Attribute-Based Two-Tier Signatures: Definition and Construction

Hiroaki Anada[1,2], Seiko Arita[2], and Kouichi Sakurai[3,1]

[1] Institute of Systems, Information Technologies and Nanotechnologies
7F, 2–1–22, Momochihama, Sawara–ku, Fukuoka, 814–0001 JAPAN
anada@isit.or.jp
[2] Institute of Information Security
2–14–1 Tsuruya–cho, Kanagawa–ku, Yokohama, 221–0835 JAPAN
arita@iisec.ac.jp
[3] Department of Informatics, Graduate School and Faculty of Information Science
and Electrical Engineering, Kyushu University
W2–712, 744 Motooka, Nishi–ku, Fukuoka, 819–0395 JAPAN
sakurai@inf.kyushu-u.ac.jp

Abstract. Attribute-based signature scheme (ABS) is a functional variant of digital signature scheme proposed in 2008 by Maji et al. The two basic requirements of ABS (and a hard task to achieve) is collusion resistance and attribute privacy. In this paper, we employ the two-tier signature (TTS) technique to achieve the collusion resistance. Here TTS was proposed in 2007 by Bellare et al., where a signer receives two tier secret keys sequentially. The secondary secret key is served as a one-time key at the timing of signing. First, we propose a definition of an attribute-based two-tier signature scheme (ABTTS). Then we provide ABTTS concretely that enjoys existential unforgeability against chosen-message attacks, collusion resistance and attribute privacy, in the standard model. For the construction, enhancing the Camenisch-Lysyanskaya signature, we construct signature bundle schemes that are secure under the Strong RSA assumption and the Strong Diffie-Hellman assumption, respectively. These signature bundle schemes enable ABTTS to achieve attribute privacy. Then, using the signature bundle as a witness in the Σ-protocol of the boolean proof, we obtain attribute-based identification schemes (ABIDs). Finally, by applying the TTS technique to ABIDs, we achieve ABTTSs. A feature of our construction is that ABTTS in the RSA setting is pairing-free.

Keywords: digital signature, attribute-based, two-tier keys.

1 Introduction

Digital signature scheme is one of the most widely recognized cryptographic primitives. Since its invention, functional variants have been proposed, which include *attribute-based signature schemes* (ABS) developed by Guo and Zeng [12] and Maji, Prabhakaran and Rosulek [14] in 2008. In ABS, a message m is associated with a signing policy f that is described as a boolean formula over signers' attributes. Then only signers with attributes that satisfy f can make a legitimate signature σ on m. A verifier can check whether the signature σ is valid in accordance with the signing policy f. The two basic requirements of ABS (and a hard task to achieve) is collusion resistance against collecting secret keys and attribute privacy. Intuitively, ABS is called to have attribute privacy if any cheating verifier cannot distinguish two distributions of signatures each of which is generated by different satisfying attribute set.

A two-tier signature scheme (TTS) is a digital signature scheme proposed in 2007 by Bellare et al. [3], in which a signer receives two tier secret keys sequentially, the latter of which is served as a one-time signing key at the timing of signing. Accordingly, two tier public keys are issued sequentially. These two-tier keys fit the Fiat-Shamir signature scheme and enable it to achieve existential unforgeability against chosen-message attacks (EUF-CMA) in the standard model.

Our Contribution Our first contribution is to define the syntax of an attribute-based two-tier signature scheme (ABTTS) for the first time. The reason why we introduce ABTTS (in a construction of ABS) is to achieve the collusion resistance by employing the TTS technique. The issuer of a secondary secret key can check integrity of components in a primary secret keys so that the issuer can avoid collusion attacks.

Our second contribution is to provide ABTTS concretely that enjoys existential unforgeability against chosen-message attacks, collusion resistance and attribute privacy, in the standard model. It is interesting from the view point of theory (and also efficiency) that our ABTTS in the RSA setting is pairing-free.

Our Approach to Concrete Construction First, enhancing the Camenisch-Lysyanskaya signature, we construct signature bundle schemes that are secure under the Strong RSA assumption and the Strong Diffie-Hellman assumption, respectively. These signature bundle schemes later enable ABTTS to achieve attribute privacy. Then, using the signature bundle as a witness in the Σ-protocol of the boolean proof, we obtain attribute-based identification schemes (ABIDs). Finally, by applying the TTS technique to ABIDs, we achieve ABTTSs.

Table 1. Comparison of security, functionality and signature length.

Scheme	Security Model	Assumption	Access Formula	Pairing-Free	Attribute Privacy	Length of Signature
Maji et al. [14]	Std.	q-SDH $\wedge$DLIN	Mono.	-	✓(info.)	$(2\lambda)\times$ $(51l+2r+18\lambda l)$
OT [16]	Std.	DLIN $\wedge$CR	Non-m.	-	✓(info.)	$(2\lambda)\times$ $(9l+11)$
Herranz [13]	R.O.	q-SRSA$\wedge$[DDH in $QR(N)$]$\wedge$CR	Mono.	✓	✓(comp.)	$\lambda_{\mathrm{rsa}}(5+\frac{\kappa}{\lambda_{\mathrm{rsa}}})l$ $+\lambda_{\mathrm{rsa}}3-\kappa(\theta-1)$
Ghadafi et al. [8]	R.O.	q-SDH$\wedge$DDH$\wedge$ DLog$\wedge$CR	Mono.	-	-	$(2\lambda)(3l+r+3)$ $+\lambda(8l+4)$
Anada et al. [2]	R.O.	[DLog$\vee$RSAInv] $\wedge$CR	Mono.	✓	-	$(2\hat{\lambda})l$ $+\hat{\lambda}(4l-1)$
Our ABTTS	Std.	[q-SRSA$\vee q$-SDH] $\wedge$CR	Mono.	✓	✓(info.)	$(2\hat{\lambda})2l$ $+\hat{\lambda}2l$

Comparison: Security, Functionality and Signature Length We compare our ABTTS with previously proposed schemes from the view point of security, functionality and signature length. The comparison is summarized in Table 1 with notations as follows. A prime of bit length λ (the security parameter in the discrete logarithm setting) is denoted by p. Though a pairing map e should be analysed for the asymmetric bilinear groups [11], we simply evaluate the symmetric case in which both source groups are $\mathbb{G}_p$ of order p. We assume that an element of $\mathbb{G}_p$ is represented by 2λ bits. l and r mean the number of rows and columns of the share-generating matrix for monotone access formula f (that is, an access structure), respectively. CR means the collision resistance of an employed hash function. q-SDH means the Strong Diffie-Hellman assumption with q-type input for bilinear groups [4]. DLIN means the Decisional Linear assumption for bilinear groups [16]. DDH means the Decisional Diffie-Hellman assumption for a cyclic group [8]. DLog means the Discrete Logarithm assumption for a cyclic group [8]. q-SRSA means the Strong RSA assumption with q-type input [13]. DDH in $QR(N)$ means the Decisional Diffie-Hellman assumption for quadratic residues modulo N (the RSA modulus) [13]. "info." means information-theoretic security and "comp." means computational security. λ_{rsa} means the security parameter in the RSA setting, and $\hat{\lambda}$ means the security parameter in either the RSA setting or the discrete logarithm setting.

First, note that our scheme assumes the secondary secret key and the secondary public key are issued as one-time keys at the timing of signing. This means the signer and the verifier should be on-line and they need to verify a certificate of the secondary public key. One possibility of executing such a process is to use Online Certificate Status Protocol (OCSP) by RFC 6990 [9].

The scheme of [16] has advantages in the security model, access formula and information-theoretically secure attribute privacy, whereas our ABS realizes shorter length of signature (less than a half). The scheme of [13] is in the RSA setting and its security parameter λ_{rsa} is almost 10 times longer than λ in the DLog setting. For example, $\lambda_{\mathrm{rsa}} = 2048$ is almost equivalent to $\lambda = 224$-bit security [17]. (θ is the threshold value of the threshold-type access structure. κ is explained in the work [13].) Therefore, our ABS in the DLog setting realizes shorter length of a signature.

2 Preliminaries

The security parameter is denoted as λ. Bit length of a string x is denoted as $|x|$. The expression "$a \overset{?}{=} b$" returns a value 1 if $a = b$ and 0 otherwise. The expression "$a \overset{?}{\in} S$" returns a value 1 if $a \in S$ and 0 otherwise.

***Σ*-protocol [6, 7]** A Σ-protocol on a binary NP relation R is a public coin 3-move protocol between interactive PPT algorithms $\mathcal{P}$ and $\mathcal{V}$ on initial input $(x, w) \in R$ for $\mathcal{P}$ and x for $\mathcal{V}$. x and w are called a statement and a witness, respectively. $\mathcal{P}$ sends the first message called a commitment $\textsc{Cmt}$, then $\mathcal{V}$ sends a random bit string called a challenge $\textsc{Cha}$, and $\mathcal{P}$ answers with a third message called a response $\textsc{Res}$. Then $\mathcal{V}$ applies a decision test on $(x, \textsc{Cmt}, \textsc{Cha}, \textsc{Res})$ to return accept (1) or reject (0). If $\mathcal{V}$ accepts, then the triple $(\textsc{Cmt}, \textsc{Cha}, \textsc{Res})$ is said to be an *accepting conversation.* $\textsc{Cha}$ is chosen uniformly at random from the challenge space $\textsc{ChaSp}(1^{\lambda}) := \{1, 0\}^{l(\lambda)}$ with $l(\cdot)$ being a super-log function.

The Σ-protocol is written by a PPT algorithm $\mathbf{\Sigma}$ as follows. $\textsc{Cmt} \leftarrow \mathbf{\Sigma}^1(x, w)$: the process of selecting the first message $\textsc{Cmt}$ according to the protocol $\mathbf{\Sigma}$ on input $(x, w) \in R$. Similarly we denote $\textsc{Cha} \leftarrow \mathbf{\Sigma}^2(1^{\lambda})$, $\textsc{Res} \leftarrow \mathbf{\Sigma}^3(x, w, \textsc{Cmt}, \textsc{Cha})$ and $b \leftarrow \mathbf{\Sigma}^{\mathrm{vrfy}}(x, \textsc{Cmt}, \textsc{Cha}, \textsc{Res})$. The Σ-protocol must possess the three properties: *completeness*, *special soundness* and *honest-verifier zero-knowledge* [6, 7]. As a zero-knowledge proof-of-knowledge system, we denote $\mathbf{\Sigma}$ as $\mathbf{ZKPK}[\gamma : \Gamma]$, where γ is a knowledge to be proved and Γ is the condition that γ should satisfy.

Signature Bundle Scheme [14] A signature bundle (a credential bundle in [14]) scheme $\mathtt{SB}$ is an extended notion of a signature scheme. It consists of three PPTs: $\mathtt{SB} = (\mathbf{SB}.\mathbf{KG}, \mathbf{SB}.\mathbf{Sign}, \mathbf{SB}.\mathbf{Vrfy})$.

$\mathbf{SB}.\mathbf{KG}(1^\lambda) \to (\mathrm{PK}, \mathrm{SK})$. Given 1^λ as input, it returns a public key PK and a secret key SK.

$\mathbf{SB}.\mathbf{Sign}(\mathrm{PK}, \mathrm{SK}, (m_i)_{i=1}^n) \to (\tau, (\sigma_i)_{i=1}^n)$. Given PK, SK and messages $(m_i)_{i=1}^n$, it returns a *tag* τ and signatures $(\sigma_i)_{i=1}^n$. n is bounded by a polynomial in λ.

$\mathbf{SB}.\mathbf{Vrfy}(\mathrm{PK}, (m_i)_{i=1}^n, (\tau, (\sigma_i)_{i=1}^n)) \to 1/0$. Given PK, messages $(m_i)_{i=1}^n$, a tag τ and signatures $(\sigma_i)_{i=1}^n$, it returns 1 or 0.

A PPT adversary $\mathcal{F}$ tries to make a forgery $((m_i^*)_{i=1}^{n^*}, (\tau^*, (\sigma_i^*)_{i=1}^{n^*}))$. Here τ^* is called a *target tag. An existential forgery by a chosen-message attack* is defined by:

```
$\mathbf{Expr}^{\text{euf-cma}}_{\mathtt{SB},\mathcal{F}}(1^\lambda, \mathcal{U})$
  $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathbf{SB}.\mathbf{KG}(1^\lambda), ((m_i^*)_{i=1}^{n^*}, (\tau^*, (\sigma_i^*)_{i=1}^{n^*})) \leftarrow \mathcal{F}^{\mathcal{SBSIGN}}(\mathrm{PK})$
  If $\mathbf{SB}.\mathbf{Vrfy}(\mathrm{PK}, (m_i^*)_{i=1}^{n^*}, (\tau^*, (\sigma_i^*)_{i=1}^{n^*})) = 1$
    then Return WIN else Return LOSE
```

Giving a vector of messages $(m_i)_{i=1}^n$, $\mathcal{F}$ queries $\mathcal{SBSIGN}(\mathrm{PK}, \mathrm{SK}, \cdot)$ for a valid signature bundle $(\tau, (\sigma_i)_{i=1}^n)$. τ^* should be different from any queried tag τ, or, whenever τ^* is equal to a queried tag τ, it should hold that $\{m_i^*\}_{i=1}^{n^*} \not\subseteq \{m_i\}_{i=1}^n$ for any queried $(m_i)_{i=1}^n$. The *advantage* of $\mathcal{F}$ over $\mathtt{SB}$ in the experiment of existential forgery by chosen-message attack is defined as $\mathbf{Adv}^{\text{euf-cma}}_{\mathtt{SB},\mathcal{F}}(\lambda, \mathcal{U}) \overset{\text{def}}{=} \Pr[\mathbf{Expr}^{\text{euf-cma}}_{\mathtt{SB},\mathcal{F}}(1^\lambda, \mathcal{U}) \text{ returns WIN}]$.

Definition 1 *$\mathtt{SB}$ is called existentially unforgeable against chosen-message attack if, for any PPT $\mathcal{F}$, $\mathbf{Adv}^{\text{euf-cma}}_{\mathtt{SB},\mathcal{F}}(\lambda, \mathcal{U})$ is negligible in λ.*

Access Structure [10] Let $\mathcal{U} = \{\mathtt{at}_i\}_{i=1}^u$ be an attribute universe. $|\mathcal{U}| = u$ is bounded by a polynomial in λ ($\mathcal{U}$ is called a small universe).

Let $f = f(X_{\mathtt{at}_1}, \dots, X_{\mathtt{at}_a})$ be a monotone boolean formula over $U = \{X_{\mathtt{at}}\}_{\mathtt{at}}$, where boolean connectives are AND-gate ($\wedge$) and OR-gate ($\vee$). In this paper, we assume that no NOT-gate ($\neg$) appears in f. In other words, we consider only a *monotone* access formula f.[4] We denote the set of subscripts (that is, attributes) $\{\mathtt{at}_1, \dots, \mathtt{at}_a\}$ as $\mathrm{At}(f)$ and the arity a as $\mathrm{arity}(f)$, respectively. For $S \in 2^{\mathcal{U}}$, we evaluate the boolean value of f at

[4] This limitation can be removed by adding *negation attributes* to $\mathcal{U}$ for each attribute in the original $\mathcal{U}$ though the size of the attribute universe $|\mathcal{U}|$ doubles.

S as: $f(S) \overset{\text{def}}{=} f\big(X_{\texttt{at}} \leftarrow [\texttt{at} \overset{?}{\in} S]; \texttt{at} \in \mathrm{At}(f)\big) \in \{1, 0\}$. We call a boolean formula f with this map an *access formula* over $\mathcal{U}$. An access formula corresponds to a signing policy in the case of attribute-based signatures.

An access formula f can be represented by a finite binary tree $\mathcal{T}_f$. Each inner node corresponds to an AND-gate ($\wedge$) or OR-gate ($\vee$) in f. Each leaf node l corresponds to a term $X_{\texttt{at}}$ (not a variable $X_{\texttt{at}}$) in f in one-to-one way. For a finite binary tree $\mathcal{T}$, we denote the root node, the set of all nodes, the set of all leaf nodes, the set of all inner nodes (all nodes excluding leaf nodes) and the set of all tree-nodes (all nodes excluding the root node) as $r(\mathcal{T})$, $\mathrm{Node}(\mathcal{T})$, $\mathrm{Leaf}(\mathcal{T})$, $\mathrm{iNode}(\mathcal{T})$ and $\mathrm{tNode}(\mathcal{T})$, respectively. Then an attribute map $\rho(\cdot)$ is defined as: $\rho : \mathrm{Leaf}(\mathcal{T}) \rightarrow \mathcal{U},\ \rho(l) \overset{\text{def}}{=} (\texttt{at}$ that corresponds to $l)$. If ρ is not injective, then we call the case *multi-use* of attributes.

Attribute-Based Identification Scheme [1] An attribute-based identification scheme, $\texttt{ABID}$, consists of four PPT algorithms: $(\mathbf{ABID}.\mathbf{Setup}, \mathbf{ABID}.\mathbf{KG}, \mathcal{P}, \mathcal{V})$.
$\mathbf{ABID}.\mathbf{Setup}(1^{\lambda},\mathcal{U}) \rightarrow (\mathbf{PK}, \mathbf{MSK})$**.** Given the security parameter 1^{λ} and an attribute universe $\mathcal{U}$, it returns a public key PK and a master secret key MSK.
$\mathbf{ABID}.\mathbf{KG}(\mathbf{PK}, \mathbf{MSK}, S) \rightarrow \mathbf{SK}_S$**.** Given the public key PK, the master secret key MSK and an attribute set $S \subset \mathcal{U}$, it returns a secret key SK_S that corresponds to S.
$\mathcal{P}(\mathbf{PK}, \mathbf{SK}_S)$ **and** $\mathcal{V}(\mathbf{PK}, f)$**.** $\mathcal{P}$ and $\mathcal{V}$ are interactive algorithms called a *prover* and a *verifier*, respectively. $\mathcal{P}$ takes as input the public key PK and the secret key SK_S. Here the secret key SK_S is given to $\mathcal{P}$ by an authority. $\mathcal{V}$ takes as input the public key PK and an access formula f. $\mathcal{P}$ is provided $\mathcal{V}$'s access formula f by the first move. $\mathcal{P}$ and $\mathcal{V}$ interact with each other for at most constant rounds. Then, $\mathcal{V}$ returns its decision 1 or 0. When it is 1, we say that $\mathcal{V}$ *accepts* $\mathcal{P}$ for f. When it is 0, we say that $\mathcal{V}$ *rejects* $\mathcal{P}$ for f. We demand correctness of $\texttt{ABID}$ that for any λ, for any S and for any f such that $f(S) = 1$, $\Pr[(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathrm{Setup}(1^{\lambda},\mathcal{U}); \mathrm{SK}_S \leftarrow \mathrm{KG}(\mathrm{PK}, \mathrm{MSK}, S); b \leftarrow \langle \mathcal{P}(\mathrm{PK}, \mathrm{SK}_S), \mathcal{V}(\mathrm{PK}, f)\rangle : b = 1] = 1$.

An adversary $\mathcal{A}$ tries to make a verifier $\mathcal{V}$ accept with an access formula f^* of his choice. Here f^* is called a *target access formula. A concurrent attack* is defined by:

$\mathbf{Exprmt}^{\mathrm{ca}}_{\texttt{ABID},\mathcal{A}}(1^{\lambda},\mathcal{U})$

$(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathbf{ABID}.\mathbf{Setup}(\lambda,\mathcal{U}),\ (f^*, st) \leftarrow \mathcal{A}^{\mathcal{KG},\mathcal{P}_j|_{j=1}^{q_{\mathrm{prv}}}}(\mathrm{PK},\mathcal{U})$

$b \leftarrow \langle \mathcal{A}(st), \mathcal{V}(\mathrm{PK}, f^*)\rangle$, If $b = 1$ then Return $\textsc{Win}$ else Return $\textsc{Lose}$

Giving an attribute set S_i, $\mathcal{A}$ queries $\mathcal{KG}(\mathrm{PK}, \mathrm{MSK}, \cdot)$ for the secret key SK_{S_i}. In addition, $\mathcal{A}$ invokes provers $\mathcal{P}_j(\mathrm{PK}, \mathrm{SK}_{\cdot})$, $j = 1, \dots, q'_{\mathrm{prv}}, \dots, q_{\mathrm{prv}}$, by giving a pair (S_j, f_j). Acting as a verifier with an access formula f_j, $\mathcal{A}$ interacts with each $\mathcal{P}_j(\mathrm{PK}, \mathrm{SK}_{S_j})$ concurrently. In the above we consider the *adaptive target* f^*. In key-extraction queries, each attribute set S_i must satisfy $f^*(S_i) = 0$. In interactions with each prover, $f^*(S_j) = 0$. The *advantage* of $\mathcal{A}$ over ABID in the game of concurrent attack is defined as $\mathbf{Adv}^{\mathrm{ca}}_{\mathtt{ABID},\mathcal{A}}(\lambda) \overset{\mathrm{def}}{=} \Pr[\mathbf{Exprmt}^{\mathrm{ca}}_{\mathtt{ABID},\mathcal{A}}(1^\lambda, \mathcal{U})$ returns $\textsc{Win}]$. ABID is called *secure against concurrent attacks* if, for any PPT $\mathcal{A}$, $\mathbf{Adv}^{\mathrm{ca}}_{\mathtt{ABID},\mathcal{A}}(\lambda)$ is negligible in λ.

Strong RSA Assumption [5] Let $p = 2p' + 1$ denote a *safe prime* (p' is also a prime). Let N denote the *special RSA modulus*; that is, $N = pq$ where $p = 2p'+1$ and $q = 2q'+1$ are two safe primes such that $|p'| = |q'| = \lambda - 1$. We denote the probabilistic algorithm that generates such N at random on input 1^λ as $\mathtt{RSAmod}$. Let $QR_N \subset \mathbb{Z}_N^*$ denote the set of quadratic residues modulo N; that is, elements $a \in \mathbb{Z}_N^*$ such that $a \equiv x^2 \bmod N$ for some $x \in \mathbb{Z}_N^*$. The strong RSA assumption [5] states that for any PPT $\mathcal{A}$, the following advantage is negligible in λ: $\mathbf{Adv}^{\mathrm{srsa}}_{\mathtt{RSAmod},\mathcal{S}}(\lambda, \mathcal{U}) := \Pr[N \leftarrow \mathtt{RSAmod}(1^\lambda), g \overset{\$}{\leftarrow} QR_N, (V, e) \leftarrow \mathcal{A}(N, g) : e > 1 \wedge V^e \equiv g \bmod N]$.

Strong Diffie-Hellman Assumption [4] Let p denote a prime of bit length λ. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ denote bilinear groups of order p, where $\mathbb{G}_1$ is generated by g, $\mathbb{G}_2$ is generated by h and $\mathbb{G}_T$ is generated by $e(g, h) \neq 1_{\mathbb{G}_T}$. We denote the probabilistic algorithm that generates such parameters params $:= (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ on input 1^λ as $\mathtt{BlGrp}$. Let q denote a number that is less than a fixed polynomial in λ. The strong Diffie-Hellman assumption [4] states that for any PPT $\mathcal{A}$, the following advantage is negligible in λ: $\mathbf{Adv}^{\mathrm{sdh}}_{\mathtt{BlGrp},\mathcal{S}}(\lambda, \mathcal{U}) := \Pr[\mathrm{params} \leftarrow \mathtt{BlGrp}(1^\lambda), \alpha \overset{\$}{\leftarrow} \mathbb{Z}_p,\ (u, e) \leftarrow \mathcal{A}(\mathrm{params}, (g, g^\alpha, g^{\alpha^2}, \dots, g^{\alpha^q}, h, h^\alpha)) : u^{\alpha+e} = g]$.

3 Syntax of Attribute-Based Two-Tier Signature Scheme

In this section, we propose a definition of syntax of an attribute-based two-tier signature scheme (ABTTS). Then, we define a chosen-message attack (CMA) on ABTTS where an adversary makes an existential forgery, and we define the existential unforgeability (EUF) against CMA.

3.1 Definition: Syntax of ABTTS

An attribute-based two-tier signature scheme, $\mathtt{ABTTS}$, consists of five PPTs: $\mathtt{ABTTS}=$ ($\mathbf{ABTTS}.\mathbf{Setup}$, $\mathbf{ABTTS}.\mathbf{KG}$, $\mathbf{ABTTS}.\mathbf{SKG}$, $\mathbf{ABTTS}.\mathbf{Sign}$, $\mathbf{ABTTS}.\mathbf{Vrfy}$).

$\mathbf{ABTTS}.\mathbf{Setup}(1^\lambda,\mathcal{U}) \to (\mathrm{MSK}, \mathrm{PK})$. Given the security parameter 1^λ and the attribute universe $\mathcal{U}$, it returns a master secret key MSK and a public key PK.

$\mathbf{ABTTS}.\mathbf{KG}(\mathrm{MSK}, \mathrm{PK}, S) \to \mathrm{SK}_S$. Given the master secret key MSK, the public key PK and an attribute set $S \subset \mathcal{U}$, it returns a secret key SK_S that corresponds to S.

$\mathbf{ABTTS}.\mathbf{SKG}(\mathrm{MSK}, \mathrm{PK}, \mathrm{SK}_S, f) \to (\mathrm{SSK}_{S,f}, \mathrm{SPK}_f)$. Given the master secret key MSK, the public key PK, a secret key SK_S and an access formula f, it returns a pair $(\mathrm{SSK}_{S,f}, \mathrm{SPK}_f)$ of a secondary secret key and a secondary public key.

$\mathbf{ABTTS}.\mathbf{Sign}(\mathrm{PK}, \mathrm{SK}_S, \mathrm{SSK}_{S,f}, \mathrm{SPK}_f, (m, f)) \to \sigma$. Given the public key PK, a secret key SK_S, a secondary secret key $\mathrm{SSK}_{S,f}$, a secondary public key SPK_f and a pair (m, f) of a message $m \in \{1, 0\}^*$ and an access formula f, it returns a signature σ.

$\mathbf{ABTTS}.\mathbf{Vrfy}(\mathrm{PK}, \mathrm{SPK}_f, (m, f), \sigma) \to 1/0$. Given the public key PK, a secondary public key SPK_f, a pair (m, f) of a message and an access formula and a signature σ, it returns a decision 1 or 0. When it is 1, we say that $((m, f), \sigma)$ is *valid*. When it is 0, we say that $((m, f), \sigma)$ is *invalid*.

We demand correctness of $\mathtt{ABTTS}$ that, for any λ, any $\mathcal{U}$, any $S \subset \mathcal{U}$ and any (m, f) such that $f(S) = 1$, $\Pr[(\mathrm{MSK}, \mathrm{PK}) \leftarrow \mathbf{ABTTS}.\mathbf{Setup}(1^\lambda,\mathcal{U})$, $\mathrm{SK}_S \leftarrow \mathbf{ABTTS}.\mathbf{KG}(\mathrm{MSK}, \mathrm{PK}, S)$, $(\mathrm{SSK}_{S,f}, \mathrm{SPK}_f)$ $\leftarrow \mathbf{ABTTS}.\mathbf{SKG}(\mathrm{MSK}, \mathrm{PK}, \mathrm{SK}_S, f)$, $\sigma \leftarrow \mathbf{ABTTS}.\mathbf{Sign}(\mathrm{SK}_S, \mathrm{PK}, \mathrm{SSK}_{S,f}, \mathrm{SPK}_f, (m, f))$, $b \leftarrow \mathbf{ABS}.\mathbf{Vrfy}(\mathrm{PK}, \mathrm{SPK}_f, (m, f), \sigma) : b = 1] = 1$.

3.2 Security against Chosen-Message Attacks on ABTTS

A PPT adversary $\mathcal{F}$ tries to make a forgery $((m^*, f^*), \sigma^*)$ that consists of a message, a target access formula and a signature. The following experiment $\mathbf{Expr}^{\text{euf-cma}}_{\mathtt{ABTTS},\mathcal{F}}(1^\lambda,\mathcal{U})$ of a forger $\mathcal{F}$ defines the *chosen-message attack making an existential forgery*.

```
$\mathbf{Expr}^{\text{euf-cma}}_{\mathtt{ABTTS},\mathcal{F}}(1^\lambda,\mathcal{U})$ :
  $(\mathrm{PK}, \mathrm{MSK}) \leftarrow \mathbf{ABTTS}.\mathbf{Setup}(1^\lambda,\mathcal{U})$
  $((m^*, f^*), \sigma^*) \leftarrow \mathcal{F}^{\mathcal{ABTTSKG},\mathcal{ABTTSSPK},\mathcal{ABTTSSIGN}}(\mathrm{PK})$
  If $\mathbf{ABTTS}.\mathbf{Vrfy}(\mathrm{PK}, \mathrm{SPK}_f, (m^*, f^*), \sigma^*) = 1$
    then Return Win else Return Lose
```

In the experiment, $\mathcal{F}$ issues key-extraction queries to its oracle $\mathcal{ABTTSKG}$, secondary public key queries to its oracle $\mathcal{ABTTSSPK}$ and signing queries to its oracle $\mathcal{ABTTSSIGN}$. Giving an attribute set S_i, $\mathcal{F}$ queries $\mathcal{ABTTSKG}(\mathrm{MSK}, \mathrm{PK}, \cdot)$ for a secret key SK_{S_i}. Giving an attribute set S and an access formula f, $\mathcal{F}$ queries $\mathcal{ABTTSSPK}(\mathrm{MSK}, \mathrm{PK}, \mathrm{SK}_{\cdot}, \cdot)$ for a secondary public key SPK_f. Giving an attribute set S_j and a pair (m_j, f_j) of a message and an access formula, $\mathcal{F}$ queries $\mathcal{ABTTSSIGN}(\mathrm{PK}, \mathrm{SK}_{\cdot}, \mathrm{SSK}_{\cdot,\cdot}, \mathrm{SPK}_{\cdot}, (\cdot, \cdot))$ for a valid signature σ when $f(S_j) = 1$. As a rule of the two-tier signature, each published secondary public key SPK_f can be used only once to obtain a signature [3].

f^* is called a *target access formula* of $\mathcal{F}$. Here we consider the *adaptive target* case in the sense that $\mathcal{F}$ is allowed to choose f^* after seeing PK and issuing three queries. Two restrictions are imposed on $\mathcal{F}$: 1) $f^*(S_i) = 0$ for all S_i in key-extraction queries; 2) (m^*, f^*) was never queried in signing queries. The numbers of key-extraction queries and signing queries are at most q_{ke} and q_{sig}, respectively, which are bounded by a polynomial in λ. The *advantage* of $\mathcal{F}$ over $\mathtt{ABTTS}$ is defined as $\mathbf{Adv}^{\text{euf-cma}}_{\mathtt{ABTTS},\mathcal{F}}(\lambda,\mathcal{U}) \overset{\text{def}}{=} \Pr[\mathbf{Expr}^{\text{euf-cma}}_{\mathtt{ABTTS},\mathcal{F}}(1^\lambda,\mathcal{U}) \text{ returns } \textsc{Win}]$.

Definition 2 (EUF-CMA of ABTTS) *ABTTS is called existentially unforgeable against chosen-message attacks if, for any PPT $\mathcal{F}$ and any $\mathcal{U}$, $\mathbf{Adv}^{\textit{euf-cma}}_{\textit{ABTTS},\mathcal{F}}(\lambda,\mathcal{U})$ is negligible in λ.*

Then we define *attribute privacy* of ABTTS.

Definition 3 (Attribute Privacy of ABTTS) *ABTTS is called to have attribute privacy if, for all $(PK, MSK) \leftarrow \mathbf{ABS}.\mathbf{Setup}(1^\lambda,\mathcal{U})$, for all message m, for all attribute sets S_1 and S_2, for all signing keys $SK_{S_1} \leftarrow \mathbf{ABS}.\mathbf{KG}(PK, MSK, S_1)$ and $SK_{S_2} \leftarrow \mathbf{ABS}.\mathbf{KG}(PK, MSK, S_2)$, for all secondary keys $(SSK_{S_1,f}, SPK_f) \leftarrow \mathbf{ABTTS}.\mathbf{SKG}(MSK, PK, SK_S, f)$ and $(SSK_{S_2,f}, SPK_f) \leftarrow \mathbf{ABTTS}.\mathbf{SKG}(MSK, PK, SK_S, f)$ and for all access formula f such that $[f(S_1) = 1 \wedge f(S_2) = 1] \vee [f(S_1) \neq 1 \wedge f(S_2) \neq 1]$, two distributions*
$\sigma_1 \leftarrow \mathbf{ABTTS}.\mathbf{Sign}(PK, SK_{S_1}, SSK_{S_1,f}, SPK_f, (m, f))$ and
$\sigma_2 \leftarrow \mathbf{ABTTS}.\mathbf{Sign}(PK, SK_{S_2}, SSK_{S_2,f}, SPK_f, (m, f))$ are identical.

4 $\boldsymbol{\Sigma}$-protocol for Monotone Access Formula

In this section, we enhance the identification protocol by Okamoto [15] to the boolean proof system $\mathbf{\Sigma}_f$ proposed by Anada et al. [2].

4.1 Our Language L_f

We assume R to be an NP-relation. Let $R(\cdot,\cdot) : (\{1,0\}^*)^2 \to \{1,0\}$ denote the relation-function which returns $(x,w) \overset{?}{\in} R$. Let $f = f((X_{i_j})_{j=1}^a)$ be a boolean formula over boolean variables $\{X_i\}_i$.

Definition 4 (Language for f) *The relation R_f and the corresponding language L_f for a boolean formula f are:*

$$R_f \overset{def}{=} \{\big(x = (x_{i_j})_{j=1}^a, w = (w_{i_j})_{j=1}^a\big) \in \{1,0\}^* \times \{1,0\}^*; f(R(x_{i_j}, w_{i_j})_{j=1}^a) = 1\},$$
$$L_f \overset{def}{=} \{x \in \{1,0\}^*; \exists w, (x,w) \in R_f\}.$$

We consider hereafter the case that w is divided into $(w_{i_j})_{j=1}^a = (e_{i_j}, s_{i_j})_{j=1}^a$. (In/after the next section, we will consider the special case that $e_{i_j}, j = 1,\dots,a$, are all equal to a single element e. The common component e will be a tag τ of a signature bundle.)

4.2 Our Σ-protocol $\mathbf{\Sigma}_f$ for L_f

Our Σ-protocol $\mathbf{\Sigma}_f$ is a zero-knowledge proof-of-knowledge $\mathbf{ZKPK}[w := (w_{\rho(l)})_l := (e_{\rho(l)}, s_{\rho(l)})_l, l \in \mathrm{Leaf}(\mathcal{T}_f) : x := \text{(equations)}]$ for the language L_f, where the equations are for all the leaf nodes:

$$Z_{\rho(l)} = Z_{\rho(l),1}^{e_{\rho(l)}} Z_{\rho(l),2}^{s_{\rho(l)}},\ l \in \mathrm{Leaf}(\mathcal{T}_f). \tag{1}$$

In the above equation, $Z_{\rho(l)}$ is represented by $(e_{\rho(l)}, s_{\rho(l)})$ to the base $(Z_{\rho(l),1}, Z_{\rho(l),2})$. A prover $\mathcal{P}(x,w,f)$ and a verifier $\mathcal{V}(x,f)$ execute our Σ-protocol in the following way.
$\underline{\mathcal{P}(x,w,f)}$. To prove the knowledge of those representations $(e_{\rho(l)}, s_{\rho(l)})$, $\mathcal{P}$ computes the first message, a commitment $(\textsc{Cmt}_l)_l$, as follows. Let $\bar{\mathbb{Z}}$ be the exponent domain for the above expression. To do the computation honestly at a leaf l, $\mathcal{P}$ chooses $\eta_{e,l}, \eta_{s,n} \overset{\$}{\leftarrow} \bar{\mathbb{Z}}$, and puts $\textsc{Cmt}_l := Z_{\rho(l),1}^{\eta_{e,l}} Z_{\rho(l),2}^{\eta_{s,n}}$. To simulate the computation at a leaf l, $\mathcal{P}$ chooses $\eta_{e,l}, \theta_{s,l} \overset{\$}{\leftarrow} \bar{\mathbb{Z}}$, and in addition, $(c_n)_n, c_n \in \bar{\mathbb{Z}}$. Here $(c_n)_n$ are chosen in accordance with the so called boolean proof system of Anada et al. [2]. Then $\mathcal{P}$ puts for each leaf l $\theta_{e,l} := \eta_{e,l} + c_l e_{\rho(l)}$, and $\textsc{Cmt}_l := Z_{\rho(l)}^{-c_l} Z_{\rho(l),1}^{\theta_{e,l}} Z_{\rho(l),2}^{\theta_{s,l}}$. $\mathcal{P}$ sends $(\textsc{Cmt}_l)_l$ to a verifier $\mathcal{V}$.
$\underline{\mathcal{V}(x,f)}$. Receiving $(\textsc{Cmt}_l)_l$, $\mathcal{V}(x,f)$ chooses the second message: a challenge $\textsc{Cha} \overset{\$}{\leftarrow} \bar{\mathbb{Z}}$, uniformly at random, and sends $\textsc{Cha}$ to $\mathcal{P}$.

$\underline{\mathcal{P}(x, w, f)}$. Receiving $\textsc{Cha}$, $\mathcal{P}$ completes to compute the third message; that is, $\mathcal{P}$ completes the division $(\textsc{Cha}_n := c_n)_n$ such that $\textsc{Cha}_{r(\mathcal{T}_f)} = \textsc{Cha}$, and a response $(\textsc{Res}_l := (\theta_{e,l}, \theta_{s,l}))_l$ with $\theta_{e,l} := \eta_{e,l} + c_l e_{\rho(l)}$, $\theta_{s,l} := \eta_{s,l} + c_l v_l$. $\mathcal{P}$ sends $(\textsc{Cha}_l)_l$ and $(\textsc{Res}_l)_l$ to $\mathcal{V}$.
$\underline{\mathcal{V}(x, f)}$. Receiving $(\textsc{Cha}_l)_l$ and $(\textsc{Res}_l)_l$, $\mathcal{V}$ checks the integrity of the division $(\textsc{Cha}_l)_l$. Then $\mathcal{V}$ verifies:

$$\textsc{Cmt}_l \overset{?}{=} Z_{\rho(l)}^{-c_l} Z_{\rho(l),1}^{\theta_{e,l}} Z_{\rho(l),2}^{\theta_{s,l}}, \ l \in \mathrm{Leaf}(\mathcal{T}_f). \tag{2}$$

According to the division rule of Anada et al. [2], the integrity of $(\textsc{Cha}_l = c_l)_l$ can be checked as follows: From the leaves to the root, and at every inner node $n \in \mathrm{iNode}(\mathcal{T}_f)$ as well as its two children ch_1, ch_2;

- If n is an AND node $(\wedge)$, then verify $c_{ch_1} \overset{?}{=} c_{ch_2}$. If so, put $c_n := c_{ch_1}$.
- Else if n is an OR node $(\vee)$, then just put $c_n := c_{ch_1} + c_{ch_2}$.
- If n is the root node, then verify $c_n \overset{?}{=} \textsc{Cha}$.
- Repeat until all $n \in \mathrm{iNode}(\mathcal{T}_f)$ are verified.

Our $\boldsymbol{\Sigma}_f$ can be shown to possess the three requirements of Σ-protocol: completeness, special soundness and honest-verifier zero-knowledge.

5 Signature Bundle Scheme in RSA

In this section, we propose a signature bundle scheme in the RSA setting by extending the Camenisch-Lysyanskaya signature scheme [5]. We first construct the scheme. Then we discuss its EUF-CMA security. (The scheme in the discrete logarithm setting is proposed in Appendix A.)

5.1 Construction of Our SB in RSA

Our signature bundle scheme $\mathtt{SB} = (\mathbf{SB}.\mathbf{KG}, \mathbf{SB}.\mathbf{Sign}, \mathbf{SB}.\mathbf{Vrfy})$ is described as follows. Let $l_{\mathcal{M}}$ be a parameter. The message space $\mathcal{M}$ consists of all binary strings of length $l_{\mathcal{M}}$. Let $n = n(\lambda)$ denote the maximum number of messages made into a bundle, which is a polynomial in λ.
$\mathbf{SB}.\mathbf{KG}(1^\lambda) \to (\mathrm{PK}, \mathrm{SK})$. Given 1^λ, it chooses a special RSA modulus $N = pq$ of length $l_N = \lambda$, where $p = 2p' + 1$ and $q = 2q' + 1$ are safe primes. For $i = 1$ to n, it chooses $g_{i,0}, g_{i,1}, g_{i,2} \overset{\$}{\leftarrow} QR_N$. It puts $\mathrm{PK} := (N, (g_{i,0}, g_{i,1}, g_{i,2})_{i=1}^n)$ and $\mathrm{SK} = p$, and returns $(\mathrm{PK}, \mathrm{SK})$.
$\mathbf{SB}.\mathbf{Sign}(\mathrm{PK}, \mathrm{SK}, (m_i)_{i=1}^n) \to (\tau, (\sigma_i)_{i=1}^n)$. Given PK, SK and messages $(m_i)_{i=1}^n$ each of which is of length $l_{\mathcal{M}}$, it chooses a prime e of length

$l_e = l_{\mathcal{M}} + 2$ at random. For $i = 1$ to n, it chooses an integer s_i of length $l_s = l_N + l_{\mathcal{M}} + l$ at random, where l is a security parameter, and it computes the value A_i:

$$A_i := (g_{i,0} g_{i,1}^{m_i} g_{i,2}^{s_i})^{\frac{1}{e}}. \qquad (3)$$

It puts $\tau = e$ and $\sigma_i = (s_i, A_i)$ for each i and returns $(\tau, (\sigma_i)_{i=1}^n)$.
SB.**Vrfy**$(\mathrm{PK}, (m_i)_{i=1}^n, (\tau, (\sigma_i)_{i=1}^n)) \to 1/0$. Given PK, $(m_i)_{i=1}^n$ and a signature bundle $(\tau, (\sigma_i)_{i=1}^n)$, it verifies whether the following holds: $e := \tau$ is of length l_e and for $i = 1$ to n: $A_i^e = g_{i,0} g_{i,1}^{m_i} g_{i,2}^{s_i}$.

5.2 Security of Our SB in RSA

Theorem 1 (EUF-CMA of Our SB in RSA) *Our signature bundle scheme SB is existentially unforgeable against chosen-message attacks under the Strong RSA assumption.*

6 Attribute-Based ID Scheme in RSA

In this section, we combine two building blocks to obtain our attribute-based identification scheme; that is, the Σ-protocol $\mathbf{\Sigma}_f$ in Section 4.2 and the signature bundle scheme SB in Section 5.1.

6.1 Construction of Our ABID in RSA

ABID.**Setup**$(1^\lambda, \mathcal{U}) \to (\mathrm{MSK}, \mathrm{PK})$. Given the security parameter 1^λ and an attribute universe $\mathcal{U}$, it chooses a special RSA modulus $N = pq, p = 2p' + 1, q = 2q' + 1$ of length $l_N = 2\lambda$. For $\mathsf{at} \in \mathcal{U}$, it chooses $g_{\mathsf{at},0}, g_{\mathsf{at},1}, g_{\mathsf{at},2} \overset{\$}{\leftarrow} QR_N$ and a hash key $\mu \overset{\$}{\leftarrow} \mathit{Hashkeysp}(1^\lambda)$ of a hash function Hash_μ with the value in $\mathbb{Z}_p$. It puts $\mathrm{PK} := (N, (g_{\mathsf{at},0}, g_{\mathsf{at},1}, g_{\mathsf{at},2})_{\mathsf{at} \in \mathcal{U}}, \mu, \mathcal{U})$ and $\mathrm{MSK} := p$. It returns PK and MSK.
ABID.**KG**$(\mathrm{MSK}, \mathrm{PK}, S) \to \mathrm{SK}_S$. Given PK, MSK and an attribute subset S, it chooses a prime e of length l_e. For $\mathsf{at} \in S$, it computes $a_{\mathsf{at}} \leftarrow \mathit{Hash}_\mu(\mathsf{at})$, $s_{\mathsf{at}} \overset{\$}{\leftarrow} \mathbb{Z}$ of length l_e, $A_{\mathsf{at}} := (g_0 g_1^{a_{\mathsf{at}}} g_2^{s_{\mathsf{at}}})^{\frac{1}{e}}$. It puts $\mathrm{SK}_S := (e, (s_{\mathsf{at}}, A_{\mathsf{at}})_{\mathsf{at} \in S})$.
$\mathcal{P}(\mathrm{SK}_S, \mathrm{PK}, f)$ and $\mathcal{V}(\mathrm{PK}, f)$ execute $\mathbf{\Sigma}_f$ with the following precomputation. For $\mathsf{at} \in \mathrm{At}(f)$, $\mathcal{P}$ chooses $r_{\mathsf{at}} \overset{\$}{\leftarrow} \mathbb{Z}$ of length l_e. If $\mathsf{at} \in S$ then $s'_{\mathsf{at}} := s_{\mathsf{at}} + e r_{\mathsf{at}}, A'_{\mathsf{at}} := A_{\mathsf{at}} g_2^{-r_{\mathsf{at}}}$. Else $s'_{\mathsf{at}} \overset{\$}{\leftarrow} \mathbb{Z}$ of length l_e, $A'_{\mathsf{at}} \overset{\$}{\leftarrow} \mathbb{Z}_N^*$. $\mathcal{P}$ puts $Z_{\mathsf{at}} := g_{\mathsf{at},0} g_{\mathsf{at},1}^{a_{\mathsf{at}}}, Z_{\mathsf{at},1} := A'_{\mathsf{at}}, Z_{\mathsf{at},2} := g_{\mathsf{at},2}$. Then the statement for $\mathbf{\Sigma}_f$ is $x := (x_{\mathsf{at}} := (Z_{\mathsf{at}}, Z_{\mathsf{at},1}, Z_{\mathsf{at},2}))_{\mathsf{at}}$ and the witness is

$w := (\tau := e, (w_{\mathtt{at}} := s'_{\mathtt{at}})_{\mathtt{at}})$, where $\mathtt{at} \in \mathrm{At}(f)$ for x and w. $\mathcal{P}$ sends the randomized values $(A'_{\mathtt{at}})_{\mathtt{at}}$ to $\mathcal{V}$ for $\mathcal{V}$ to be able to compute the statement x.

After the above precomputation, $\mathcal{P}$ and $\mathcal{V}$ can execute $\mathbf{\Sigma}_f$ for the language L_f. In other words, $\mathcal{P}$ and $\mathcal{V}$ execute $\mathbf{ZKPK}[(e, s'_{\rho(l)})_l, l \in \mathrm{Leaf}(\mathcal{T}_f)$: equations], for the language L_f, where equations are: $Z_{\rho(l)} = Z^{e}_{\rho(l),1} Z^{s'_{\rho(l)}}_{\rho(l),2}, l \in \mathrm{Leaf}(\mathcal{T}_f)$. Note that $\mathcal{V}$ verifies whether the following verification equations hold or not for all the leaf nodes:

$$\textsc{Cmt}_l \overset{?}{=} Z^{-c_l}_{\rho(l)} Z^{\theta_{e,l}}_{\rho(l),1} Z^{\theta_{s',l}}_{\rho(l),2}, l \in \mathrm{Leaf}(\mathcal{T}_f). \tag{4}$$

$\mathcal{V}$ returns 1 or 0 accordingly.

6.2 Security of Our ABID in RSA

Claim 1 (Concurrent Security of Our ABID under a Single Tag) *Our ABID is secure against concurrent attacks if our signature bundle scheme SB is existentially unforgeable against chosen-message attacks and if the extracted values e by the extractor of the underlying Σ-protocol $\mathbf{\Sigma}_f$ is a single value.*

Note that Claim 1 is needed only as intermediate result. That is, the assumption that the extracted value e is a single value is assured by the two-tier keys issuer, **ABTTS**.**SKG**, in the next section.

7 Attribute-Based Two-Tier Signature Scheme in RSA

In this section, we construct our ABTTS concretely. By applying the method of two-tier keys to our ABID in the last section, we attain the ABTTS scheme. Our ABTTS enjoys EUF-CMA, collusion resistance and attribute privacy, in the standard model.

The critical point is that the secondary key generator **ABTTS**.**SKG** can issue a legitimate statement x for the boolean proof system $\mathbf{\Sigma}_f$. Hence our ABTTS can avoid collusion attacks on secret keys.

7.1 Construction of Our ABTTS in RSA

ABTTS.**Setup**$(1^\lambda, \mathcal{U}) \to (\mathrm{MSK}, \mathrm{PK})$. Given the security parameter 1^λ and an attribute universe $\mathcal{U}$, it chooses a special RSA modulus $N = pq, p = 2p' + 1, q = 2q' + 1$ of length $l_N = 2\lambda$. For $\mathtt{at} \in \mathcal{U}$, it chooses

$g_{\mathsf{at},0}, g_{\mathsf{at},1}, g_{\mathsf{at},2} \xleftarrow{\$} QR_N$ and a hash key $\mu \xleftarrow{\$} \mathit{Hashkeysp}(1^\lambda)$ of a hash function Hash_μ with the value in $\mathbb{Z}_p$. It puts $\mathrm{PK} := (N, (g_{\mathsf{at},0}, g_{\mathsf{at},1}, g_{\mathsf{at},2})_{\mathsf{at}\in\mathcal{U}}, \mu, \mathcal{U})$ and $\mathrm{MSK} := p$. It returns PK and MSK.

ABTTS.KG$(\mathrm{MSK}, \mathrm{PK}, S) \to \mathrm{SK}_S$. Given PK, MSK and an attribute subset S, it chooses a prime e of length l_e. For $\mathsf{at} \in S$, it computes $a_{\mathsf{at}} \leftarrow \mathit{Hash}_\mu(\mathsf{at})$, $s_{\mathsf{at}} \xleftarrow{\$} \mathbb{Z}$ of length l_e, $A_{\mathsf{at}} := (g_0 g_1^{a_{\mathsf{at}}} g_2^{s_{\mathsf{at}}})^{\frac{1}{e}}$. It puts $\mathrm{SK}_S := (e, (s_{\mathsf{at}}, A_{\mathsf{at}})_{\mathsf{at}\in S})$ and returns SK_S.

ABTTS.SKG$(\mathrm{MSK}, \mathrm{PK}, \mathrm{SK}_S, f) \to (\mathrm{SSK}_{S,f}, \mathrm{SPK}_f)$. Given MSK, PK, the secret key SK_S and an access formula f, it first checks whether the components $e_{\rho(l)}$ in SK_S, $\rho(l) \in S$, are equal to a single value e or not. If it is false, then it aborts. Then it computes the statement for $\mathbf{\Sigma}_f$, $x := (x_{\mathsf{at}} := (Z_{\mathsf{at}}, Z_{\mathsf{at},1}, Z_{\mathsf{at},2}))_{\mathsf{at}}$, and the witness $w := (\tau := e, (w_{\mathsf{at}} := s'_{\mathsf{at}})_{\mathsf{at}})$, where $\mathsf{at} \in \mathrm{At}(f)$ for x and w. Then it runs the prover $\mathcal{P}$ according to $\mathbf{\Sigma}_f$ as $((\textsc{Cmt}_l)_l, st) \leftarrow \mathcal{P}(\mathrm{SK}_S, \mathrm{PK}, f)$. Then it puts $\mathrm{SSK}_{S,f} := (w, \textsc{Cmt} \parallel st)$ and $\mathrm{SPK}_f := (x, \textsc{Cmt})$. It returns $\mathrm{SSK}_{S,f}$ and SPK_f.

ABTTS.Sign$(\mathrm{PK}, \mathrm{SK}_S, \mathrm{SSK}_{S,f}, \mathrm{SPK}_f, (m, f)) \to \sigma$. Given PK, SK_S, the secondary secret key $\mathrm{SSK}_{S,f}$, the secondary public key SPK_f, and a pair (m, f) of a message in $\{1,0\}^{l_{\mathcal{M}}}$ and an access formula f, it computes $\textsc{Cha} \leftarrow \mathit{Hash}_\mu((A'_{\mathsf{at}})_{\mathsf{at}} \parallel (\textsc{Cmt}_l)_l \parallel m)$. Then, it runs the prover $\mathcal{P}$ according to $\mathbf{\Sigma}_f$ as $((\textsc{Cha}_l)_l, (\textsc{Res}_l)_l \leftarrow \mathcal{P}((\textsc{Cmt}_l)_l \parallel \textsc{Cha} \parallel, st)$. Finally, it returns the signature $\sigma := ((A'_{\mathsf{at}})_{\mathsf{at}}, (\textsc{Cmt}_l)_l, (\textsc{Cha}_l)_l, (\textsc{Res}_l)_l)$.

ABTTS.Vrfy$(\mathrm{PK}, \mathrm{SPK}_f, (m, f), \sigma) \to 1/0$. Given PK, the secondary public key SPK_f, a pair (m, f) and a signature σ, it first computes the statement for $\mathbf{\Sigma}_f$, $x := (x_{\mathsf{at}} := (Z_{\mathsf{at}}, Z_{\mathsf{at},1}, Z_{\mathsf{at},2}))_{\mathsf{at}}$, and the witness $w := (\tau := e, (w_{\mathsf{at}} := s'_{\mathsf{at}})_{\mathsf{at}})$, where $\mathsf{at} \in \mathrm{At}(f)$ for x and w. Then it computes $\textsc{Cha} \leftarrow \mathit{Hash}_\mu((A'_{\mathsf{at}})_{\mathsf{at}} \parallel (\textsc{Cmt}_l)_l \parallel m)$. Then, it runs the verifier $\mathcal{V}$ according to $\mathbf{\Sigma}_f$ as acc or $0 \leftarrow \mathcal{V}(\mathrm{PK}, f, (\textsc{Cmt}_l)_l \parallel \textsc{Cha} \parallel (\textsc{Res}_l)_l)$. It returns 1 or 0 accordingly.

7.2 Security of Our ABTTS in RSA

Theorem 2 (EUF-CMA of Our ABTTS in RSA) *Our attribute-based two-tier signature scheme ABTTS is existentially unforgeable against chosen-message attacks under the Strong RSA assumption in the standard model.*

Theorem 3 (Attribute Privacy of Our ABTTS in RSA) *Our attribute-based two-tier signature scheme ABTTS has attribute privacy.*

8 Conclusions

We defined the attribute-based two-tier signature scheme (ABTTS). Then we provided ABTTS concretely that enjoys EUF-CMA, collusion resistance and attribute privacy, in the standard model.

Acknowledgements Concerning the first and the second authors, this work is partially supported by Grants-in-Aid for Scientific Research; Research Project Number:15K00029.

References

1. H. Anada, S. Arita, S. Handa, and Y. Iwabuchi. Attribute-based identification: Definitions and efficient constructions. In *ACISP 2013*, volume 7959 of *LNCS*, pages 168–186. Springer, 2013.
2. H. Anada, S. Arita, and K. Sakurai. Attribute-based signatures without pairings via the fiat-shamir paradigm. In *ASIAPKC2014*, volume 2 of *ACM-ASIAPKC*, pages 49–58. ACM, 2014.
3. M. Bellare and S. Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In *Public Key Cryptography - PKC 2007, 10th International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, April 16-20, 2007, Proceedings*, pages 201–216, 2007.
4. D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004.
5. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *Security in Communication Networks, Third International Conference, SCN 2002, Amalfi, Italy, September 11-13, 2002. Revised Papers*, pages 268–289, 2002.
6. R. Cramer. *Modular Designs of Secure, yet Practical Cyptographic Protocols*. PhD thesis, University of Amsterdam, Amsterdam, the Netherlands, 1996.
7. I. Damgård. On σ-protocols. In Course Notes, https://services.brics.dk/java/courseadmin/CPT/documents, 2011.
8. A. El Kaafarani, L. Chen, E. Ghadafi, and J. H. Davenport. Attribute-based signatures with user-controlled linkability. In *Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings*, pages 256–269, 2014.
9. I. E. T. Force. Request for comments: 6960. http://tools.ietf.org/html/rfc6960.
10. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM-CCS '06*, volume 263, pages 89–98. ACM, 2006.
11. R. Granger, T. Kleinjung, and J. Zumbrägel. Breaking '128-bit secure' supersingular binary curves - (or how to solve discrete logarithms in f_{2}4 1223 and f_{2}12 367). In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 126–145, 2014.
12. S. Guo and Y. Zeng. Attribute-based signature scheme. In *ISA '08*, pages 509–511. IEEE, 2008.

13. J. Herranz. Attribute-based signatures from rsa. *Theoretical Computer Science*, 527:73–82, 2014.
14. H. K. Maji, M. Prabhakaran, and M. Rosulek. Attribute-based signatures. In *CT-RSA 2011*, volume 6558 of *LNCS*, pages 376–392. Springer, 2011. Full version available at IACR Cryptology ePrint Archive, 2010/595, http://eprint.iacr.org/.
15. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, pages 31–53, 1992.
16. T. Okamoto and K. Takashima. Efficient attribute-based signatures for non-monotone predicates in the Standard Model. In *PKC 2011*, volume 6571 of *LNCS*, pages 35–52. Springer, 2011.
17. M. Yasuda, T. Shimoyama, J. Kogure, and T. Izu. On the strength comparison of the ECDLP and the IFP. In *Security and Cryptography for Networks - 8th International Conference, SCN 2012, Amalfi, Italy, September 5-7, 2012. Proceedings*, pages 302–325, 2012.

A Signature Bundle Scheme in Discrete Log

Our pairing-based Signature Bundle Scheme, $\mathtt{SB} = (\mathbf{SB}.\mathbf{KG}, \mathbf{SB}.\mathbf{Sign}, \mathbf{SB}.\mathbf{Vrfy})$, is described as follows.
$\mathbf{SB}.\mathbf{KG}(1^\lambda) \to (\mathrm{PK}, \mathrm{SK})$. Given 1^λ, it executes a group generator $\mathtt{BlGrp}(1^\lambda)$ to get $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot,\cdot))$. For $i = 1$ to n, it chooses $g_{i,0}, g_{i,1}, g_{i,2} \overset{\$}{\leftarrow} \mathbb{G}_1, h_0 \overset{\$}{\leftarrow} \mathbb{G}_2, \alpha \overset{\$}{\leftarrow} \mathbb{Z}_p$ and it puts $h_1 := h_0^\alpha$. It puts $\mathrm{PK} := ((g_{i,0}, g_{i,1}, g_{i,2})_{i=1}^n, h_0, h_1)$ and $\mathrm{SK} := \alpha$, and returns $(\mathrm{PK}, \mathrm{SK})$.
$\mathbf{SB}.\mathbf{Sign}(\mathrm{PK}, \mathrm{SK}, (m_i)_{i=1}^n) \to (\tau, (\sigma_i)_{i=1}^n)$. Given PK, SK and messages $(m_i)_{i=1}^n$ each of which is of length $l_{\mathcal{M}}$, it chooses $e \overset{\$}{\leftarrow} \mathbb{Z}_p$. For $i = 1$ to n, it chooses $s_i \overset{\$}{\leftarrow} \mathbb{Z}_p$, and it computes the value A_i:

$$A_i := (g_{i,0} g_{i,1}^{m_i} g_{i,2}^{s_i})^{\frac{1}{\alpha+e}}. \tag{5}$$

It puts $\tau = e$ and $\sigma_i = (s_i, A_i)$ for each i and returns $(\tau, (\sigma_i)_{i=1}^n)$.
$\mathbf{SB}.\mathbf{Vrfy}(\mathrm{PK}, (m_i)_{i=1}^n, (\tau, (\sigma_i)_{i=1}^n)) \to 1/0$. Given PK, $(m_i)_{i=1}^n$ and $(\tau, (\sigma_i)_{i=1}^n)$, it verifies whether the following holds: $e(A_i, h_0^e h_1) = e(g_{i,0} g_{i,1}^{m_i} g_{i,2}^{s_i}, h_0),\ i = 1, \ldots, n$.

Theorem 4 (EUF-CMA of Our SB in Discrete Log) *Our signature bundle scheme SB is existentially unforgeable against chosen-message attack under the Strong Diffie-Hellman assumption.*

Our ABID and ABTTS in the discrete logarithm setting will be given in the full version.