-defined circuits, by offering a more efficient secret sharing method. This could be only a first step towards constructing more expressive and efficient ABE systems. Finding other structures with efficient secret sharing techniques may further enlarge the class of practically efficient access structures. A new possible direction could be in finding such access structures and constructing ABE systems for them. This could, in the end, could lead us to an efficient ABE construction for Boolean circuits.

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. Beimel, A.: Secret-sharing schemes: a survey. In: International conference on coding and cryptology. pp. 11–46. Springer (2011)
5. Beimel, A., et al.: Secure schemes for secret sharing and key distribution (1996)
6. 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)
7. Bolocan, D.: Key-policy attribute-based encryption scheme for general circuits. PROCEEDINGS OF THE ROMANIAN ACADEMY SERIES A-MATHEMATICS PHYSICS TECHNICAL SCIENCES INFORMATION SCIENCE **21**(1), 11–19 (2020)
8. Chase, M.: Multi-authority attribute based encryption. In: Theory of Cryptography Conference. pp. 515–534. Springer (2007)
9. 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)
10. 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
11. Ezhilarasi, T., Sudheer Kumar, N., Latchoumi, T., Balayesu, N.: A secure data sharing using idss cp-abe in cloud storage. In: Advances in Industrial Automation and Smart Manufacturing, pp. 1073–1085. Springer (2021)
12. 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)
13. Ghodosi, H., Pieprzyk, J., Safavi-Naini, R.: Secret sharing in multilevel and compartmented groups. In: Australasian Conference on Information Security and Privacy. pp. 367–378. Springer (1998)
14. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. Journal of the ACM (JACM) **62**(6), 1–33 (2015)
15. 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)
16. 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)
17. Green, M., Hohenberger, S., Waters, B., et al.: Outsourcing the decryption of abe ciphertexts. In: USENIX security symposium. vol. 2011 (2011)
18. 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)
19. 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)
20. 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)

21. 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)
22. 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`
23. 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)
24. Tiplea, F.L., Ionita, A., Nica, A.: Practically efficient attribute-based encryption for compartmented access structures. In: Samarati, P., di Vimercati, S.D.C., Obaidat, M.S., Ben-Othman, J. (eds.) Proceedings of the 17th International Joint Conference on e-Business and Telecommunications, ICETE 2020 - Volume 2: SECRYPT, Lieusaint, Paris, France, July 8-10, 2020. pp. 201–212. ScitePress (2020). https://doi.org/10.5220/0009887202010212, `https://doi.org/10.5220/0009887202010212`
25. Touati, L., Challal, Y.: Collaborative kp-abe for cloud-based internet of things applications. In: 2016 IEEE International Conference on Communications (ICC). pp. 1–7. IEEE (2016)
26. 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)
27. 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)
28. Ţiplea, F.L.: Multi-linear maps in cryptography. In: Conference on Mathematical Foundations of Informatics. pp. 241–258 (2018)

Appendix

Procedures used in security demonstration:

$\underline{PolySat\_CAS(y_1, t_i, k_i, \beta)}$ receives as input a value $y_i \in \mathbb{Z}_p$ and integers t_i and k_i representing threshold and number of component of some compartment in a CAS-node. $\beta = \beta_1, \dots \beta_T$ represent the additional values, common in all compartments to be used as polynomial coefficients. Knowing y, we simply construct a polynomial q' of grade $t_i - 1$ by randomly choosing its coefficients with y_i as free term. Then construct the polynomial $q(x) = q'(x) + \beta_1 x^{t_i} + \dots + \beta_T^{t_i+T-1}$.

For every satisfied child j set $q(j)$ as the value associated to that wire, and for every unsatisfied wire j set the value $g^{q(j)}$.

$\underline{PolyUnSat\_CAS(y_i, t_i, k_i, \beta')}$: We stress that this function will receive the β coefficients in G_1: $\beta' = \langle g^{\beta_1}, g^{\beta_2}, \dots g^{\beta_T} \rangle$

- Let ℓ be the number of satisfied input wires, and the wires: $j_1, j_2, ..., j_\ell$, $\ell < t_i$
- Choose randomly $x_{j_1}, x_{j_2}, ...x_{j_\ell}$ and assign them to the satisfied input wires
- Choose randomly $\ell' = t_i - \ell - 1$ more values $x_{j_{\ell+1}} \dots x_{t_i - 1}$ to completely define the polynomial q, such that $q(i) = x_i$.

- Then, the coefficients a_i of this polynomial can be computed in G_1:

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

- Evaluate this polynomial in some points to obtain the rest of the values. Since β' contains values from G_1 we can only compute the G_1 value of the evaluation of q in some point, but this is enough, since these values will be passed anyway to unsatisfied wires.

$\underline{fake\_share\_CAS(\Gamma)}$:
We consider that Γ is a CAS-gate with k compartments with general threshold t, and each one of the compartments has k_i elements and threshold t_i, where $i \in \overline{1,k}$:

- Set $T = t - t_1 - t_2 - \ldots - t_k$ the difference between the general and the sum of partial thresholds.
- if the gate Γ is satisfied, then simply return $share\_CAS(Out(\Gamma))$. (the secret sharing technique used in our scheme).
- if the gate Γ is not satisfied, then $Out(\Gamma) = g^y$ is a value from G_1.
  Generate $y_1, y_2, \ldots y_k$ and compute $P(\Gamma)$ from G_1 such that:

$$P(\Gamma) \cdot g^{\sum_{j=1}^{k} y_j} = Out(\Gamma).$$

  Then randomly generate parameters $\beta_1, \ldots \beta_T$ from $\mathbb{Z}_p$, and for each compartment i choose values for its input wires $In_{i.j}(\Gamma), j \in \overline{1, n_1}$, by using $PolySat\_CAS$, or $PolyUnsat\_CAS$, depending if the respective compartement is satisfied or not. We say that a compartment i is satisfied if its number of satisfied input wires is strictly greater than the threshold of the compartment.
- Return $In(\Gamma)$ and public gate parameter $P(\Gamma)$.