Forward Secure Attribute-Based Signatures

Tsz Hon Yuen[1], Joseph K. Liu[2], Xinyi Huang[3,⋆], Man Ho Au[4], Willy Susilo[4], and Jianying Zhou[2]

[1] University of Hong Kong, Hong Kong
thyuen@cs.hku.hk
[2] Institute for Infocomm Research, Singapore
{ksliu,jyzhou}@i2r.a-star.edu.sg
[3] School of Mathematics and Computer Science,
Fujian Normal University, China
xyhuang81@gmail.com
[4] School of Computer Science and Software Engineering,
University of Wollongong, Australia
{aau,wsusilo}@uow.edu.au

Abstract. Attribute-Based Signatures (ABS) is a versatile primitive which allows an entity to sign a message with fine-grained control over identifying information. A valid ABS only attests to the fact that "A single user, whose attributes satisfy the predicate, has endorsed the message". While ABS has been well investigated since its introduction, it is unfortunate that key exposure–an inherent weakness of digital signatures–has never been formally studied in the scenario of ABS. We fill this gap by proposing a new notion called forward secure ABS, its formal security models and a generic (also the first) design based on well established crypto primitives.

1 Introduction

Attribute-Based Signatures [13,15] (or, ABS for short) is a primitive proposed to provide signer anonymity. An ABS allows an entity to sign a message with fine-grained control over identifying information. A valid ABS signature attests to the fact that "A single user, whose attributes satisfy the predicate, has endorsed the message". Ring signatures [18,5,19] and group signatures [9,3,6] are comparable to special cases of ABS, in which the only allowed predicates are disjunctions over the universe of attributes (identities). In ABS, each entity possesses a set of attributes and a key-authority generates the associated private keys, with which one can sign a message with a predicate satisfied by his/her attributes. The signature reveals no more than the fact that a single user with some set of attributes satisfying the predicate has attested to the message. In particular, ABS does not provide any information on the particular set of attributes used to satisfy the predicate. For example, an "(Engineer, Department A)" or an "(Engineer, Department B)" can independently generate an ABS to assure the recipient that the signature was

⋆ Corresponding author.

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 167–177, 2012.

produced by an "Engineer" without disclosing the department information. Furthermore, users of ABS cannot collude to pool their attributes together (which separates ABS from mesh signatures): It is never possible for an "(Engineer, Department A)" and an "(Auditor, Department B)" to collude and generate an ABS satisfying the predicate "(Auditor, Department A)".

1.1 Key Exposure Problem

Ordinary digital signatures have a fundamental limitation: If the private key of a signer is compromised, all the signatures of that signer become worthless. This may become quite a realistic threat since if the private key is compromised, any message can be forged. All future signatures are invalidated as a result of such a compromise, and more importantly, no previously issued signatures can be trusted. Once a leakage has been identified, there may exist some key revocation mechanism to be involved immediately in order to prevent the generation of any signature using the compromised private key. However, this does not solve the problem of forgeability for past signatures. It is not possible to ask the signer to re-issue all previous signatures due to many physical and practical limitations. The problem of key exposure in ABS is more serious. In ABS, if a user's secret key is exposed to an adversary, the adversary can generate not only ABS for any documents, but can also sign any documents on behalf of any users with the same attributes. The exposure of one user's secret key not only requires changing the attribute name for the whole group, but also renders all previously obtained ABS invalid, because one cannot distinguish whether a signature is generated by an adversary after it has obtained one of the secret keys or by a legitimate user before key exposure.

Forward Secure Signature. Forward-secure signature schemes are designed to resolve the key exposure: a fundamental limitation of digital signature. The goal of a forward-secure signature scheme is to preserve the validity of past signatures even if the current secret key has been compromised. The concept was first suggested by Anderson [2], and solutions were designed by Bellare and Miner [4]. The idea is that even a compromise of the present secret key does not enable an adversary to forge signatures pertaining to the past. This can be achieved by the key evolution paradigm: dividing the total time of the validity of the public key into T time periods, and using a different secret key in each time period while the public key remains the same. Each subsequent secret key is computed from the current secret key via an update algorithm, while any past secret key cannot be computed by the current one. The time period during which a message is signed becomes part of the signature as well. The property of forward security means that even if the current secret key is compromised, a forger cannot forge signatures for past time periods. In other words, the forger can only forge signatures for documents pertaining to time periods after the exposure but not before. The integrity of documents signed before the exposure remains intact.

1.2 Contribution

We propose a new notion called *Forward Secure Attribute-Based Signatures (FS-ABS)*. It is similar to a normal ABS but providing forward security. That is, even when a secret key is compromised, previously generated signatures remain valid and do not need to be re-generated. It can greatly reduce the damage of exposure of any secret key of users in the environment. We formally define the security of FSABS, provide a generic design of FSABS and suggest some efficient instantiations.

2 Preliminaries

This section briefly reviews the preliminaries required in our scheme.

2.1 Monotone Span Programs

Let $\Upsilon : \{0,1\}^n \to \{0,1\}$ be a monotone boolean function. A monotone span program for Υ over a field $\mathbb{F}$ is an $\ell \times t$ matrix $\mathbf{M}$ with entries in $\mathbb{F}$, along with a labeling function $a : [\ell] \to [n]$ that associates each row of $\mathbf{M}$ with an input variable of Υ, that for every $(x_1, ..., x_n) \in \{0,1\}^n$, satisfies the following:

$$\Upsilon(x_1, ..., x_n) = 1 \Longleftrightarrow \exists \boldsymbol{v} \in \mathbb{F}^{1\times\ell} : \ \boldsymbol{v}\mathbf{M} = [1, 0, 0, ..., 0], \text{ and } (\forall i : x_{a(i)} = 0 \Rightarrow v_i = 0).$$

In other words, $\Upsilon(x_1, ..., x_n) = 1$ if and only if the rows of $\mathbf{M}$ indexed by $\{i | x_{a(i)} = 1\}$ span the vector $[1, 0, 0, ..., 0]$. We call ℓ the length and t the width of the span program, and $\ell + t$ the size of the span program.

2.2 NIWI Proof of Knowledge

We give a brief overview of the non-interactive witness-indistinguishable (NIWI) proof of knowledge. We refer the reader to [10,11] for detailed definitions.

Let R be an efficiently computable ternary relation. For triplets $(gk, x, w) \in R$ we call gk the setup, x the statement and w the witness. Given some gk we let L be the language consisting of statements in R. A non-interactive proof system for a relation R comprised of the following algorithms:

- Setup: Outputs a setup (gk, sk).
- CRSGen: On input (gk, sk), outputs a reference string crs.
- Prove: On input (gk, crs, x, w), where $(gk, x, w) \in R$, outputs a proof π.
- Verify: On input $(gk, \mathsf{crs}, x, \pi)$, outputs 1 if the proof is acceptable and 0 if rejecting the proof.

We call (Setup, CRSGen, Prove, Verify) a non-interactive proof system for R with Setup if it has the completeness and soundness properties described below.

– The *perfect completeness* requirement is that for all adversaries $\mathfrak{A}$ we have

$$\Pr[(gk, sk) \leftarrow \mathsf{Setup}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{CRSGen}(gk, sk); (x, w) \leftarrow \mathfrak{A}(gk, \mathsf{crs}); \\ \pi \leftarrow \mathsf{Prove}(gk, \mathsf{crs}, x, w) : \mathsf{Verify}(gk, \mathsf{crs}, x, \pi) = 1 \text{ if } (gk, x, w) \in R] = 1.$$

– The *perfect soundness* requirement is that for all adversaries $\mathfrak{A}$ we have

$$\Pr[\quad (gk, sk) \leftarrow \mathsf{Setup}(1^\lambda); \qquad \mathsf{crs} \leftarrow \mathsf{CRSGen}(gk, sk); \\ (x, \pi) \leftarrow \mathfrak{A}(gk, \mathsf{crs}) : \mathsf{Verify}(gk, \mathsf{crs}, x, \pi) = 0 \text{ if } x \notin L] = 1.$$

A non-interactive proof is *composable witness indistinguishable* if there is a probabilistic polynomial time simulator CRSSim, such that for all non-uniform polynomial time adversaries $\mathfrak{A}$ we have

$$\Pr[(gk, sk) \leftarrow \mathsf{Setup}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{CRSGen}(gk, sk) : \mathfrak{A}(gk, \mathsf{crs}) = 1] \\ \approx \Pr[(gk, sk) \leftarrow \mathsf{Setup}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{CRSSim}(gk, sk) : \mathfrak{A}(gk, \mathsf{crs}) = 1],$$

and for all adversaries $\mathfrak{A}$ we have:

$$\Pr[(gk, sk) \leftarrow \mathsf{Setup}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{CRSSim}(gk, sk); \\ (x, w_0, w_1) \leftarrow \mathfrak{A}(gk, \mathsf{crs}); \pi \leftarrow \mathsf{Prove}(gk, \mathsf{crs}, x, w_0) : \mathfrak{A}(\pi) = 1] \\ = \Pr[(gk, sk) \leftarrow \mathsf{Setup}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{CRSSim}(gk, sk); \\ (x, w_0, w_1) \leftarrow \mathfrak{A}(gk, \mathsf{crs}); \pi \leftarrow \mathsf{Prove}(gk, \mathsf{crs}, x, w_1) : \mathfrak{A}(\pi) = 1],$$

where we require $(gk, x, w_0), (gk, x, w_1) \in R$.

A non-interactive proof is a proof of knowledge (*perfect knowledge extraction*) if there is a probabilistic polynomial time knowledge extractor $(\mathsf{Ext}_1, \mathsf{Ext}_2)$, such that for all non-uniform polynomial time adversaries $\mathfrak{A}$ we have

$$\Pr[(gk, sk) \leftarrow \mathsf{Setup}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{CRSGen}(gk, sk) : \mathfrak{A}(gk, \mathsf{crs}) = 1] \\ \approx \Pr[(gk, sk) \leftarrow \mathsf{Setup}(1^\lambda); (\mathsf{crs}, \tau) \leftarrow \mathsf{Ext}_1(gk, sk) : \mathfrak{A}(gk, \mathsf{crs}) = 1],$$

and for all adversaries $\mathfrak{A}$ we have:

$$\Pr[(gk, sk) \leftarrow \mathsf{Setup}(1^\lambda); (\mathsf{crs}, \tau) \leftarrow \mathsf{Ext}_1(gk, sk); (x, \pi) \leftarrow \mathfrak{A}(gk, \mathsf{crs}); \\ w \leftarrow \mathsf{Ext}_2(gk, \mathsf{crs}, \tau, x, \pi) : \mathsf{Verify}(gk, \mathsf{crs}, x, \pi) = 0 \text{ or } (gk, x, w) \in R] = 1.$$

Definition 1. *A non-interactive proof system is a perfect non-interactive witness indistinguishable (NIWI) proof of knowledge if it has perfect completeness, perfect soundness, composable witness indistingushable and perfect knowledge extraction.*

3 Security Models

We give our security models of forward secure attributed-based signatures and define relevant security notions.

3.1 Syntax of Forward Secure Attribute-Based Signatures

Let $\mathbb{A}$ be the universe of possible attributes. A *claim-predicate* over $\mathbb{A}$ is a monotone boolean function, whose inputs are associated with attributes of $\mathbb{A}$. We say that an attribute set $\mathcal{A} \subseteq \mathbb{A}$ satisfies a claim-predicate Υ if $\Upsilon(\mathcal{A}) = 1$ (where an input is set to be true if its corresponding attribute is present in $\mathcal{A}$).

Definition 2. *A forward secure attribute-based signature scheme is a tuple of six algorithms parameterized by a universe of possible attributes $\mathbb{A}$, a total number of time period T and a message space $\mathbb{M}$:*

- FSABS.TSetup (to be run by a trustee): On input the security parameter 1^λ, generates public reference information TPK.
- FSABS.ASetup (to be run by an attribute-issuing authority): On input the security parameter 1^λ, generates a key pair (APK, ASK).
- FSABS.AttrGen: On input $(ASK, \mathcal{A} \subseteq \mathbb{A})$, outputs an assoicated signing key $sk_{\mathcal{A},0}$.
- FSABS.Update: On input $sk_{\mathcal{A},i}$ and a time period j (where $i < j \leq T$), outputs an assoicated signing key $sk_{\mathcal{A},j}$.
- FSABS.Sign: On input $(PK = (TPK, APK), sk_{\mathcal{A},t}, m \in \mathbb{M}, \Upsilon, t)$, where $\Upsilon(\mathcal{A}) = 1$ and t is the time period, outputs a signature π.
- FSABS.Verify: On input $(PK = (TPK, APK), m, \Upsilon, \pi, t)$, outputs accept or reject.

Correctness. FSABS schemes must satisfy that signatures signed according to specification are accepted during verification.

3.2 Notions of Security of Forward Secure Attribute-Based Signatures

Security of forward secure attributed-based signature schemes has unforgeability and privacy.

1. Unforgeability.
   The unforgeability for forward secure attributed-based signature schemes is defined in the following game between the Challenger $\mathfrak{C}$ and the Adversary $\mathfrak{A}$ in which $\mathfrak{A}$ is given access to oracles $\mathcal{JO}$, $\mathcal{CO}$ and $\mathcal{SO}$:
   (a) $\mathfrak{C}$ generates

   $$TPK \leftarrow \mathsf{FSABS.TSetup}(1^\lambda) \quad \text{and} \quad (APK, ASK) \leftarrow \mathsf{FSABS.TSetup}(1^\lambda).$$

   $\mathfrak{C}$ gives $\mathfrak{A}$ the public information $PK = (TPK, APK)$.
   (b) $\mathfrak{A}$ may query the following oracles according to any adaptive strategy.
   - $sk_{\mathcal{A},t} \leftarrow \mathcal{GO}(\mathcal{A}, t)$. The AttrGen Oracle, on input an attribute set $\mathcal{A}$ and a time period t, returns the corresponding secret key $sk_{\mathcal{A},t} \leftarrow \mathsf{FSABS.Update}(\mathsf{LABS.AttrGen}(ASK, \mathcal{A}), t)$. We require for the same $(i, \mathcal{A}, t)$ as input, the same $sk_{\mathcal{A},t}$ is the output.

– $\sigma \leftarrow \mathcal{SO}(\mathcal{A}, t, m, \Upsilon)$. The Sign Oracle, on input an attribute set $\mathcal{A}$, a time period t, a message m and a claim-predicate Υ where $\Upsilon(\mathcal{A}) = 1$, returns a valid signature $\sigma \leftarrow \mathsf{FSABS.Sign}(PK, sk_{\mathcal{A},t} \leftarrow \mathsf{FSABS.Update}(\mathsf{FSABS.AttrGen}(ASK, \mathcal{A}), t), m, \Upsilon, t)$.

(c) $\mathfrak{A}$ gives $\mathfrak{C}$ a time period t^*, a claim-predicate Υ^*, a message m^* and a signature π^*.

$\mathcal{B}$ wins the game if:
(1) $\mathsf{FSABS.Verify}(PK, m^*, \Upsilon^*, \pi^*, t^*)$=accept ;
(2) $(\cdot, t^*, m^*, \Upsilon^*)$ is not a query input to $\mathcal{SO}$; and
(3) $\Upsilon^*(\mathcal{A}) = 0$ for all $(\mathcal{A}, t)$ queried to $\mathcal{GO}$ with $t \leq t^*$.
We denote by

$$\mathbf{Adv}_{\mathfrak{A}}^{unf} = \Pr[\mathfrak{A} \text{ wins the game }].$$

Definition 3 (Unforgeability). *A Forward Secure Attribute-Based Signature scheme is unforgeable if for all PPT adversary* $\mathfrak{A}$, $\mathbf{Adv}_{\mathfrak{A}}^{unf}$ *is negligible.*

The unforgeability ensures that a valid signature must be signed by a user with attributes satisfying the predicate in the current time period.

2. Privacy.
In order to protect privacy, forward secure ABS must hide the attributes used during signature generation. This is defined in the following game between the Challenger $\mathfrak{C}$ and the Adversary $\mathfrak{A}$ in which $\mathfrak{A}$ is given the ASK. $\mathfrak{A}$ does not need to query any oracle since it can generate the signing keys by himself.

(a) $\mathfrak{C}$ generates

$$TPK \leftarrow \mathsf{FSABS.TSetup}(1^\lambda) \text{ and } (APK, ASK) \leftarrow \mathsf{FSABS.TSetup}(1^\lambda).$$

$\mathfrak{C}$ gives $\mathfrak{A}$ the public information $PK = (TPK, APK)$ and also ASK.

(b) $\mathfrak{A}$ sends $\mathfrak{C}$ $(\mathcal{A}_0, \mathcal{A}_1, m, t, \Upsilon)$, where $\Upsilon(\mathcal{A}_0) = \Upsilon(\mathcal{A}_1) = 1$.

(c) $\mathfrak{C}$ chooses a random bit $b \in \{0, 1\}$ and generates

$$sk_{\mathcal{A}_b,t} \leftarrow \mathsf{FSABS.Update}(\mathsf{FSABS.AttrGen}(ASK, \mathcal{A}_b), t).$$

It generates the signature $\pi_b \leftarrow \mathsf{FSABS.Sign}(PK, sk_{\mathcal{A}_b,t}, m, \Upsilon, t)$ and sends σ_b to $\mathfrak{A}$.

(d) $\mathfrak{A}$ outputs a bit b'.

$\mathfrak{A}$ wins the game if $b' = b$. We denote by

$$\mathbf{Adv}_{\mathfrak{A}}^{Anon} = \left| \Pr[\ \mathfrak{A} \text{ wins the game }] - \frac{1}{2} \right|.$$

Definition 4 (Privacy). *A Forward Secure Attribute-Based Signature scheme is private if for all PPT adversary* $\mathfrak{A}$, $\mathbf{Adv}_{\mathfrak{A}}^{Anon}$ *is negligible.*

The privacy property ensures that it is hard to distinguish between two signatures, each associated with different attributes, which both satisfy the claim-predicate.

4 Our Generic Forward Secure Attribute-Based Signature Scheme

Our scheme is motivated by the attribute-based signature scheme from [15].

4.1 Forward Secure Credential Bundle

We extend the *credential bundle* primitive in [15] with forward security.

Definition 5 (Forward Secure Credential Bundle). *A forward secure credential bundle scheme is parameterized by a message space $\mathcal{M}$ and a time period T, and consists of the following four algorithms.*

- $\mathsf{CB.Setup}$*: On input a security parameter 1^λ, outputs a verification key vk and a secret key sk.*
- $\mathsf{CB.Gen}$*: On input sk and a set of messages $\{m_1, \ldots, m_n\} \subseteq \mathcal{M}$, outputs a credential c_0 (of time 0), which consists of a tag τ_0 and values $\sigma_{1,0}, \ldots, \sigma_{n,0}$.*
- $\mathsf{CB.Update}$*: On input a credential c_{t_1} of time t_1 and a new time period t_2 (with $t_1 < t_2 \leq T$), outputs a new credential c_{t_2}.*
- $\mathsf{CB.Ver}$*: On input vk, a message m, a time period t and a credential (τ, σ), outputs 1 for* accept *and 0 for* reject*.*

The scheme is correct *if for all $(vk, sk) \leftarrow \mathsf{CB.Setup}(1^\lambda)$, $c_0 = (\tau_0, \sigma_{1,0}, \ldots, \sigma_{n,0})$ $\leftarrow \mathsf{CB.Gen}(sk, (m_1, \ldots, m_n))$ and $c_t = (\tau_t, \sigma_{1,t}, \ldots, \sigma_{n,t}) \leftarrow \mathsf{CB.Update}(c_0, t)$, we have $\mathsf{CB.Ver}(vk, m_i, t, (\tau_t, \sigma_{i,t})) = 1$ for all $i \in [1, n]$.*

Observe that one can generate a new bundle on a subset of attributes. Our security definition below requires that taking a subset of a single bundle and update is the only way to obtain a new bundle in time t_2 from existing bundles at time $t_1 \leq t_2$. In particular, attributes from several bundles cannot be combined; and credentials of the present time cannot be used to find credentials of the past.

Definition 6. *A credential bundle scheme is forward secure if the success probability of any polynomial-time adversary in the following experiment is negligible:*

1. *Run $(vk, sk) \leftarrow \mathsf{CB.Setup}(1^\lambda)$, and give vk to the adversary.*
2. *The adversary is given access to an extract oracle with input $(t, (m_1, \ldots, m_n))$. It obtains $c_t \leftarrow \mathsf{CB.Update}(c_0, t)$, where $c_0 \leftarrow \mathsf{CB.Gen}(sk, (m_1, \ldots, m_n))$.*
3. *Finally the adversary outputs $(t^*, \tau^*, (m_1^*, \sigma_1^*), \ldots, (m_{n^*}^*, \sigma_{n^*}^*))$.*

We say the adversary succeeds if $\mathsf{CB.Ver}(vk, m_i^*, t^*, (\tau^*, \sigma_i^*)) = 1$ for all $i \in [1, n^*]$, and if no superset of $(m_1^*, \ldots, m_{n^*}^*)$, was ever queried (in a single query) to the extract oracle with time $t \leq t^*$.

Instantiation. From any plain forward secure digital signature scheme (e.g. [12,1,16,8,7,14]) we can easily construct a credential bundle scheme in which the bundle is a collection of signatures of messages "$\tau || m_i$", where each m_i is the name of an attribute and τ is an identifier that is unique to each user. Conversely, when a credential bundle scheme is restricted to singleton sets of messages, its forward security definition is equivalent to normal forward secure digital signature.

4.2 Forward Secure ABS Construction

Let $\mathbb{A}$ be the desired universe of ABS attributes. Let $\mathbb{A}'$ denote a space of pseudo-attributes, where $\mathbb{A} \cap \mathbb{A}' = \emptyset$. For every message m and claim-predicate Υ we associate a psuedo-attribute $a_{m,\Upsilon} \in \mathbb{A}'$. Let CB be a secure credential bundle scheme, with message space $\mathbb{A} \cap \mathbb{A}'$, and let ($\mathsf{NIWI.Setup}$, $\mathsf{NIWI.CRSGen}$, $\mathsf{NIWI.Prove}$, $\mathsf{NIWI.Verify}$) be a perfect NIWI proof of knowledge scheme. Our ABS construction is as follows:

$\underline{\mathsf{FSABS.TSetup}}$: Let λ be a security parameter. The signature trustee runs $(gk, sk) \leftarrow \mathsf{NIWI.Setup}(1^\lambda)$, $\mathsf{crs} \leftarrow \mathsf{NIWI.CRSGen}(gk, sk)$ as well as $(tvk, tsk) \leftarrow \mathsf{CB.Setup}(1^\lambda)$ and publishes $TPK = (gk, \mathsf{crs}, tvk)$.

$\underline{\mathsf{FSABS.ASetup}}$: The attribute-issuing authority runs $(avk, ask) \leftarrow \mathsf{CB.Setup}(1^\lambda)$ and publishes $APK = avk$ and sets $ASK = ask$.

$\underline{\mathsf{FSABS.AttrGen}}$: The key generation algorithm takes as input a subset of attributes $\mathcal{A} \subset \mathbb{A}$ and the secret key ASK. Ensure that $\mathcal{A}$ contains no pseudo-attributes. Then output the result of $sk_{\mathcal{A},0} \leftarrow \mathsf{CB.Gen}(ASK, \mathcal{A})$.

$\underline{\mathsf{FSABS.Update}}$: On input a signing key $sk_{\mathcal{A},i}$ for attribute $\mathcal{A}$ and new time period j, if $i < j \leq T$ the user updates the secret key by $sk_{\mathcal{A},j} \leftarrow \mathsf{CB.Update}(sk_{\mathcal{A},i}, j)$.

$\underline{\mathsf{FSABS.Sign}}$: The signing algorithm takes as input the public keys TPK, APK, a signing key $sk_{\mathcal{A},i}$ for attribute $\mathcal{A}$ and current time period j, a message m and a claim-predicate Υ. Assume $\Upsilon(\mathcal{A}) = 1$. Parse $sk_{\mathcal{A},j}$ as $(\tau, \{\sigma_{a,j} | a \in \mathcal{A}\})$. Define $\tilde{\Upsilon} := \Upsilon \vee a_{m,\Upsilon}$, where $a_{m,\Upsilon} \in \mathbb{A}'$ is the pseudo-attribute associated with (m, Υ). Thus, we still have $\tilde{\Upsilon}(\mathcal{A}) = 1$. Let $\{a_1, \ldots a_n\}$ denote the attributes appearing in $\tilde{\Upsilon}$. Let vk_i be avk if attribute a_i is a pseudo-attribute, and tvk otherwise. Finally, let $\Phi[vk, m, \Upsilon, j]$ denote the following boolean expression:

$$\exists \tau, \sigma_1, \ldots, \sigma_n : \tilde{\Upsilon}\left(\{a_i | \mathsf{CB.Ver}\left(vk_i, a_i, j, (\tau, \sigma_i)\right) = 1\}\right) = 1$$

For each i, set $\hat{\sigma_{i,j}} = \sigma_{a_i,j}$ from $sk_{\mathcal{A},i}$ if it is present, and to any arbitrary value otherwise. Compute $\pi \leftarrow \mathsf{NIWI.Prove}(\mathsf{crs}, \Phi[vk, m, \Upsilon, j], (\tau, \hat{\sigma_{1,j}}, \ldots, \hat{\sigma_{n,j}}))$. Output π as the ABS signature.

$\underline{\mathsf{FSABS.Verify}}$: The verification algorithm takes as input a message M, a signature π, a time period j, a signing policy Υ and the public keys TPK, APK. Output the result of $\mathsf{NIWI.Verify}(\mathsf{crs}, \Phi[vk, m, \Upsilon, j], \pi)$.

Security of the Generic Construction

Theorem 1. *The scheme is private if the NIWI is composable witness indistinguishable.*

The privacy follows directly from the composable witness indistinguishable property of the NIWI.

Theorem 2. *The scheme is unforgeable if the NIWI has knowledge extraction and the* CB *is forward secure.*

Proof. If the NIWI scheme is sound, we can show that any adversary A that violates ABS unforgeability can be used to construct an algorithm B that breaks the security of the underlying credential bundle scheme, with non-negligible probability. Suppose B receives vk from the challenger C of the CB security experiment. Let B flips a random coin $b = 0/1$ and perform one of the following two simulations:

Simulation 0: B runs $(gk, sk) \leftarrow \mathsf{NIWI.Setup}(1^\lambda)$, $(\mathsf{crs}, \tau) \leftarrow \mathsf{NIWI.Ext}_1(gk, sk)$ and sets $tvk = vk$. Note that A cannot distinguish a real CRS from a simulated CRS by the security of the NIWI proof system. B gives $TPK = (gk, \mathsf{crs}, tvk)$ to A. B runs $(APK, ASK) \leftarrow \mathsf{FSABS.ASetup}(1^\lambda)$ honestly and gives APK to A.

When A makes a query $\mathcal{A} \subseteq \mathbb{A}$ to the FSABS.AttrGen Oracle, B computes the response honestly using ASK. When A makes a query $(\mathcal{A}, t, m, \Upsilon)$ to the FSABS.Sign Oracle, B requests from C the CB.Gen Oracle a singleton bundle for the pseudo-attribute associated with (m, Υ) and time t. B uses the result as a witness to generate a NIWI proof of $\Phi[vk, m, \Upsilon, t]$ to use as the simulated ABS signature.

Finally A outputs a valid forgery $(m^*, \Upsilon^*, \pi^*, t^*)$, B uses $\mathsf{NIWI.Ext}_2$ with the trapdoor τ to extract a witness for $\Phi[vk, m^*, \Upsilon^*, t^*]$. Extraction succeeds with overwhelming probability, thus we obtain a bundle that contains the pseudo-attribute associated with (m^*, Υ^*) and time t^*, or sufficient attributes to satisfy Υ^*. If the bundle contains the pseudo-attribute, then it represents a forgery against $tvk = vk$ from the experiment with C, since B has never requested $(t^*, (m^*, \Upsilon^*))$ from the CB.Gen Oracle.

Simulation 1: Similar to above, except that B sets $avk = vk$ instead of tvk. B honestly generates tvk as in FSABS.TSetup. B gives simulated ABS signatures to A by generating bundle signatures on the pseudo-attribute. B forwards all of A's queries on its FSABS.AttrGen Oracle to the CB.Gen Oracle provided by C. Finally A outputs an ABS forgery and B uses $\mathsf{NIWI.Ext}_2$ with the trapdoor τ to extract a witness. If the extracted bundle satisfies Υ^* (rather than contains the associated pseudo-attribute), then B returns the bundle as a forgery in the experiment with C.

The above simulations are identical from the view of A. Any valid forgery by A must be extracted to give a forgery suitable for one of the two simulations. Therefore, we can see that the advantage of one of the two simulations in its unforgeability game is comparable to that of A in the ABS forgery game (losing only a factor of $1/2$). □

Instantiation. We can instantiate our generic construction using the CB scheme is section 4.1 and the NIWI proof of Groth and Sahai [11]. Note that the Groth and Sahai's proof (for the SXDH and DLIN instantiation) only has a knowledge

extractor for the group elements, but not the exponent elements [17]. Observe that the private keys of our CB scheme only consist of the group elements. Therefore, we simply use the SXDH or DLIN instantiation of the Groth and Sahai's proof together with our CB scheme.

5 Conclusion

Key exposure is a fundamental limitation of ordinary digital signatures: If the secret key of a signer is compromised, all the signatures of that signer become worthless. This issue can be properly addressed using forward-secure techniques, which ensures that past signatures remain valid even if the current secret key is leaked. While the notion of attributed-based signatures was introduced in 2008 and many variants have been proposed, the issue of key exposure in ABS has never been formally studied. We filled this gap by giving a generic (also the first) design of forward-secure attributed-based signatures with provable security. We believe the result presented in this paper will draw the attention of cryptographers and anticipate more efficient designs of forward-secure attributed-based signatures.

Acknowledgement. Joseph K. Liu, Xinyi Huang and Jianying Zhou are supported by the EMA project SecSG-EPD090005RFP(D). Willy Susilo is supported by the ARC Future Fellowship (FT0991397).

References

1. Abdalla, M., Reyzin, L.: A New Forward-Secure Digital Signature Scheme. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 116–129. Springer, Heidelberg (2000)
2. Anderson, R.: Two remarks on public-key cryptology. Manuscript, September 2000. Relevant material presented by the author in an invited lecture at the Fourth ACM Conference on Computer and Communications Security (1997)
3. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003)
4. Bellare, M., Miner, S.: A Forward-Secure Digital Signature Scheme. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 431–448. Springer, Heidelberg (1999)
5. Bender, A., Katz, J., Morselli, R.: Ring Signatures: Stronger Definitions, and Constructions Without Random Oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 60–79. Springer, Heidelberg (2006)
6. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
7. Boyen, X., Shacham, H., Shen, E., Waters, B.: Forward-secure signatures with untrusted update. In: ACM Conference on Computer and Communications Security, pp. 191–200. ACM (2006)
8. Camenisch, J., Koprowski, M.: Fine-grained forward-secure signature schemes without random oracles. Discrete Applied Mathematics 154(2), 175–188 (2006)

9. Chaum, D., van Heyst, E.: Group Signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
10. Groth, J.: Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Heidelberg (2006)
11. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
12. Krawczyk, H.: Simple forward-secure signatures from any signature scheme. In: ACM Conference on Computer and Communications Security, pp. 108–115. ACM (2000)
13. Li, J., Au, M.H., Susilo, W., Xie, D., Ren, K.: Attribute-based signature and its applications. In: Feng, D., Basin, D.A., Liu, P. (eds.) ASIACCS, pp. 60–69. ACM (2010)
14. Libert, B., Quisquater, J.-J., Yung, M.: Forward-secure signatures in untrusted update environments: efficient and generic constructions. In: ACM Conference on Computer and Communications Security, pp. 266–275. ACM (2007)
15. Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-Based Signatures. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 376–392. Springer, Heidelberg (2011)
16. Malkin, T., Micciancio, D., Miner, S.K.: Efficient Generic Forward-Secure Signatures with an Unbounded Number of Time Periods. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 400–417. Springer, Heidelberg (2002)
17. Meiklejohn, S.: An extension of the groth-sahai proof system. Master's thesis, Brown University (2009)
18. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001)
19. Shacham, H., Waters, B.: Efficient Ring Signatures Without Random Oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 166–180. Springer, Heidelberg (2007)