Attribute-Based Signatures for Circuits from Bilinear Map

Yusuke Sakai[⋆], Nuttapong Attrapadung, and Goichiro Hanaoka

AIST, Japan
{yusuke.sakai,n.attrapadung,hanaoka-goichiro}@aist.go.jp

Abstract. In attribute-based signatures, each signer receives a signing key from the authority, which is associated with the signer's attribute, and using the signing key, the signer can issue a signature on any message under a predicate, if his attribute satisfies the predicate. One of the ultimate goals in this area is to support a wide class of predicates, such as the class of *arbitrary circuits*, with *practical efficiency* from *a simple assumption*, since these three aspects determine the usefulness of the scheme. We present an attribute-based signature scheme which allows us to use an arbitrary circuit as the predicate with practical efficiency from the symmetric external Diffie-Hellman assumption. We achieve this by combining the efficiency of Groth-Sahai proofs, which allow us to prove algebraic equations efficiently, and the expressiveness of Groth-Ostrovsky-Sahai proofs, which allow us to prove any NP relation via circuit satisfiability.

Keywords: attribute-based signatures, Groth-Sahai proofs, Groth-Ostrovsky-Sahai proofs

1 Introduction

1.1 Attribute-Based Signatures

In an ordinary digital signature scheme, a signer has a signing key and publicizes its corresponding verification key. The verification is performed with respect to such a public key, and hence during the verification process, those who made the signature is uniquely determined. In other words, digital signatures provide nothing for privacy or anonymity requirements.

The concept of attribute-based signatures is introduced by Maji, Prabhakaran, and Rosulek [21], in order to relax this firm correspondence between a signer and a signature. In an attribute-based signature scheme, there is an attribute authority, and each signer receives from the authority a signing key associated with his attribute. Once a signer receives a signing key, he is able to issue a signature on any message, under a predicate satisfied by his attribute. The signature is *anonymous*, that is, the signature tells a verifier that the party

[⋆] The first author is supported by a JSPS Fellowship for Young Scientists.

who generates the signature has an attribute satisfying the predicate, but further information on the signer's identity or attribute is completely hidden from the verifier.

One of the active lines of research on attribute-based signatures is to support a larger class of predicates with practical efficiency. The state-of-the-art results along this line is the scheme by Okamoto and Takashima for non-monotone span programs from bilinear groups [24] and the scheme by Tang, Li, and Liang for any circuits from multilinear maps [27]. The ultimate goal in this line is achieving a large class of predicate, such as *the class of arbitrary circuits*, while keeping the scheme *practically efficient* and relying on *a simple assumption*, since these three aspects determine the usefulness of the scheme in practice. However, neither of above two schemes and in fact neither of any existing scheme does not achieve this ultimate goal.

Bellare and Fuchsbauer proposed a versatile cryptographic primitive called policy-based signatures [2]. They showed a generic construction of an attribute-based signature scheme from a policy-based signature scheme. There are two ways of instantiating their generic construction. Namely, the one is an instantiation with NIZK for general NP languages such as the Groth-Ostrovsky-Sahai proof system [13], and the other is an instantiation with NIZK for specific algebraic equations such as the Groth-Sahai proof system [14]. Although the authors of [2] did not explicitly mention (they only dealt with monotone predicates), the former may be extended to support the class of arbitrary circuits. However, it suffers from a large overhead of the signature size due to a Karp reduction to an NP-complete problem. The latter can be instantiated efficiently, but the supported class is restricted to conjunctions and disjunctions of pairing-product equations.

In summary, it still remains open whether it is possible to construct an attribute-based signature scheme that supports circuit predicates with practical efficiency from simple assumptions.

1.2 Efficient Non-interactive Zero-knowledge

In this section we review non-interactive zero-knowledge (NIZK) proofs, which can be useful building blocks for constructing attribute-based signatures.

NIZK proofs allow us to prove that a secret information satisfies a public condition without revealing the secret beyond the truth of the condition. This primitive is extremely useful and widely studied in the area of cryptography. It has been an important research topic to expand the class of the predicate that proof systems support, as well as to improve the efficiency of proof systems.

Recent developments in zero-knowledge proofs include the proof system by Groth, Ostrovsky, and Sahai [13] and the one by Groth and Sahai [14]. The former can prove any NP relation via circuit satisfiability, but it suffers from large overhead due to a Karp reduction. The latter is very efficient, but the class of the relation is restricted to algebraic equations, and hence it cannot treat arbitrary NP relation in general.

A natural question is whether it is possible to construct a proof system which is as expressive as the Groth-Ostrovsky-Sahai proof system, and is at the same time as efficient as the Groth-Sahai proof system. In this paper, we investigate a case study of a fusion of Groth-Ostrovsky-Sahai and Groth-Sahai proofs in case of attribute-based signatures, and show that by this idea, we can construct a practical attribute-based signature for circuits from bilinear maps.

1.3 Our Contribution

In this paper, we present an attribute-based signature scheme for arbitrary circuits of unbounded size and depth with practical efficiency, from a simple assumption over bilinear groups. Our attribute-based signature scheme satisfies perfect privacy and adaptive unforgeability. The scheme is based on a witness indistinguishable and extractable non-interactive proof system and an existentially unforgeable signature scheme. All the building blocks can be instantiated solely from the symmetric external Diffie-Hellman (SXDH) assumption [14, 16], and thus we can obtain a perfectly private and adaptively unforgeable scheme from the same assumption.

Our scheme is fairly practical. The signature size grows as around one kilobyte per each gate, which is comparable to the existing schemes such as the schemes by Maji et al. [21] and the scheme by Okamoto and Takashima [24]. We note that Maji et al.'s schemes and the Okamoto-Takashima scheme are less expressive than ours, namely, Maji et al.'s schemes support monotone span programs, while the Okamoto-Takashima scheme supports non-monotone span programs. In addition, our scheme drastically improves efficiency when we compare it with related schemes of Bellare and Fuchsbauer [2] and Tang, Li, and Liang [27]. As stated above, the former scheme is a generic construction of attribute-based signatures from policy-based signatures and the latter scheme is an attribute-based signature scheme for circuits from multilinear maps.

It would be interesting to note the contrast between our scheme and its encryption counterparts, namely, the attribute-based encryption schemes for circuits [11, 9, 12]. We highlight that our scheme only requires a simple and popular bilinear map assumption, namely the SXDH assumption to prove its security, whereas the encryption counterparts require powerful lattice assumptions or multilinear maps. This is reminiscent of the fact that an identity-based signature scheme can be constructed only from a standard digital signature scheme [22, 17, 3], while identity-based encryption requires a very strong assumption [5].

1.4 Technique

The basic idea behind our construction is simple: to sign anonymously, a signer receives a signature on his attribute from the authority, and proves the knowledge of this signature together with a proof that shows the signed attribute satisfies a public circuit. The signature that the signer receives works as a certificate, which certifies the signer having the attribute, and forbids the third party from signing in the name of his attribute.

To implement this idea, we need to overcome two difficulties. The first difficulty is (1) *simultaneously and efficiently proving circuit satisfiability of the attribute and the validity of the certificate on that attribute.* The other difficulty is (2) *binding the proof from the first part to a message to be signed.* In the following we give more detailed explanations on these difficulties and our idea for overcoming them.

(1) Proving circuit satisfiability and certificate validity. The first difficulty is expressing circuit satisfiability of an attribute in zero-knowledge, while keeping the entire proof system efficient. We need to prove not only circuit satisfiability of an attribute, but also validity of a certificate. The Groth-Ostrovsky-Sahai proof system enables us to prove circuit satisfiability, but its direct use does not allow us to prove the validity of the certificate *efficiently*, since, if we were to use the Groth-Ostrovsky-Sahai proof system, we must represent validity of a certificate in a circuit via a Karp reduction, which is highly inefficient.

Nevertheless, our starting point is still the technique of Groth, Ostrovsky, and Sahai [13]. In this technique, to prove circuit satisfiability, the prover first computes commitments to assignments to each wire, and then proves that for each gate the incoming wires u and v and the outgoing wire w satisfy the NAND relation $\neg(u \wedge v) = w$.

We instantiate this idea with Groth-Sahai proofs. We need Groth-Sahai proofs, rather than a simple adoption of the Groth-Ostrovsky-Sahai proof system because we need to handle not only Boolean relations (for the NAND gates as above), but also algebraic equations at the same time. The need for algebraic equations comes from the necessity to certifying attributes. As stated above, the authority signs on attributes to certify that each signer can sign in the name of his attribute. Hence we need to prove the validity of the certificate, and for this purpose we employ Groth-Sahai proofs, together with structure-preserving signatures [1].

Therefore, we need to translate the idea of the Groth-Ostrovsky-Sahai proof system into the Groth-Sahai proof system. Namely, we need to translate the NAND relation $\neg(u \wedge v) = w$ into a bilinear equation, which is what the Groth-Sahai proofs can prove. We do this by *arithmetizing* the relation. That is, let u and v be the assignments to incoming wires then w be the assignment to the outgoing wire, and the prover proves the equation $1 - u \cdot v = w$ to prove the NAND relation.

(2) Binding the proof to a message. The other difficulty is binding the proof to a single message in order to resist chosen-message attacks. Although we want to prove knowledge of certificates to sign anonymously, this dose not suffice for resisting chosen-message attacks. This is because the proof is not bound to the message, and hence the adversary can reuse the signature (the proof) on some message to a signature on another message.

To overcome this difficulty, we introduce an OR-proof technique, following Maji et al. [21]. In this technique, the signer proves the knowledge of the certificate *or* a signature on a dummy attribute, which is an extra attribute unused in the real protocol, and differs message by message.

The point is that different messages have different dummy attributes. To be more specific, an (attribute-based) signature on message M proves the knowledge of a signature on some attribute *or* a signature on a dummy attribute x, while a signature on a different message M^* proves the knowledge of a signature on an attribute *or* a signature on another dummy attribute x^*. By this means, if an adversary sees a signature on M and forges a signature on M^*, then a reduction extracts a witness from the forgery and obtains a signature on x^* of the underlying signature scheme. With this x^* the reduction reduces the forgery for the attribute-based signature scheme to a forgery for the underlying signature scheme.

1.5 Related Work

Maji et al. [20, 21] introduced the notion of attribute-based signatures, and presented three constructions which have perfect privacy and adaptive unforgeability. The first two schemes combine a digital signature scheme and Groth-Sahai proofs. These two schemes are instantiated respectively with the Boneh-Boyen signature scheme [4] and with the Waters signature scheme [29]. The third construction is proven secure in the generic group model. Following Maji et al.'s results, Li and Kim [19], Siamak and Safavi-Naini [26], and Li et al. [18] presented attribute-based signature schemes, which are proven secure only in the selective model of unforgeability. Another drawback of these schemes is relatively narrow class of the supported predicates. Namely Li and Kim's scheme [19] only supports conjunction predicates, while Siamak and Safavi-Naini's scheme [26] and Li et al.'s scheme [18] support threshold predicates. Escala, Herranz, and Morillo presented an attribute-based signature with adaptive unforgeability [8]. Okamoto and Takashima presented an attribute-based signature scheme which is adaptively unforgeable and supports non-monotone span programs as predicates [24]. Recently, Herranz et al. [15], followed by Chen et al. [6], presented attribute-based signature schemes with constant-size signatures for threshold predicates. The former has selective unforgeability while the latter has adaptive unforgeability. Wang and Chen [28] presented an attribute-based signature scheme from a lattice assumption with selective unforgeability. Tang, Li, and Liang [27] presented an attribute-based signature scheme for bounded-depth circuits from multilinear maps. Most recently, Mridul and Pandit presented various attribute-based signature schemes such as for Boolean formulas or for regular languages from q-type assumptions [23].

Escala, Herranz, and Morillo presented a traceable attribute-based signature scheme (under the name of "revocable" attribute-based signatures) [8], which allows a trusted authority to identify who made a signatures. Okamoto and Takashima presented a decentralized attribute-based signature scheme [25], which removes the necessity of any trusted setup in the system. Following these works, El Kaafarani, Ghadafi, and Khader presented a decentralized traceable attribute-based signature scheme [7]. Ghadafi revisited the security notion of decentralized traceable attribute-based signatures, and introduced, among other things, a new security notion of non-frameability [10].

As for attribute-based encryption for circuits, Gorbunov, Vaikuntanathan, and Wee [11] presented the first attribute-based encryption scheme for circuits. After that, Garg et al. [9] presented an attribute-based encryption scheme for circuits from multilinear maps. Recently, Gorbunov, Vaikuntanathan, and Wee presented a predicate encryption scheme for circuits from a class of learning-with-errors assumptions [12].

2 Preliminary

We say that a function $f\colon \mathbb{N} \to \mathbb{R}$ is negligible if for all $c \in \mathbb{N}$ there exists $x_0 \in \mathbb{N}$ such that $f(x) \leq x^{-c}$ for all $x \geq x_0$.

Representation of circuit. Here we explain notation for circuits, especially how we identify a circuit. Let C be a circuit with L-bit input and N gates. We assume C is entirely represented by NAND gates. We distinguish the input wires, the internal wires, and the output wire by indices 1, $\ldots$, L, $L+1$, $\ldots$, $L+N$, where 1, $\ldots$, L are the input wires, $L+1$, $\ldots$, $L+N-1$ are the internal wires, and $L+N$ is the output wire. The topology of the circuit is specified by two functions I_1, $I_2\colon \{L+1, \ldots, L+N\} \to \{1, \ldots, L+N-1\}$. They map a non-input wire to its first and second incoming wires in which these three wires are connected by a NAND gate. We require that $I_1(i) < i$ and $I_2(i) < i$.

Bilinear groups. Let $\mathcal{G}$ be a probabilistic polynomial-time algorithm that on input 1^k outputs a group description $\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$ where p is a prime, $\mathbb{G}_1$ and $\mathbb{G}_2$ are multiplicative groups generated by g and $\tilde{g}$, respectively, $\mathbb{G}_T$ is a multiplicative group of order p, and $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate efficiently computable bilinear map.

We say that the decision Diffie-Hellman assumption on $\mathbb{G}_1$ holds if for any probabilistic polynomial-time adversary $\mathcal{A}$

$$\begin{aligned}&|\Pr[\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, e, g, \tilde{g}) \leftarrow \mathcal{G}(1^k); x, y \leftarrow \mathbb{Z}_p : \mathcal{A}(\mathsf{gk}, g^x, g^y, g^{xy}) = 1] \\ &\quad - \Pr[\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, e, g, \tilde{g}) \leftarrow \mathcal{G}(1^k); x, y, z \leftarrow \mathbb{Z}_p : \mathcal{A}(\mathsf{gk}, g^x, g^y, g^z) = 1]|\end{aligned}$$

is negligible. The decision Diffie-Hellman assumption on $\mathbb{G}_2$ is defined similarly. Namely, we say the decision Diffie-Hellman assumption holds on $\mathbb{G}_2$ if

$$\begin{aligned}&|\Pr[\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, e, g, \tilde{g}) \leftarrow \mathcal{G}(1^k); x, y \leftarrow \mathbb{Z}_p : \mathcal{A}(\mathsf{gk}, \tilde{g}^x, \tilde{g}^y, \tilde{g}^{xy}) = 1] \\ &\quad - \Pr[\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, e, g, \tilde{g}) \leftarrow \mathcal{G}(1^k); x, y, z \leftarrow \mathbb{Z}_p : \mathcal{A}(\mathsf{gk}, \tilde{g}^x, \tilde{g}^y, \tilde{g}^z) = 1]|\end{aligned}$$

is negligible. We say that the symmetric external Diffie-Hellman (SXDH) assumption holds if the decision Diffie-Hellman assumptions on both $\mathbb{G}_1$ and $\mathbb{G}_2$ hold.

Groth-Sahai and Groth-Ostrovsky-Sahai proofs. A non-interactive proof system for the NP relation $R \subset \{0,1\}^* \times \{0,1\}^*$ is defined by following three algorithms $(\mathsf{WISetup}, \mathsf{WIProve}, \mathsf{WIVerify})$: the setup algorithm $\mathsf{WISetup}$ takes as input the security parameter 1^k and outputs a common reference string crs; the proof algorithm takes as input the common reference string crs, a statement x, and a witness w, and outputs a proof π; the verification algorithm $\mathsf{WIVerify}$ takes as input the common reference string crs, the statement x, and the proof π, and outputs 1 or 0 which indicate validity of the proof. As a correctness condition, we require that for all $k \in \mathbb{N}$, $(x,w) \in R$, and $\mathsf{crs} \leftarrow \mathsf{WISetup}(1^k)$, it holds that $\mathsf{WIVerify}(\mathsf{crs}, x, \mathsf{WIProve}(\mathsf{crs}, x, w)) = 1$.

We require a proof system to be perfectly witness indistinguishable (WI) and perfectly extractable. A proof system is perfectly witness indistinguishable if for any $\mathsf{crs} \leftarrow \mathsf{WISetup}(1^k)$, $x \in \{0,1\}^*$, $w_0 \in \{0,1\}^*$, $w_1 \in \{0,1\}^*$ such that (x, w_0), $(x, w_1) \in R$, the two distributions $\mathsf{WIProve}(\mathsf{crs}, x, w_0)$ and $\mathsf{WIProve}(\mathsf{crs}, x, w_1)$ distributes identically. The proof system is perfectly extractable if there are two algorithms $\mathsf{ExtSetup}$ and $\mathsf{Extract}$ that satisfy the following two properties: (1) for any probabilistic polynomial-time adversary $\mathcal{A}$,

$$|\Pr[\mathsf{crs} \leftarrow \mathsf{WISetup}(1^k) : \mathcal{A}(\mathsf{crs}) = 1] \\ - \Pr[(\mathsf{crs}, \mathsf{ek}) \leftarrow \mathsf{ExtSetup}(1^k) : \mathcal{A}(\mathsf{crs}) = 1]|$$

is negligible, and (2) for any probabilistic polynomial-time adversary $\mathcal{A}$,

$$\Pr[(\mathsf{crs}, \mathsf{ek}) \leftarrow \mathsf{ExtSetup}(1^k); (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}); \\ w \leftarrow \mathsf{Extract}(\mathsf{crs}, \mathsf{ek}, x, \pi) : \mathsf{WIVerify}(\mathsf{crs}, x, \pi) = 1 \text{ and } (x, w) \notin R] = 0.$$

The Groth-Sahai proof system [14] is a proof system which can prove satisfiability of a set of algebraic equations called pairing-product equations in a witness-indistinguishable and extractable manner under the SXDH assumption. In particular the Groth-Sahai proof system can prove satisfiability of a set of pairing-product equations, which are the equation of the form

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{Y}_i) \prod_{j=1}^{m} e(\mathcal{X}_j, \mathcal{B}_j) \prod_{i=1}^{n} \prod_{j=1}^{m} e(\mathcal{X}_i, \mathcal{Y}_j)^{\gamma_{i,j}} = T$$

in which $\mathcal{A}_i \in \mathbb{G}_1$, $\mathcal{B}_j \in \mathbb{G}_2$, $\gamma_{i,j} \in \mathbb{Z}_p$, and $T \in \mathbb{G}_T$ are public constants, and $\mathcal{X}_i \in \mathbb{G}_1$ and $\mathcal{Y}_j \in \mathbb{G}_2$ are private variables (witness). To prove the knowledge of a satisfying assignment, the prover first computes commitments to each witness (we call this the Groth-Sahai commitment), and then computes proofs demonstrating the witness satisfies the equations. The commitment consists of the two group elements in the same group as the witness. For proving a pairing-product equation, Groth-Sahai proofs require eight group elements, in particular four elements in $\mathbb{G}_1$ and four elements in $\mathbb{G}_2$, for each equation. In the case that $n = 0$ and thus the equation to be proved has the form

$$\prod_{j=1}^{m} e(\mathcal{X}_j, \mathcal{B}_j) = T,$$

it only requires two group elements in $\mathbb{G}_2$. See [14] for further detail.

Groth-Ostrovsky-Sahai proofs are the proof system which can prove satisfiability of a circuit which solely consists of NAND gates. The proof algorithm proceeds with a similar way to the Groth-Sahai proofs. Namely, the prover first computes commitments to the assignments to the wires, and then proves each triple (u, v, w) of wires connected by a NAND gate satisfies the NAND relation $\neg(u \wedge v) = w$. See [13] for further detail.

Structure-preserving signatures. A signature scheme consists of the following three algorithms $(\mathsf{Kg}, \mathsf{Sign}, \mathsf{Verify})$: the key generation algorithm takes as input a security parameter 1^k and outputs a pair $(\mathsf{vk}, \mathsf{sk})$ of the verification key and the signing key; the signing algorithm Sign takes as input the signing key sk and a message m and outputs a signature θ; the verification algorithm Verify takes as input the verification key vk, the message m, and the signature θ, and outputs 1 or 0 indicating validity of the signature. As the correctness condition, it is required to hold that for all $k \in \mathbb{N}$, $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Kg}(1^k)$, and $m \in \{0,1\}^*$, it $\mathsf{Verify}(\mathsf{vk}, m, \mathsf{Sign}(\mathsf{sk}, m)) = 1$.

A signature scheme $(\mathsf{Kg}, \mathsf{Sign}, \mathsf{Verify})$ is said to be existentially unforgeable, if the probability $\Pr[(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Kg}(1^k); (m^*, \theta^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{vk}) : \mathsf{Verify}(\mathsf{vk}, m^*, \theta^*) = 1 \wedge m^* \text{ is not queried}]$ is negligible for all probabilistic polynomial-time adversaries $\mathcal{A}$.

Our scheme can be instantiated using any structure-preserving signature scheme. For concreteness, we employ the recent scheme by Kiltz, Pan, and Wee (KPW) [16], which is efficient and based on the SXDH assumption. For completeness, we describe the KPW signature scheme below. In the description, for a matrix $A = (a_{i,j}) \in {\mathbb{Z}_p}^{n \times m}$ we denote by $[A]_1$

$$[A]_1 = \begin{pmatrix} g^{a_{1,1}} & \cdots & g^{a_{1,m}} \\ \vdots & \ddots & \vdots \\ g^{a_{n,1}} & \cdots & g^{a_{n,m}} \end{pmatrix} \in {\mathbb{G}_1}^{n \times m},$$

and similarly for $[A]_2 \in {\mathbb{G}_2}^{n \times m}$ with generator $\tilde{g}$, and $[A]_T$ with generator $e(g, \tilde{g})$. For two matrices A and B, we denote $e([A]_1, [B]_2) = [AB]_T$.

$\mathsf{Kg}(\mathsf{gk}, 1^L)$**.** Given a description $\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$ of bilinear groups and a message length L, choose a, $b \leftarrow \mathbb{Z}_p$, $K \leftarrow {\mathbb{Z}_p}^{(L+1) \times 2}$, let $A = (1|a)^\top \in {\mathbb{Z}_p}^{2 \times 1}$, $B = (1|b)^\top \in {\mathbb{Z}_p}^{2 \times 1}$, choose K_0, $K_1 \leftarrow {\mathbb{Z}_p}^{2 \times 2}$, let $C \leftarrow KA$, $C_0 \leftarrow K_0 A$, $C_1 \leftarrow K_1 A$, $P_0 \leftarrow B^\top K_0$, $P_1 \leftarrow B^\top K_1$. Let $\mathsf{vk}_{\mathsf{Sign}} \leftarrow ([C_0]_2, [C_1]_2, [C]_2, [A]_2)$ and $\mathsf{sk}_{\mathsf{Sign}} \leftarrow (\mathsf{vk}_{\mathsf{Sign}}, K, [P_0]_1, [P_1]_1, [B]_1)$, and output $(\mathsf{vk}_{\mathsf{Sign}}, \mathsf{sk}_{\mathsf{Sign}})$.

$\mathsf{Sign}(\mathsf{sk}_{\mathsf{Sign}}, [\boldsymbol{m}]_1)$**.** Given a signing key $\mathsf{sk}_{\mathsf{Sign}} \leftarrow (\mathsf{vk}_{\mathsf{Sign}}, K, [P_0]_1, [P_1]_1, [B]_1)$ and a message $[\boldsymbol{m}]_1 \in {\mathbb{G}_1}^L$, choose $\boldsymbol{r} \leftarrow {\mathbb{Z}_p}^2$ and $\tau \leftarrow \mathbb{Z}_p$, compute

$$\begin{aligned}
\theta_1 &\leftarrow [(1|\boldsymbol{m}^\top)K + \boldsymbol{r}^\top(P_0 + \tau P_1)]_1 \in {\mathbb{G}_1}^{1\times 2},\\
\theta_2 &\leftarrow [\boldsymbol{r}^\top B^\top]_1 \in {\mathbb{G}_1}^{1\times 2},\\
\theta_3 &\leftarrow [\boldsymbol{r}^\top B^\top \tau]_1 \in {\mathbb{G}_1}^{1\times 2},\\
\theta_4 &\leftarrow [\tau]_2 \in \mathbb{G}_2.
\end{aligned}$$

Let $\theta \leftarrow (\theta_1, \theta_2, \theta_3, \theta_4)$ and output θ.

$\mathsf{Verify}(\mathsf{vk}_{\mathsf{Sign}}, \theta)$**.** Given the verification key $\mathsf{vk}_{\mathsf{Sign}} = ([C_0]_2, [C_1]_2, [C]_2, [A]_2)$, a message $[\boldsymbol{m}]_1 \in {\mathbb{G}_1}^L$, and a signature $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$, check

$$\begin{aligned}
e(\theta_1, [A]_2) &= e([(1|\boldsymbol{m})]_1, [C]_2) e(\theta_2, [C_0]_2) e(\theta_3, [C_1]_2),\\
e(\theta_2, \theta_4) &= e(\theta_3, [1]_2).
\end{aligned}$$

If they hold, output 1. Otherwise output 0.

Collision-resistant hash functions. A collision-resistant hash function family is defined as a pair $(\mathcal{H}, \mathsf{Hash})$ of two algorithms: the hash key generation algorithm $\mathcal{H}$ is a probabilistic polynomial-time algorithm that on input security parameter 1^k outputs a hash key hk; the hash algorithm Hash is a deterministic polynomial-time algorithm that on input the hash key hk and a message M outputs a hash value h; a collision-resistant hash function family is required to satisfy that for all probabilistic polynomial-time algorithms $\mathcal{A}$ the probability $\Pr[\mathsf{hk} \leftarrow \mathcal{H}(1^k); (M, M') \leftarrow \mathcal{A}(\mathsf{hk}) : \mathsf{Hash}(\mathsf{hk}, M) = \mathsf{Hash}(\mathsf{hk}, M')]$ is negligible in k. We assume that the length of the hash value h is determined by the security parameter 1^k and denote $\ell_{\mathcal{H}} = \ell_{\mathcal{H}}(k)$.

Attribute-based signatures. An attribute-based signature scheme is defined by the following four algorithms:

$\mathsf{AttrSetup}(1^k, 1^\ell) \rightarrow (\mathsf{pp}, \mathsf{msk})$**.** The setup algorithm takes as input the security parameter 1^k and the length ℓ of attributes, and outputs the public parameter pp and the master secret key msk.

$\mathsf{AttrGen}(\mathsf{pp}, \mathsf{msk}, x) \rightarrow \mathsf{sk}_x$**.** The signing key generation algorithm takes as input the public parameter pp, the master secret key msk, and the attribute x, and outputs the signing key sk_x for x.

$\mathsf{AttrSign}(\mathsf{pp}, \mathsf{sk}_x, M, C) \rightarrow \sigma$**.** The signing algorithm takes as input the public parameter pp, the signing key sk_x, the message M, and the circuit C, and outputs the signature σ.

$\mathsf{AttrVerify}(\mathsf{pp}, M, C, \sigma) \rightarrow 1/0$**.** The verification algorithm takes as input the public parameter pp, the message M, the circuit C, and the signature σ, and outputs 1 or 0 indicating the validity of the signature.

As the correctness condition, it is required to satisfy that for all k, $\ell \in \mathbb{N}$, $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{AttrSetup}(1^k, 1^\ell)$, $x \in \{0,1\}^\ell$, $\mathsf{sk}_x \leftarrow \mathsf{AttrGen}(\mathsf{pp}, \mathsf{msk}, x)$, $M \in \{0,1\}^*$, and C such that $C(x) = 1$, it holds that $\mathsf{AttrVerify}(\mathsf{pp}, M, C, \mathsf{AttrSign}(\mathsf{pp}, \mathsf{sk}_x, M, C)) = 1$.

We define two security notions for attribute-based signatures. The first notion is privacy, which requires the signature to not leak any information on the signer's identity and attribute beyond the fact that the attribute satisfies the predicate. The other notion is unforgeability, which requires any collusion of signers is unable to forge a new signature with a predicate which is not satisfied by any attribute in the collusion even if they see signatures on messages of their choice.

Definition 1. *An attribute-based signature scheme is perfectly private, if for all* k*,* $\ell \in \mathbb{N}$*,* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{AttrSetup}(1^k, 1^\ell)$*,* x_0*,* $x_1 \in \{0,1\}^\ell$*,* C *such that* $C(x_0) = C(x_1) = 1$*,* $\mathsf{sk}_0 \leftarrow \mathsf{AttrGen}(\mathsf{pp}, \mathsf{msk}, x_0)$*,* $\mathsf{sk}_1 \leftarrow \mathsf{AttrGen}(\mathsf{pp}, \mathsf{msk}, x_1)$*, and* $M \in \{0,1\}^*$*, the distribution* $\mathsf{AttrSign}(\mathsf{pp}, \mathsf{sk}_0, M, C)$ *and* $\mathsf{AttrSign}(\mathsf{pp}, \mathsf{sk}_1, M, C)$ *distributes identically.*

Definition 2. *An attribute-based signature scheme is adaptively unforgeable if the probability that the adversary wins in the following experiment is negligible in* k*:*

1. *The experiment sets up a public parameter and a master secret key as* $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{AttrSetup}(1^k, 1^\ell)$*. Then the experiment sends the adversary* pp*.*
2. *The adversary is allowed to access the key reveal oracle and the signing oracle: the former, given a query* x*, returns* $\mathsf{sk}_x \leftarrow \mathsf{AttrGen}(\mathsf{pp}, \mathsf{msk}, x)$*; the latter, given a query* (M, C)*, returns* $\sigma \leftarrow \mathsf{AttrSign}(\mathsf{pp}, \mathsf{sk}, M, C)$ *with arbitrary* $\mathsf{sk} \leftarrow \mathsf{AttrGen}(\mathsf{pp}, \mathsf{msk}, x)$ *such that* $C(x) = 1$*.*
3. *The adversary halts with output* (M^*, C^*, σ^*)*.*
4. *The adversary wins if the following three conditions hold: (i)* $\mathsf{AttrVerify}(\mathsf{pp}, M^*, C^*, \sigma^*) = 1$*, (ii) the adversary did not query* x *such that* $C^*(x) = 1$*, and (iii) the adversary did not query* (M^*, C^*) *to the signing oracle.*

3 Attribute-Based Signatures for Circuits

In this section we present our attribute-based signature scheme. We assume the input length ℓ is longer than or equal to the output length $\ell_{\mathcal{H}}$ of the hash function, i.e., $\ell \geq \ell_{\mathcal{H}}$. If it does not, we can simply think of a circuit that ignores the extra inputs.

Before presenting the concrete scheme, we explain an overview of the scheme.

As stated in the introduction, the basic idea is that the authority issues a signature (a certificate) on an attribute to certify that the corresponding signer is allowed to sign in the name of his attribute. This corresponds to the $\mathsf{AttrGen}$ algorithm, which computes a structure-preserving signature on the given attribute.

To sign anonymously, the signer proves the knowledge of the certificate received from the authority, as well as proves that the certified attribute satisfies the public circuit. To do this, the signer computes commitments to all the assignments to each wire. Then for each triple (u, v, w) which are connected by a NAND gate, the signer proves that the triple satisfies the NAND relation $1 - u \cdot v = w$. This is implemented by Eqs. (2), (5), and (6).

Since we are instantiating our scheme with a Type III pairing, for each wire we need two commitments in both $\mathbb{G}_1$ and $\mathbb{G}_2$. This is because we need to take a pairing of two wire assignments (Eqs. (5) and (6)) for proving the NAND relation of the three wires. This further requires the signer to prove that two commitments are commitments to the same message. This is done by proving Eqs. (3) and (4), which ensure that the exponents of W_i and $\tilde{W}_i$ are identical.

Lastly, the OR-proof technique is implemented by modifying the circuit C into $\hat{C}$ as in Eq. (1). This circuit ensures that the input $(X_2, \ldots, X_{l+1})$ is either a satisfying assignment of C or the hash value h. Eq. (2) ensures that θ is a valid signature on $(X_2, \ldots, X_{l+1})$. They constitute a proof of knowledge of a signature on an attribute or a signature on the dummy attribute determined by the message.

The full description of our scheme is as follows.

$\mathsf{AttrSetup}(1^k, 1^\ell)$**.** Given a security parameter 1^k and an input size 1^ℓ for circuit, generate bilinear group parameter $\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g}) \leftarrow \mathcal{G}(1^k)$, a witness indistinguishable common reference string $\mathsf{crs} \leftarrow \mathsf{WISetup}(\mathsf{gk})$, a verification key and a signing key $(\mathsf{vk}_{\mathsf{Sign}}, \mathsf{sk}_{\mathsf{Sign}}) \leftarrow \mathsf{Kg}(\mathsf{gk}, 1^{\ell+1})$ and a hash key $\mathsf{hk} \leftarrow \mathcal{H}(1^k)$. Set $\mathsf{pp} = (\ell, \mathsf{crs}, \mathsf{vk}_{\mathsf{Sign}}, \mathsf{hk})$ and $\mathsf{msk} \leftarrow \mathsf{sk}_{\mathsf{Sign}}$, and output $(\mathsf{pp}, \mathsf{msk})$.

$\mathsf{AttrGen}(\mathsf{pp}, \mathsf{msk}, x)$**.** Parse x as $(x_1, \ldots, x_\ell)$. Generate a structure-preserving signature θ on the message

$$(g^0, g^{x_1}, \ldots, g^{x_\ell}) \in {\mathbb{G}_1}^{\ell+1}.$$

Set $\mathsf{sk}_x \leftarrow (x, \theta)$ and output sk_x.

$\mathsf{AttrSign}(\mathsf{pp}, \mathsf{sk}_x, M, C)$**.** Parse sk_x into $((x_1, \ldots, x_\ell), \theta)$ and proceed as follows:

1. Let $h \leftarrow \mathsf{Hash}(\mathsf{hk}, \langle M, C \rangle)$. Expand the circuit C into a larger circuit $\hat{C}$ with $\ell + 1$-bit input as

$$\begin{aligned}\hat{C}(X_1, X_2, \ldots, X_{\ell+1}) = 1 \\ \iff \Big(X_1 = 0 \wedge C(X_2, \ldots, X_{\ell+1}) = 1\Big) \\ \vee \Big(X_1 = 1 \wedge X_2 \| \cdots \| X_{\ell_{\mathcal{H}+1}} = h\Big) \quad (1)\end{aligned}$$

where the hash value h is hard-wired into $\hat{C}$. Let N be the number of gates in $\hat{C}$ and I_1 and I_2 be the functions that specify the topology of $\hat{C}$.

2. Let $X_1 \leftarrow 0$, $X_2 \leftarrow x_1$, ..., $X_{\ell+1} \leftarrow x_\ell$, and then compute the assignment to each non-input wires in $\hat{C}$: for all $i = (\ell+1)+1$, ..., $(\ell+1)+(N-1)$

$$X_i \leftarrow 1 - X_{I_1(i)} \cdot X_{I_2(i)}.$$

3. For all $i = 1, \ldots, (\ell + 1) + (N - 1)$, let
$$W_i \leftarrow g^{X_i}, \qquad \tilde{W}_i \leftarrow \tilde{g}^{X_i}.$$
4. Compute a Groth-Sahai commitment com_θ to θ.
5. For all $i = 1, \ldots, (\ell + 1) + (N - 1)$, compute Groth-Sahai commitments com_{W_i} to W_i and $\mathsf{com}_{\tilde{W}_i}$ to $\tilde{W}_i$.
6. Generate a proof π_{Sign} for the verification equation
$$\mathsf{Verify}(\mathsf{vk}_{\mathsf{Sign}}, (W_1, \ldots, W_{\ell+1}), \theta) = 1. \tag{2}$$
7. For all $i = 1, \ldots, \ell + 1$, generate proofs π_i proving the equation
$$e(g, \tilde{W}_i) = e(W_i, \tilde{g}). \tag{3}$$
8. For all $i = (\ell + 1) + 1, \ldots, (\ell + 1) + (N - 1)$, generate proofs π_i proving the equations
$$e(g, \tilde{W}_i) = e(W_i, \tilde{g}), \tag{4}$$
$$e(W_{I_1(i)}, \tilde{W}_{I_2(i)})e(W_i, \tilde{g}) = e(g, \tilde{g}). \tag{5}$$
9. Generate a proofs $\pi_{(\ell+1)+N}$ proving
$$e(W_{I_1((\ell+1)+N)}, \tilde{W}_{I_2((\ell+1)+N)}) = 1. \tag{6}$$
10. Let
$$\sigma = (\mathsf{com}_\theta, \mathsf{com}_{W_1}, \ldots, \mathsf{com}_{W_{(\ell+1)+(N-1)}}, \mathsf{com}_{\tilde{W}_1}, \ldots, \mathsf{com}_{\tilde{W}_{(\ell+1)+(N-1)}}, \pi_{\mathsf{Sign}}, \pi_1, \ldots, \pi_{(\ell+1)+N})$$
and output σ.

$\mathsf{AttrVerify}(\mathsf{pp}, M, C, \sigma)$. Verify the proofs with respect to the circuit $\hat{C}$ in Eq. (1) and its topology I_1, I_2 defined by given M and C. Output 1 if all the proofs are verified as valid. Otherwise output 0.

Theorem 1. *Provided the proof system is perfectly witness indistinguishable, the above attribute-based signature scheme is perfectly private. Provided the proof system is perfectly extractable and perfectly witness indistinguishable, the signature scheme is existentially unforgeable, and the hash function family is collision resistant, the above attribute-based signature scheme is adaptively unforgeable.*

Proof. Perfect privacy directly followed from witness indistinguishability of the proof system.

For adaptive unforgeability, the proof proceeds with the following sequence of games:

Game 1. This game is identical to the experiment for adaptive unforgeability.

Game 2. In this game, the behavior of the signing oracle is modified as follows. Given a signing query (M, C), the experiment computes the hash value $h \leftarrow \mathsf{Hash}(\mathsf{hk}, \langle M, C \rangle)$, let $(h_1 \| \cdots \| h_{\ell_{\mathcal{H}}}) \leftarrow h$, compute a signature θ on the message

$$(g^1, g^{h_1}, \ldots, g^{h_{\ell_{\mathcal{H}}}}, 1, \ldots, 1) \in {\mathbb{G}_2}^{\ell+1}$$

with the master secret key $\mathsf{msk} = \mathsf{sk}_{\mathsf{Sign}}$, and then use θ as the witness to compute a signature σ.

Game 3. In this game, the common reference string crs in pp is switched to the extractable common reference string crs generated by the $\mathsf{ExtSetup}$ algorithm as $(\mathsf{crs}, \mathsf{ek}) \leftarrow \mathsf{ExtSetup}(1^k)$.

We denote by succ_i the event that the adversary wins in Game i. We hereafter bound $\Pr[\mathsf{succ}_1]$ to be negligible. From the triangle inequality,

$$\begin{aligned}\Pr[\mathsf{succ}_1] &= \Pr[\mathsf{succ}_1] - \Pr[\mathsf{succ}_2] + \Pr[\mathsf{succ}_2] - \Pr[\mathsf{succ}_3] + \Pr[\mathsf{succ}_3] \\ &\le |\Pr[\mathsf{succ}_1] - \Pr[\mathsf{succ}_2]| + |\Pr[\mathsf{succ}_2] - \Pr[\mathsf{succ}_3]| + \Pr[\mathsf{succ}_3].\end{aligned}$$

We bound these three terms. The first term $|\Pr[\mathsf{succ}_1] - \Pr[\mathsf{succ}_2]|$ is negligible, due to the witness indistinguishability of the Groth-Sahai proof system. The second term $|\Pr[\mathsf{succ}_2] - \Pr[\mathsf{succ}_3]|$ is also negligible, because the two types of common reference string are indistinguishable.

For the last term, we introduce an event coll. The event coll denotes the event that $\mathsf{Hash}(\mathsf{hk}, \langle M^*, C^* \rangle)$ collides to some of $\mathsf{Hash}(\mathsf{hk}, \langle M, C \rangle)$ where (M, C) is one of the signing queries. Now we have that

$$\Pr[\mathsf{succ}_3] = \Pr[\mathsf{succ}_3 \land \mathsf{coll}] + \Pr[\mathsf{succ}_3 \land \neg\mathsf{coll}].$$

The probability $\Pr[\mathsf{succ}_3 \land \mathsf{coll}]$ is negligible due to the collision-resistance of the hash function. For a formal proof, we construct a simulator that attacks the collision resistance of the hash function family.

Setup. The simulator receives a hash key hk from the experiment. The simulator then generates an extractable common reference string as $(\mathsf{crs}, \mathsf{ek}) \leftarrow \mathsf{ExtSetup}(1^k)$ and verification and signing keys $(\mathsf{vk}_{\mathsf{Sign}}, \mathsf{sk}_{\mathsf{Sign}}) \leftarrow \mathsf{Kg}(1^k)$, and then sets $\mathsf{pp} \leftarrow (\ell, \mathsf{crs}, \mathsf{vk}, \mathsf{hk})$ and sends pp to the adversary.

Key reveal query. When the adversary requests the signing key for $x = (x_1, \ldots, x_\ell)$, the simulator runs the signing algorithm to obtain a signature $\theta \leftarrow \mathsf{Sign}(\mathsf{sk}_{\mathsf{Sign}}, (g^0, g^{x_1}, \ldots, g^{x_\ell}))$. The simulator responds with $\mathsf{sk}_x = (x, \theta)$.

Signing query. When the adversary requests a signature on M under a circuit C, the simulator computes the hash value $h \leftarrow \mathsf{Hash}(\mathsf{hk}, \langle M, C \rangle)$, lets $(h_1 \| \cdots \| h_{\ell_{\mathcal{H}}}) \leftarrow h$, then further computes the signature $\theta \leftarrow \mathsf{Sign}(\mathsf{sk}_{\mathsf{Sign}}, (g^1, g^{h_1}, \ldots, g^{h_{\ell_{\mathcal{H}}}}, 1, \ldots, 1))$, the circuit $\hat{C}$ as in Eq. (1), and proof π using θ as the witness. The simulator responds with $\sigma = \pi$.

Forgery. When the adversary outputs a tuple (M^*, C^*, σ^*), the simulator searches for a signing query (M, C) that satisfies $\mathsf{Hash}(\mathsf{hk}, \langle M, C \rangle) = \mathsf{Hash}(\mathsf{hk}, \langle M^*, C^* \rangle)$. If it is found and the winning condition (i)–(iii) in Definition 2 is satisfied, the simulator outputs $(\langle M, C \rangle, \langle M^*, C^* \rangle)$ as a collision. Otherwise, the simulator outputs $(\perp, \perp)$.

The simulator successfully outputs a collision, if the event $\mathsf{succ}_3 \wedge \mathsf{coll}$ occurs. In particular, whenever the simulator outputs $(\langle M, C\rangle, \langle M^*, C^*\rangle)$, we have that $\langle M, C\rangle \neq \langle M^*, C^*\rangle$. This is because the winning condition forbids the adversary to output M^* and C^* which are queried to the signing oracle, and thus (M, C) differs from (M^*, C^*). Hence $\Pr[\mathsf{succ}_3 \wedge \mathsf{coll}]$ is negligible.

For $\Pr[\mathsf{succ}_3 \wedge \neg\mathsf{coll}]$, we construct a simulator that attacks the existential unforgeability of the underlying signature scheme. The construction of the simulator is as follows.

Setup. The simulator is given a verification key $\mathsf{vk}_{\mathsf{Sign}}$ of the signature scheme. The simulator sets up the extractable common reference string of the proof system as $(\mathsf{crs}, \mathsf{ek}) \leftarrow \mathsf{ExtSetup}(1^k)$. The simulator sends $\mathsf{pp} = (\ell, \mathsf{crs}, \mathsf{vk}_{\mathsf{Sign}}, \mathsf{hk})$ to the adversary.

Key reveal query. When the adversary requests the signing key for an attribute $x = (x_1, \ldots, x_\ell)$, the simulator requests, to its signing oracle, a signature on the message

$$(g^0, g^{x_1}, \ldots, g^{x_\ell}) \in {\mathbb{G}_1}^{\ell+1}.$$

Then the simulator receives a signature θ. The simulator sends $\mathsf{sk}_x = \theta$ to the adversary.

Signing query. When the adversary requests a signature on a message M under the circuit C, the simulator computes the hash value $h = (h_1 \| \cdots \| h_{\ell_{\mathcal{H}}}) \leftarrow \mathsf{Hash}(\mathsf{hk}, \langle M, C\rangle)$, then requests a signature on the message

$$(g^1, g^{h_1}, \ldots, g^{h_{\ell_{\mathcal{H}}}}, 1, \ldots, 1) \in {\mathbb{G}_1}^{\ell+1}$$

to its signing oracle. The simulator receives a signature θ. The simulator computes a proof π using the signature θ as the witness. The simulator sends $\sigma = \pi$ to the adversary.

Forgery. When the adversary outputs a forgery (M^*, C^*, σ^*), the simulator extracts the witness

$$\theta,\, W_1,\, \ldots,\, W_{(\ell+1)+(N-1)},\, \tilde{W}_1,\, \ldots,\, \tilde{W}_{(\ell+1)+(N-1)}.$$

Due to the extractability of the Groth-Sahai proof system, we can assume that the witness satisfies Eqs. (2)–(6).

Now below we argue that the pair

$$((W_1, \ldots, W_{\ell+1}), \theta)$$

constitutes a legitimate forgery for the underlying signature scheme. We have three cases to be dealt with.

1. Assume that $(W_1, \ldots, W_{\ell+1})$ is of the form

$$(g^{X_1}, \ldots g^{X_{\ell+1}}) \in {\mathbb{G}_2}^{\ell+1} \quad \text{where } X_1 = 0 \text{ and } X_2,\, \ldots,\, X_{\ell+1} \in \{0, 1\}.$$

In this case, due to Eqs. (3)–(6), we have that $\hat{C}(X_1, \ldots, X_{\ell+1}) = 1$, and hence we also have that $C(X_2, \ldots X_{\ell+1}) = 1$. Because the experiment forbids the adversary to query such $(X_2, \ldots, X_{\ell+1})$ as a key reveal query, we can conclude that the simulator has not queried $(g^0, g^{X_2}, \ldots g^{X_{\ell+1}})$ to its signing oracle. Hence, due to the equation Eq. (2), the pair $((g^0, g^{X_2}, \ldots, g^{X_{\ell+1}}), \theta)$ constitutes a legitimate forgery to the signature scheme.

2. Assume that $(W_1, \ldots, W_{\ell+1})$ is of the form

$$(g^{X_1}, \ldots, g^{X_{\ell+1}}) \in {\mathbb{G}_2}^{\ell+1}$$
$$\text{where } X_1 = 1,\ X_2,\ \ldots,\ X_{\ell_{\mathcal{H}}+1} \in \{0,1\},\ X_{\ell_{\mathcal{H}}+2} = \cdots = X_{\ell+1} = 0$$

In this case, due to Eqs. (3)–(6), we have that $(X_2 \| \cdots \| X_{\ell_{\mathcal{H}}+1}) = \mathsf{Hash}(\mathsf{hk}, \langle C^*, M^* \rangle)$. Since we are now considering the event $\neg\mathsf{coll}$, we have that the adversary has not queried (C, M) such that $\mathsf{Hash}(\mathsf{hk}, \langle C, M \rangle) = \mathsf{Hash}(\mathsf{hk}, \langle C^*, M^* \rangle)$ to the signing oracle. Therefore the simulator has not queried $(g^1, g^{X_2}, \cdots g^{X_{\ell_{\mathcal{H}}}}, 1, \ldots, 1)$ to its signing oracle, and thus $((g^1, g^{X_2}, \cdots g^{X_{\ell_{\mathcal{H}}}}, 1, \ldots, 1), \theta)$ constitutes a legitimate forgery.

3. Assume that $(W_1, \ldots, W_{\ell+1})$ is neither of the above two forms. In this case, the simulator does not issue any query of this form at all, and thus $((W_1, \ldots, X_{\ell+1}), \theta)$ is a legitimate forgery.

In any case, the pair $((W_1, \ldots, X_{\ell+1}), \theta)$ constitutes the forgery, and thus the simulator outputs this pair as a forgery.

The above construction shows that whenever the event $\mathsf{succ}_3 \wedge \neg\mathsf{coll}$ occurs, the simulator succeeds in producing the forgery of the signature scheme. It implies that $\Pr[\mathsf{succ}_3 \wedge \neg\mathsf{coll}]$ is negligible. □

4 Performance

In this section we compare our scheme with the Maji et al. (MPR11) schemes [21] and the Okamoto-Takashima (OT11) scheme [24]. Table 1 shows a brief comparison among the existing schemes and our scheme. In the table the three MPR11 schemes (1)–(3) are respectively the Boneh-Boyen signature based scheme, the Waters signature based scheme, and the scheme proven secure in the generic group model. For the first two schemes, we show the performance in the SXDH setting. Our scheme is instantiated with the Kiltz-Pan-Wee structure-preserving signature scheme from the SXDH assumption [16] and the Groth-Sahai proof system in the SXDH setting [14]. We also note that all the five schemes in the table are instantiated in prime order groups.

Table 2 shows a detailed calculation of the signature size of our scheme. The center and right columns respectively show the number of the group elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ that is required for each component of a signature.

As the table shows, our scheme achieves a comparable performance with the existing schemes, while the class of supported predicates is drastically wider than the existing schemes. In addition, the assumption from which the scheme is proven secure is also comparable with or in some case identical to the existing schemes.

Table 1. Comparison among pairing-based attribute-based signature schemes.

Scheme	Signature size	Assumption	Predicate
MPR11 (1) [21]	$36s + 2t + 24ks$	q-SDH, SXDH	Monotone span program
MPR11 (2) [21]	$28s + 2t + 12k + 8$	SXDH	Monotone span program
MPR11 (3) [21]	$s + t + 2$	Generic group	Monotone span program
OT11 [24]	$9s + 11$	DLIN	Non-monotone span program
Ours	$12\ell + 20N + 26$	SXDH	Non-monotone circuit

k: The security parameter
$s \times t$: The size of the monotone span program
ℓ: The input length of the circuit
N: The number of the gate in the circuit

Table 2. Signature size of our scheme.

	$\mathbb{G}_1$	$\mathbb{G}_2$
com_θ	12	2
com_{W_i}	$2(\ell + N)$	
$\mathsf{com}_{\tilde{W}_i}$		$2(\ell + N)$
π_{Sign}	4	8
$\pi_1, \ldots, \pi_{\ell+1}$	$4(\ell + 1)$	$4(\ell + 1)$
$\pi_{(\ell+1)+1}, \ldots, \pi_{(\ell+1)+(N-1)}$	$8(N - 1)$	$8(N - 1)$
$\pi_{(\ell+1)+N}$	4	4
Total	$6\ell + 10N + 16$	$6\ell + 10N + 10$

ℓ: The input length of the circuit
N: The number of the gates in the circuit

References

1. Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-preserving signatures and commitments to group elements. In: Rabin, T. (ed.) CRYPTO 2010, Lecture Notes in Computer Science, vol. 6223, pp. 209–236. Springer Berlin Heidelberg (2010)
2. Bellare, M., Fuchsbauer, G.: Policy-based signatures. In: Krawczyk, H. (ed.) PKC 2014, Lecture Notes in Computer Science, vol. 8383, pp. 520–537. Springer Berlin Heidelberg (2014)
3. Bellare, M., Namprempre, C., Neven, G.: Security proofs for identity-based identification and signature schemes. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004, Lecture Notes in Computer Science, vol. 3027, pp. 268–286. Springer Berlin Heidelberg (2004)
4. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of Cryptology 21(2), 149–177 (Sep 2008)

5. Boneh, D., Papakonstantinou, P.A., Rackoff, C., Vahlis, Y., Waters, B.: On the impossibility of basing identity based encryption on trapdoor permutations. In: 49th Annual Symposium on Foundations of Computer Science. pp. 283–292. IEEE (2008)
6. Chen, C., Chen, J., Lim, H.W., Zhang, Z., Feng, D., Ling, S., Wang, H.: Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures. In: Dawson, E. (ed.) CT-RSA 2013, Lecture Notes in Computer Science, vol. 7779, pp. 50–67. Springer Berlin Heidelberg (2013)
7. El Kaafarani, A., Ghadafi, E., Khader, D.: Decentralized traceable attribute-based signatures. In: Benaloh, J. (ed.) CT-RSA 2014, Lecture Notes in Computer Science, vol. 8366, pp. 327–348. Springer International Publishing (2014)
8. Escala, A., Herranz, J., Morillo, P.: Revocable attribute-based signatures with adaptive security in the standard model. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011, Lecture Notes in Computer Science, vol. 6737, pp. 224–241. Springer Berlin Heidelberg (2011)
9. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II, Lecture Notes in Computer Science, vol. 8043, pp. 479–499. Springer Berlin Heidelberg (2013)
10. Ghadafi, E.: Stronger security notions for decentralized traceable attribute-based signatures and more efficient constructions. In: Nyberg, K. (ed.) CT-RSA 2015, Lecture Notes in Computer Science, vol. 9048, pp. 391–409. Springer International Publishing (2015)
11. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing. pp. 545–554. ACM (2013)
12. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Predicate encryption for circuits from LWE. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part II, Lecture Notes in Computer Science, vol. 9216, pp. 503–523. Springer Berlin Heidelberg (2015)
13. Groth, J., Ostrovsky, R., Sahai, A.: New techniques for noninteractive zero-knowledge. J. ACM 59(3), 11:1–11:35 (Jun 2012)
14. Groth, J., Sahai, A.: Efficient noninteractive proof systems for bilinear groups. SIAM J. Comput. 41(5), 1193–1232 (Oct 2012)
15. Herranz, J., Laguillaumie, F., Libert, B., Ràfols, C.: Short attribute-based signatures for threshold predicates. In: Dunkelman, O. (ed.) CT-RSA 2012, Lecture Notes in Computer Science, vol. 7178, pp. 51–67. Springer Berlin Heidelberg (2012)
16. Kiltz, E., Pan, J., Wee, H.: Structure-preserving signatures from standard assumptions, revisited. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015, Part II, Lecture Notes in Computer Science, vol. 9216. Springer Berlin Heidelberg (2015)
17. Kurosawa, K., Heng, S.H.: From digital signature to ID-based identification/signature. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004, Lecture Notes in Computer Science, vol. 2947, pp. 248–261. Springer Berlin Heidelberg (2004)
18. Li, J., Au, M.H., Susilo, W., Xie, D., Ren, K.: Attribute-based signature and its applications. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. pp. 60–69. ACM (2010)
19. Li, J., Kim, K.: Attribute-based ring signatures. Cryptology ePrint Archive, Report 2008/394 (2008), `http://eprint.iacr.org/`
20. Maji, H., Prabhakaran, M., Rosulek, M.: Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. Cryptology ePrint Archive, Report 2008/328 (2008), `http://eprint.iacr.org/`

21. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-based signatures. In: Kiayias, A. (ed.) CT-RSA 2011, Lecture Notes in Computer Science, vol. 6558, pp. 376–392. Springer Berlin Heidelberg (2011)
22. Maurer, U.M., Yacobi, Y.: Non-interactive public-key cryptography. In: Davies, D.W. (ed.) EUROCRYPT 1991, Lecture Notes in Computer Science, vol. 547, pp. 498–507. Springer Berlin Heidelberg (1991)
23. Nandi, M., Pandit, T.: On the power of pair encodings: Frameworks for predicate cryptographic primitives. Cryptology ePrint Archive, Report 2015/955 (2015), http://eprint.iacr.org/
24. Okamoto, T., Takashima, K.: Efficient attribute-based signatures for non-monotone predicates in the standard model. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011, Lecture Notes in Computer Science, vol. 6571, pp. 35–52. Springer Berlin Heidelberg (2011)
25. Okamoto, T., Takashima, K.: Decentralized attribute-based signatures. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013, Lecture Notes in Computer Science, vol. 7778, pp. 125–142. Springer Berlin Heidelberg (2013)
26. Shahandashti, S.F., Safavi-Naini, R.: Threshold attribute-based signatures and their application to anonymous credential systems. In: Preneel, B. (ed.) AFRICACRYPT 2009, Lecture Notes in Computer Science, vol. 5580, pp. 198–216. Springer Berlin Heidelberg (2009)
27. Tang, F., Li, H., Liang, B.: Attribute-based signatures for circuits from multilinear maps. In: Chow, S.S., Camenisch, J., Hui, L.C., Yiu, S.M. (eds.) ISC 2014, Lecture Notes in Computer Science, vol. 8783, pp. 54–71. Springer International Publishing (2014)
28. Wang, Q., Chen, S.: Attribute-based signature for threshold predicates from lattices. Security and Communication Networks 8(5), 811–821 (Mar 2015)
29. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005, Lecture Notes in Computer Science, vol. 3494, pp. 114–127. Springer Berlin Heidelberg (2005)