Attribute-Based Signatures[⋆]

Hemanta K. Maji[1], Manoj Prabhakaran[1], and Mike Rosulek[2]

[1] Department of Computer Science, University of Illinois, Urbana-Champaign
{hmaji2,mmp}@uiuc.edu
[2] Department of Computer Science, University of Montana
mikero@cs.umt.edu

Abstract. We introduce *Attribute-Based Signatures (ABS)*, a versatile primitive that allows a party to sign a message with fine-grained control over identifying information. In ABS, a signer, who possesses a set of attributes from the authority, can sign a message with a predicate that is satisfied by his 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, the signature hides the attributes used to satisfy the predicate and any identifying information about the signer (that could link multiple signatures as being from the same signer). Furthermore, users cannot collude to pool their attributes together.

We give a general framework for constructing ABS schemes, and then show several practical instantiations based on groups with bilinear pairing operations, under standard assumptions. Further, we give a construction which is secure even against a malicious attribute authority, but the security for this scheme is proven in the generic group model. We describe several practical problems that motivated this work, and how ABS can be used to solve them. Also, we show how our techniques allow us to extend Groth-Sahai NIZK proofs to be simulation-extractable and identity-based with low overhead.

1 Introduction

Alice, a finance manager in a big corporation, while going through her company's financial records, has learned about a major international scandal. She decides to send these records to a major newspaper, retaining her anonymity, but with a proof that she indeed has access to the records in question. It turns out that several people, due to a combination of reasons, may have access to these records: those in the New York, London or Tokyo office who are either finance managers associated with project Skam, or internal auditors. Alice considers using a *ring signature* [26] to endorse her message anonymously, but realizes that it is infeasible not only because of the large number of people involved, but also because she does not know who these people are. She realizes she cannot use a *group signature* [14] either, because the set of people Alice needs to refer to here is idiosyncratic to her purposes, and may not have been already collected into a group.[1] She is also aware of *mesh signatures* [9], but mesh signatures provide

[⋆] Partially supported by NSF grants CNS 07-16626 and CNS 07-47027.

[1] Even if a group exists, the group manager could identify Alice as the informant.

A. Kiayias (Ed.): CT-RSA 2011, LNCS 6558, pp. 376–392, 2011.

no way to convince the newspaper that the financial record was endorsed by a single person, not, say, a programmer in the New York office colluding with an internal auditor in the Smalltown office.

Alice's needs in this story reflect the challenges in a system where the roles of the users depend on the *combination* of attributes they possess. In such systems, users obtain multiple attributes from one or more *attribute authorities*, and a user's capabilities in the system (e.g., sending or receiving messages, access to a resource) depend on their attributes. While offering several advantages, attribute-based systems also present fundamental cryptographic challenges. For instance, suppose Alice wants to simply send a message to the above group of people using an "attribute-based messaging" system; then to provide *end-to-end* secure communication, it must be possible for her to encrypt a message using attribute-keys (rather than individual users' keys). Recently cryptographic tools have emerged to tackle some of these challenges for encryption [27,16,3,31]. In this work, we provide a solution for authentication, which among other things, will let Alice in the above example leak the financial records anonymously, but with the appropriate claim regarding her credentials.

Why Attribute-Based Signatures?

The kind of authentication required in an attribute-based system differs from that offered by digital signatures, in much the same way public-key encryption does not fit the bill for attribute-based encryption. An attribute-based solution requires a richer semantics, including anonymity requirements, similar to signature variants like group signatures [14], ring signatures [26], and mesh signatures [9]. The common theme in all these signature primitives is that they provide a guarantees of *unforgeability* and *signer anonymity*. A valid signature can only be generated in particular ways, but the signature does not reveal any further information about which of those ways was actually used to generate it.

More specifically, group and ring signatures reveal only the fact that a message was endorsed by one of a list of possible signers. In a ring signature, the list is public, chosen by the signer *ad hoc*, and given explicitly. In a group signature, the group must be prepared in advance by a group manager, who can revoke the anonymity of any signer. In mesh signatures, a valid signature describes an access structure and a list of pairs (m_i, vk_i), where each vk_i is the verification key of a standard signature scheme. A valid mesh signature can only be generated by someone in posession of enough standard signatures σ_i, each valid under vk_i, to satisfy the given access structure.

In this work we introduce *attribute-based signatures (ABS)*. Signatures in an ABS scheme describe a message and a predicate over the universe of attributes. A valid ABS signature attests to the fact that "a single user, whose attributes satisfy the predicate, endorsed the message." We emphasize the word "single" in this informal security guarantee; ABS signatures, as in most attribute-based systems, require that colluding parties not be able to pool their attributes together.[2] Furthermore, attribute signatures do

[2] Note that for attribute-based *encryption*, if collusion is allowed there are fairly easy solutions; but for ABS, even after allowing collusion (for instance by considering all users to have the same identity while generating keys), the residual primitive is essentially a mesh signature, which is already a non-trivial cryptographic problem.

not reveal more than the claim being made regarding the attributes, even in the presence of other signatures.

Ring and group signatures are then comparable to special cases of ABS, in which the only allowed predicates are *disjunctions* over the universe of attributes (identities). Only one attribute is required to satisfy a disjunctive predicate, so in these cases collusion is not a concern. As in ring signatures, ABS signatures use *ad hoc* predicates. Mesh signatures allow more fine-grained predicates, but do not provide hiding of signature data that would be needed in an ABS scheme. A straight-forward application of mesh signatures as an ABS scheme would either allow collusion (as in the previous example, a New York programmer colluding with a Smalltown auditor to satisfy the "New York auditor" predicate) or allow signatures to be associated with a pseudonym of the signer (thus linking several signatures as originating from the same signer).

Applications

Attribute-based signatures have natural applications in many systems where users' capabilities depend on possibly complex combinations of attributes. ABS is a natural choice for simple authentication in such systems. One of our motivations for developing such schemes comes from the authentication requirements in an Attribute-Based Messaging (ABM) system. In addition to the "leaking secrets" application described above, in Section 6 we also identify applications in trust negotiation systems.

Overview of Our Results

We introduce the concept of Attribute-Based Signatures (ABS) as a powerful primitive with several applications and several efficient instantiations. Our main technical contributions in this work are the following:

- A formal security definition for ABS, that includes the guarantees of unforgeability (even in the presence of collusion) and privacy for the signer.
- A general framework for constructing ABS schemes. Our framework consists of a "credential bundle" representing the attributes associated with a single user and a non-interactive proof of credential ownership that can be bound to a message. The credential bundle must have the property that multiple users should not be able to collude and combine their credentials. The proof system must have some zero-knowledge-like guarantee so that the signature does not leak information about the signer's identity or attributes.

  We instantiate this framework using Boneh-Boyen [6] or Waters [30] signatures as the credential bundle, and Groth-Sahai NIZK proofs [18] as the efficient non-interactive proof system. These instantiations provide practical ABS schemes secure under standard assumptions in bilinear groups.
- We present a practical ABS scheme suitable for high throughput systems. This construction deviates from our framework of credential bundles and proof of credential ownership. In this scheme we do employ a credential bundle scheme (same as the one in the last item above), but use a novel randomization technique to blind the actual attributes. This gives the best efficiency among our schemes. Further, this scheme

remains secure even against a corrupt attribute-authority. However, the security of this scheme is proven in the heuristic generic-group model (augmented to handle groups with bilinear pairings).

– One of the most striking features of our construction is that it is very easily amenable to natural multi-authority settings. We describe practical considerations related to such a deployment.

– In the full version we show how our techniques of incorporating digital signatures and non-interactive proofs can be used to add *simulation-extractability* to the Groth-Sahai proof system, several orders of magnitude more efficiently than the only other comparable scheme, constucted by Groth in [17].

Which among the above schemes will suit an application will depend on the specific efficiency and security requirements in the system. In all these schemes, the privacy is unconditional, and it is only the unforgeability that depends on computational assumptions. Within a large enterprise setting (with pre-authenticated users) where the threat of forgery may be limited but the volume of signatures may be large, the final scheme may be the most suited. In more susceptible systems with a high security requirement, one of the schemes based on the Groth-Sahai proof systems maybe more suitable (at the expense of efficiency). The choice also depends on whether the application demands high-volume real-time performance (as in a messaging system) or involves only offline signing and verification (as in leaking a secret).

All of our instantiations depend on expressing the attribute predicate as a monotone-span program, which is the state of the art for attribute-based cryptography [16,3,31]. We remark that unlike in many constructions of attribute-based encryption schemes, we achieve "full security" in all our constructions. That is, we do not weaken the definition in the manner of "selective-ID" security. Nor do we need to limit our construction to a small universe of attributes. In all our instantiations, attributes can be arbitrary strings: given a collision-resistant hash function, an *a priori* unbounded attribute universe can be used.

Further Related Work

Groups with bilinear pairings have been used to construct identity-based (e.g., [8]) and attribute-based encryption schemes [27,16,3]. Non-interactive zero-knowledge proofs (including identity-based proofs) have previously been used in the context of efficient constructions of signature primitives [1,20,10,17].

Khader [22,21] proposes a notion called *attribute-based group signatures*. This primitive hides only the identity of the signer, but reveals which attributes the signer used to satisfy the predicate. It also allows a group manager to identify the signer of any signature (which is similar to the semantics of group signatures [14]); in contrast we require signer privacy to hold against everyone, including all authorities.

Subsequent to a preliminary (unpublished) version of this work, Li and Kim [24] gave an ABS scheme that supports predicates which are solely conjunctions of attributes (hence privacy is required only for the identity of the signer and not for the attributes used in satisfying the predicate), and is restricted to a "selective" unforgeability definition. Guo and Zeng [19] construct an attribute-based signature scheme, although their definition of security did not include any privacy for the signer. Shahandashti and Safavi-Naini [28] and Li et al. [23] construct efficient ABS schemes that support predicates consisting of a single threshold gate.

Binding a non-interactive proof to a message, as we do, is also a feature of *identity-based* proofs [20], in which every proof is bound to some identity, and proofs under one identity cannot be used to forge any proofs under a different identity. Indeed, such ID-based proofs have been used to construct signature-like primitives; however the construction from [20] does not have all the properties we need.

Anonymous credentials [13] is one primitive that has some parallels with ABS, but with goals that differ from ABS in several important ways. ABS could be considered as providing some of the functionality of AC as a very special case, but with a weaker anonymity guarantee. Conversely, some of the techniques used to construct efficient AC systems bear some resemblance to some of our efficient ABS constructions. In the full version we discuss these similarities and differences in more detail.

Another related primitive (but much simpler than ABS) is identity-based signatures (IBS) [29]. It is well-known that a simple scheme using traditional certificates realizes IBS, but dedicated schemes aimed at achieving better efficiency have been widely studied. We refer the reader to a comprehensive survey by Bellare et al. [2] for details.

Supporting multiple attribute-authorities is crucial to many attribute-based systems. Previously, there has been much interest on this aspect for attribute-based *encryption* schemes; see Chase et al. [11,12]. The constructions in this paper readily generalize to the multi-authority setting.

2 Preliminaries

2.1 Groups with Bilinear Pairings

Let $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$ be cyclic (multiplicative) groups of order p, where p is a prime. Let g be a generator of $\mathbb{G}$, and h be a generator of $\mathbb{H}$. Then $e : \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{G}_T$ is a *bilinear pairing* if $e(g, h)$ is a generator of $\mathbb{G}_T$, and $e(g^a, h^b) = e(g, h)^{ab}$ for all a, b. We review several standard cryptographic assumptions in such groups:

Definition 1 (q-SDH assumption [6]). *Let* $\mathbb{G}$, $\mathbb{H}$, *and* $\mathbb{G}_T$ *be as above. The* q-Strong Diffie-Hellman (q-SDH) assumption *holds in* $(\mathbb{G}, \mathbb{H})$ *if, given the elements* $(g, g^x, g^{x^2}, \ldots, g^{x^q}, h, h^x) \in \mathbb{G}^{q+1} \times \mathbb{H}^2$, *for random choice of* $x \leftarrow \mathbb{Z}_p$ *and random generators* $g \in \mathbb{G}, h \in \mathbb{H}$, *it is computationally infeasible to compute any pair of the form* $\left(c, g^{\frac{1}{x+c}}\right) \in \mathbb{Z}_p \times \mathbb{G}$.

Definition 2 (SXDH assumption [18]). *Let* $\mathbb{G}$, $\mathbb{H}$, *and* $\mathbb{G}_T$ *be as above. The* Symmetric External Diffie-Hellman (SXDH) assumption *holds in* $(\mathbb{G}, \mathbb{H})$ *if the standard Decisional Diffie-Hellman (DDH) assumption holds simultaneously in* $\mathbb{G}$ *and* $\mathbb{H}$.

Definition 3 (DLIN assumption [7]). *Let* $\mathbb{G}$, $\mathbb{H}$, *and* $\mathbb{G}_T$ *be as above, but with* $\mathbb{G} = \mathbb{H}$. *The* Decision-Linear (DLIN) assumption *holds in* $\mathbb{G}$ *if, given the elements* $(g^x, g^y, g^{rx}, g^{sy}, g^t) \in \mathbb{G}^5$, *for a random choice of* $x, y, r, s \leftarrow \mathbb{Z}_p$, *it is computationally infeasible to determine whether* $t = r + s$ *or* t *is random in* $\mathbb{Z}_p$.

2.2 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 $(x_1, \dots, x_n) \in \{0,1\}^n$, satisfies the following:

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

In other words, $\Upsilon(x_1, \dots, x_n) = 1$ if and only if the rows of $\mathbf{M}$ indexed by $\{i \mid x_{a(i)} = 1\}$ span the vector $[1, 0, 0, \dots, 0]$.

We call ℓ the *length* and t the *width* of the span program, and $\ell + t$ the *size* of the span program. Every monotone boolean function can be represented by some monotone span program, and a large class do have compact monotone span programs. In particular, given a circuit expressed using *threshold gates*, with the i-th gate being an $\binom{\ell_i}{t_i}$ threshold gate, it is easy to recursively construct a monotone span program with length $\sum_i(\ell_i - 1) + 1$ and width $\sum_i(t_i - 1) + 1$.

2.3 Non-interactive Proofs

We refer the reader to [18] for detailed definitions of non-interactive witness-indistinguishable (NIWI) proofs, but give a brief overview of the necessary definitions here. A NIWI scheme is comprised of the following main algorithms:

- NIWI.Setup: Outputs a reference string crs.
- NIWI.Prove: On input $(crs; \Phi; x)$, where Φ is a boolean formula and $\Phi(x) = 1$, outputs a proof π.
- NIWI.Verify: On input $(crs; \Phi; \pi)$, outputs a boolean.

The completeness requirement is that $\mathsf{NIWI.Verify}(crs; \Phi; \mathsf{NIWI.Prove}(crs; \Phi; x)) = 1$, if $\Phi(x) = 1$ (i.e., x is a *witness* for Φ). The (perfect) witness indistinguishability requirement is that the distributions $\mathsf{NIWI.Prove}(crs; \Phi; x_1)$ and $\mathsf{NIWI.Prove}(crs; \Phi; x_2)$ are identical when x_1 and x_2 are witnesses for Φ. For the soundness/proof of knowledge requirement, we require the following additional algorithms:

- NIWI.SimSetup: Outputs a simulated reference string crs and trapdoor ψ.
- NIWI.Extract: On input $(crs, \psi; \Phi; \pi)$, outputs a witness x.

We require that the crs output by NIWI.SimSetup is indistinguishable to that of NIWI.Setup. Further, we require that for every $(crs, \psi) \leftarrow$ NIWI.SimSetup, if $\mathsf{NIWI.Verify}(crs; \Phi; \pi) = 1$ then $\mathsf{NIWI.Extract}(crs, \psi; \Phi; \pi)$ outputs a valid witness for Φ, with overwhelming probability.

3 Attribute-Based Signatures: Definitions and Security

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 4 (ABS). *An* Attribute-Based Signature (ABS) scheme *is parameterized by a universe of possible attributes $\mathbb{A}$ and message space $\mathbb{M}$, and consists of the following four algorithms.*

- $\mathsf{ABS.TSetup}$ *(to be run by a* signature trustee*: Generates public reference information TPK.*
- $\mathsf{ABS.ASetup}$ *(to be run by an* attribute-issuing authority*): generates a key pair $APK, ASK \leftarrow \mathsf{ABS.ASetup}$.*
- $\mathsf{ABS.AttrGen}$*: On input $(ASK, \mathcal{A} \subseteq \mathbb{A})$, outputs a signing key $SK_{\mathcal{A}}$.*[3]
- $\mathsf{ABS.Sign}$*: On input $(PK = (TPK, APK), SK_{\mathcal{A}}, m \in \mathbb{M}, \Upsilon)$, where $\Upsilon(\mathcal{A}) = 1$, outputs a signature σ.*
- $\mathsf{ABS.Ver}$*: On input $(PK = (TPK, APK), m, \Upsilon, \sigma)$, outputs a boolean value.*

Definition 5 (Correctness). *We call an ABS scheme* correct *if for all $TPK \leftarrow \mathsf{ABS.TSetup}$, all purported APK, all messages m, all attribute sets $\mathcal{A}$, all signing keys $SK_{\mathcal{A}} \leftarrow \mathsf{ABS.AttrGen}(ASK, \mathcal{A})$, all claim-predicates Υ such that $\Upsilon(\mathcal{A}) = 1$, and all signatures $\sigma \leftarrow \mathsf{ABS.Sign}\big(PK = (TPK, APK), SK_{\mathcal{A}}, m, \Upsilon\big)$, we have $\mathsf{ABS.Ver}(PK = (TPK, APK), m, \Upsilon, \sigma) = 1$.*

We present two formal definitions that together capture our desired notions of security. Slightly weaker security requirements may also be useful for most applications, but we use the stronger ones because our constructions satisfy them and because they are much easier to work with.

For simplicity, we only present definitions for the simpler case of a single attribute-issuing authority. The definitions for multiple authorities are analogous, and we discuss this case in Section 5.

Definition 6 (Perfect Privacy). *An ABS scheme is* perfectly private *if, for all honestly generated $TPK \leftarrow \mathsf{ABS.TSetup}$, all purported APK, all attribute sets $\mathcal{A}_1, \mathcal{A}_2$, all $SK_1 \leftarrow \mathsf{ABS.AttrGen}(ASK, \mathcal{A}_1)$, $SK_2 \leftarrow \mathsf{ABS.AttrGen}(ASK, \mathcal{A}_2)$, all messages m, and all claim-predicates Υ such that $\Upsilon(\mathcal{A}_1) = \Upsilon(\mathcal{A}_2) = 1$, the distributions $\mathsf{ABS.Sign}(PK, SK_1, m, \Upsilon)$ and $\mathsf{ABS.Sign}(PK, SK_2, m, \Upsilon)$ are equal.*

In other words, the signer's privacy relies only on the signature trustee, and not the attribute-issuing authority. Even a malicious and computationally unbounded attribute-issuing authority cannot link a signature to a set of attributes or the signing key used to generate it.

We slightly overload notation and write $\mathsf{ABS.Sign}(ASK, m, \Upsilon)$ (i.e., with the attribute authority's private key ASK instead of PK and $SK_{\mathcal{A}}$) to denote the following procedure: first, run $SK_{\mathcal{A}} \leftarrow \mathsf{ABS.AttrGen}(ASK, \mathcal{A})$ for any arbitrary $\mathcal{A}$ satisfying Υ; then output the result of $\mathsf{ABS.Sign}(PK, SK_{\mathcal{A}}, m, \Upsilon)$. For convenience in the experiment below we use $\mathsf{ABS.Sign}(ASK, \cdot, \cdot)$ to generate signatures requested by the adversary. This is reasonable when the scheme satisfies perfect privacy, since any other way of letting the adversary obtain signatures will result in the same distribution.

[3] For simplicity, we treat the signing key as a monolithic quantity. However, in our construction the signing key consists of separate components for each attribute in $\mathcal{A}$, and the $\mathsf{ABS.Sign}$ algorithm needs only as much of $SK_{\mathcal{A}}$ as is relevant to the claim-predicate.

Definition 7 (Unforgeability). *An ABS scheme is* unforgeable *if the success probability of any polynomial-time adversary in the following experiment is negligible:*

1. *Run* $TPK \leftarrow \mathsf{ABS.TSetup}$ *and* $(APK, ASK) \leftarrow \mathsf{ABS.ASetup}$*. Give* $PK = (TPK, APK)$ *to the adversary.*
2. *The adversary is given access to two oracles:* $\mathsf{ABS.AttrGen}(ASK, \cdot)$ *and* $\mathsf{ABS.Sign}(ASK, \cdot, \cdot)$*.*
3. *At the end the adversary outputs* $(m^*, \Upsilon^*, \sigma^*)$*.*

We say the adversary succeeds if (m^*, Υ^*) *was never queried to the* $\mathsf{ABS.Sign}$ *oracle, and* $\mathsf{ABS.Ver}(PK, m^*, \Upsilon^*, \sigma^*) = 1$*, and* $\Upsilon^*(\mathcal{A}) = 0$ *for all* $\mathcal{A}$ *queried to the* $\mathsf{ABS.AttrGen}$ *oracle.*

Thus any signature which could not have been legitimately made by a *single* one of the adversary's signing keys is considered a forgery. Note that we do not consider it a forgery if the adversary can produce a *different* signature on (m, Υ) than the one he received from the signing oracle.

4 Constructing ABS Schemes

4.1 Credential Bundles

We introduce a new generic primitive called *credential bundles*, which we use in our ABS constructions. Credential bundles model the intuitive requirements of publicly verifiable attributes that resist collusion.

Definition 8 (Credential bundle scheme). *A* credential bundle scheme *is parameterized by a message space* $\mathbb{M}$*, and consists of the following three algorithms.*

- $\mathsf{CB.Setup}$*: Outputs a verification key* vk *and a secret key* sk*.*
- $\mathsf{CB.Gen}$*: On input* $(sk, \{m_1, \ldots, m_n\} \subseteq \mathbb{M})$*, outputs a tag* τ *and values* $\sigma_1, \ldots, \sigma_n$*.*
- $\mathsf{CB.Ver}$*: On input* $(vk, m, (\tau, \sigma))$*, outputs a boolean value.*

The scheme is correct *if, for all* $(\tau, \sigma_1, \ldots, \sigma_n) \leftarrow \mathsf{CB.Gen}(sk, m_1, \ldots, m_n)$*, we have* $\mathsf{CB.Ver}(vk, m_i, (\tau, \sigma_i)) = 1$ *for all* i*.*

Clearly by excluding some of the σ_i's from an existing bundle, one can generate a new bundle on a subset of attributes. Our main security definition requires that taking a subset of a *single* bundle is the only way to obtain a new bundle from existing bundles; in particular, attributes from several bundles cannot be combined.

Definition 9. *A credential bundle scheme is* secure *if the success probability of any polynomial-time adversary in the following experiment is negligible:*

1. *Run* $(vk, sk) \leftarrow \mathsf{CB.Setup}$*, and give* vk *to the adversary.*
2. *The adversary is given access to an oracle* $\mathsf{CB.Gen}(sk, \cdot)$*.*
3. *At the end the adversary outputs* $(\tau^*, (m_1^*, \sigma_1^*), \ldots, (m_n^*, \sigma_n^*))$*.*

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

From any plain digital signature scheme 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 (e.g., an email address). Conversely, when a credential bundle scheme is restricted to singleton sets of messages, its unforgeability definition is equivalent to normal digital signature unforgeability. Despite this equivalence under black-box reductions, the syntax of credential bundles more closely models our desired semantics for ABS.

4.2 A Framework for ABS

Our generic ABS construction for the case of a *single attribute authority* is given in Figure 1. The construction generalizes easily to the multiple attribute authority case (Section 5). At a high level, to sign a message m with claim-predicate Υ, the signer proves that she possesses either a credential bundle containing either sufficient attributes to satisfy Υ, or a "pseudo-attribute" identified with the pair (m, Υ). Only the signature trustee is capable of generating bundles involving pseudo-attributes (these are verified against the trustee's verification key tvk), but it never does so. Thus the proof is convincing that the signer satisfied Υ. However, in the security reduction, the pseudo-attribute provides a mechanism to bind the NIWI proof to a message and give simulated signatures. In the full version we prove the following:

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 $\mathbb{A} \cup \mathbb{A}'$, and let NIWI be a perfect NIWI proof of knowledge scheme. Our ABS construction is as follows:

ABS.TSetup: The signature trustee runs $crs \leftarrow \mathsf{NIWI.Setup}$ as well as $(tvk, tsk) \leftarrow \mathsf{CB.Setup}$ and publishes $TPK = (crs, tvk)$.

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

ABS.AttrGen$(ASK, \mathcal{A})$: Ensure that $\mathcal{A}$ contains no pseudo-attributes. Then output the result of $\mathsf{CB.Gen}(ask, \mathcal{A})$.

ABS.Sign$(PK, SK_{\mathcal{A}}, m, \Upsilon)$: Assume that $\Upsilon(\mathcal{A}) = 1$. Parse $SK_{\mathcal{A}}$ as $(\tau, \{\sigma_a \mid a \in \mathcal{A}\})$. Υ is a formula over formal variables $\mathbb{A}$. Define $\widetilde{\Upsilon} := \Upsilon \vee a_{m,\Upsilon}$, where $a_{m,\Upsilon} \in \mathbb{A}'$ is the pseudo-attribute associated with (m, Υ). Thus, we still have $\widetilde{\Upsilon}(\mathcal{A}) = 1$. Let $\{a_1, \ldots, a_n\}$ denote the attributes appearing in $\widetilde{\Upsilon}$. Let vk_i be avk if attribute a_i is a pseudo-attribute, and tvk otherwise. Finally, let $\varPhi[vk, m, \Upsilon]$ denote the following boolean expression:

$$\exists\, \tau, \sigma_1, \ldots, \sigma_n : \widetilde{\Upsilon}\Big(\big\{a_i \mid \mathsf{CB.Ver}(vk_i, a_i, (\tau, \sigma_i)) = 1\big\}\Big) = 1 \tag{1}$$

For each i, set $\hat{\sigma}_i = \sigma_{a_i}$ from $SK_{\mathcal{A}}$ if it is present, and to any arbitrary value otherwise (since then its value does not matter). Compute $\pi \leftarrow \mathsf{NIWI.Prove}\big(crs; \varPhi[vk, m, \Upsilon]; (\tau, \hat{\sigma}_1, \ldots, \hat{\sigma}_n)\big)$. Output π as the ABS signature.

ABS.Ver(PK, m, Υ, π): Output the result of $\mathsf{NIWI.Verify}(crs; \varPhi[vk, m, \Upsilon]; \pi)$.

Fig. 1. General framework for an ABS scheme

Theorem 1. *Given a NIWI argument of knowledge scheme and any secure credential bundle scheme (equivalently, any digital signature scheme), the construction in Figure 1 is a secure ABS scheme. Further, if the NIWI argument is perfectly hiding, the ABS scheme is perfectly private.*

4.3 Practical Instantiation 1

Our first practical instantiation uses Groth-Sahai proofs [18] as the NIWI component and Boneh-Boyen signatures [5] as the credential bundle component. One notable feature of this choice is that attributes in the scheme are simply Boneh-Boyen signatures on messages of the form "userid||attr".

This instantiation requires cyclic groups of prime order equipped with bilinear pairings (Section 2.1). The Groth-Sahai system can prove satisfiability of *pairing-product equations* in such groups, and the main challenge in this instantiation is expressing the logic of the claim-predicate and the Boneh-Boyen signature verification in this limited vocabulary. We identify $\mathbb{Z}_p^*$ with the universe of attributes, where p is the size of the cyclic group used in the scheme.[4]

Boneh-Boyen signatures. We briefly review the Boneh-Boyen digital signature scheme [6]. As before, we suppose there is a bilinear pairing $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, where $\mathbb{G}$ and $\mathbb{H}$ have prime order p, and where g is a generator of $\mathbb{G}$, and h is a generator of $\mathbb{H}$. The scheme, described below, is strongly unforgeable under the q-SDH assumpion (Definition 1).

- $\mathsf{DS.KeyGen}$**:** Choose random $b, c, d \leftarrow \mathbb{Z}_p$ and compute $B = g^b$, $C = g^c$, $D = g^d$. The verification key is $(B, C, D) \in \mathbb{G}^3$, and the signing key is $(b, c, d) \in (\mathbb{Z}_p)^3$.
- $\mathsf{DS.Sign}(sk, m \in \mathbb{Z}_p)$**:** Choose random $r \leftarrow \mathbb{Z}_p$; output $\sigma = \left(h^{1/(b+cm+dr)}, t\right) \in \mathbb{H} \times \mathbb{Z}_p$.
- $\mathsf{DS.Ver}(vk, m, \sigma = (S, r))$**:** Output 1 if $e(BC^m D^r, S) = e(g, h)$, and 0 otherwise.

Expressing the Non-Interactive Proof using Pairing Equations. We use the notation introduced in Figure 1. We must show how the statement $\varPhi[vk, m, \varUpsilon]$ (equation 1) can be efficiently encoded in the Groth-Sahai system when the credential bundles use Boneh-Boyen signatures.

Groth-Sahai proofs work by first giving a commitment to the values of the witness, and then proving that the commited values satisfy given pairing equations. Suppose we commit to a group element Z (where the group $\mathbb{G}$ or $\mathbb{H}$ will be clear from context), then we will let $\langle Z \rangle$ denote the formal variable corresponding to that commitment. Thus, we express the statements to be proven as pairing equations whose formal variables we will write in the $\langle Z \rangle$ notation.

[4] More precisely $\mathbb{A} \cup \mathbb{A}' \subseteq \mathbb{Z}_p^*$ where $\mathbb{A}'$ is the universe of pseudo-attributes. As is standard, the universe of (pseudo-)attributes can be extended to $\{0,1\}^*$ by applying a collision-resistant hash with range $\mathbb{Z}_p^*$.

Suppose the modified predicate $\widetilde{\Upsilon}$ has a canonical monotone span program $\mathbf{M}$ of size $\ell \times t$, where the ith row corresponds to the $a(i)$-th attribute mentioned in $\widetilde{\Upsilon}$. To establish $\Phi[vk, m, \Upsilon]$, we prove the following equation, which implies it:

$$\exists\, \tau, \sigma_1, \ldots, \sigma_n, v_1, \ldots, v_n : \boldsymbol{v}\mathbf{M} = [1, 0, \ldots, 0] \\ \wedge \bigwedge_{i=1}^{\ell} \Big[v_i \neq 0 \Rightarrow \mathsf{CB.Ver}(vk, a_{a(i)}, (\tau, \sigma_{a(i)})) = 1 \Big]$$

Then, in addition to τ, $\{\sigma_i\}$, we will have the signer commit to the vector $\boldsymbol{v}$ which can be canonically computed from his satisfying assignment of $\widetilde{\Upsilon}$.

This new boolean expression is a conjunction of two kinds of clauses: The first has the form $\exists \boldsymbol{v} : \boldsymbol{v}\mathbf{M} = [1, \ldots, 0]$. To prove it, we commit to the values g^{v_i} and prove the following pairing equations (for each $j \in [t]$):

$$\prod_{i=1}^{\ell} e(\langle g^{v_i} \rangle, h^{\mathbf{M}_{i,j}}) = \begin{cases} e(g, h) & \text{if } j = 1 \\ e(g^0, h) & \text{otherwise} \end{cases}$$

The other clauses have the form $\exists\, \tau, \sigma, v : \big[v \neq 0 \Rightarrow \mathsf{CB.Ver}(vk, m, (\tau, \sigma)) = 1 \big]$. When we use Boneh-Boyen signatures as the instantiation of credential bundles, these clauses can be simplified to

$$\exists\, \tau, \sigma, v : \big[v \neq 0 \Rightarrow \mathsf{DS.Ver}(vk, \tau \| m, \sigma) = 1 \big]$$

where $\mathsf{DS.Ver}$ is the Boneh-Boyen signature verification.

It is crucial that the proof is a proof *of knowledge*, so the simulator can extract the credential bundles. Thus we commit to τ and r *bitwise*, since they are elements of $\mathbb{Z}_p$and could not otherwise be efficiently extracted in the Groth-Sahai scheme. In this way, the extractor can extract the bits and reconstruct the entire witness τ and r.[5] Let $(\tau, \sigma = (S, r), v)$ be a witness to the above expression. Express τ bitwise as $\tau = \sum_i \tau_i 2^i$. Then $\tau \| m$ may be identified with a number $m2^{|\tau|} + \sum_i \tau_i 2^i$. Similarly, interperet r bitwise as $r = \sum_i r_i 2^i$.

Using the same notation as before, we can prove satisfiability of the clause as follows. We commit to each r_i and τ_i in both groups, as $g^{r_i}, h^{r_i}, g^{\tau_i}, h^{\tau_i}$, and then prove that each is indeed a single bit, using the following pairing equations for all i:

$$\begin{aligned} e(\langle g^{r_i} \rangle, h) &= e(g, \langle h^{r_i} \rangle); & e(\langle g^{\tau_i} \rangle, h) &= e(g, \langle h^{\tau_i} \rangle); \\ e(\langle g^{r_i} \rangle, \langle h^{r_i} \rangle) &= e(\langle g^{r_i} \rangle, h); & e(\langle g^{\tau_i} \rangle, \langle h^{\tau_i} \rangle) &= e(\langle g^{\tau_i} \rangle, h). \end{aligned}$$

Next, observe that the pairing equation $e(BC^{\tau \| m} D^r, S^v) = e(g^v, h)$ is logically equivalent to the expression $v \neq 0 \Rightarrow \mathsf{DS.Ver}(vk, \tau \| m, (S, r)) = 1$, which we need to prove. However, the prover cannot directly compute $BC^{\tau \| m} D^r$ or S^v given

[5] We remark that the proof need not be a proof of knowledge with respect to $\boldsymbol{v}$, so it was safe to use these values directly in $\mathbb{Z}_p$.

the committed values. Thus the prover commits to some additional intermediate values $S^v \in \mathbb{H}$ and $C^\tau, D^r \in \mathbb{G}$, and proves the following equations:

$$\begin{aligned} &e(\langle D^r \rangle, h) = \textstyle\prod_i e(D^{2^i}, \langle h^{r_i} \rangle); \qquad e(\langle g^v \rangle, \langle S \rangle) = e(g, \langle S^v \rangle); \\ &e(\langle C^\tau \rangle, h) = \textstyle\prod_i e(C^{2^i}, \langle h^{\tau_i} \rangle); \\ &e(\langle g^v \rangle, h) = e(BC^{2^{|\tau|}m}, \langle S^v \rangle)\, e(\langle C^\tau \rangle, \langle S^v \rangle)\, e(\langle D^r \rangle, \langle S^v \rangle). \end{aligned}$$

Note that since m and $|\tau|$ are public, all the coefficients in these equations can be publicly computed. This completes the description of how we encode the required logic into the Groth-Sahai proof system.

There are two instantiations of the Groth-Sahai proof system over prime order groups, based on the DLIN and SXDH assumptions, both of which are suitable for our purposes. Using these we obtain the following (a more detailed analysis of the efficiency is given in the full version).

Theorem 2. *Under the q-SDH and either DLIN or SXDH assumptions, there is an ABS scheme supporting claim-predicates represented as monotone span programs, with signatures consisting of $O(ks)$ group elements, where s is the size of the monotone span program.*

4.4 Practical Instantiation 2

We can also instantiate our framework using the same approach as above, but with the signature scheme of Waters [30]. Signatures in Waters' scheme do not include any elements of $\mathbb{Z}_p$. This fact allows us to avoid the inefficiency of committing to many components of the Boneh-Boyen signatures in a bitwise fashion. Furthermore, Waters signatures are secure under the much weaker BDH assumption, which is implied by the assumptions required for Groth-Sahai proofs. Thus this instantiation does not require the additional q-SDH assumption. However, as a tradeoff, the Waters instantiation requires larger public parameters: a linear (in the security parameter) number of group elements, not the constant number of group elements needed by the Boneh-Boyen instantiation.

The details of this instantiation follow a similar approach as the previous one, incorporating the verification equation of the Waters signature. We refer the reader to the full version for the complete details.

Theorem 3. *Under either the DLIN or SXDH assumptions, there is an ABS scheme supporting claim-predicates represented as monotone span programs, with signatures consisting of $O(k + s)$ group elements, where s is the size of the monotone span program.*

4.5 Practical Instantiation 3

We now present an ABS scheme which is our most practical. Signatures in the scheme consist of *exactly* $s + 2$ group elements, where s is the size of the claim-predicate's monotone span program. This scheme does not use the Groth-Sahai proof system; we use our own randomization techniques to blind the attributes that are used in signing. One additional advantage of avoiding a NIZK proof system is that the privacy of the signers is provided even against a malicious signature trustee; in contrast the above NIZK-based constructions rely on the signature trustee to set up a common reference string honestly.

Our approach is motivated by the construction of mesh signatures [9], but incorporates the efficient credential bundles of the previous construction, as well as the concept of "pseudo-attributes" to bind a message to the signature. In the full version we give a high-level motivation of the details of this scheme. Below we give a description of the construction:

This construction supports all claim-predicates whose monotone span programs have width at most t_{max}, where t_{max} is an arbitrary parameter. We treat $\mathbb{A} = \mathbb{Z}_p^*$ as the universe of attributes, where p is the size of the cyclic group used in the scheme.[6]

ABS.TSetup: Choose suitable cyclic groups G and H of prime order p, equipped with a bilinear pairing $e : G \times H \to G_T$. Choose a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p^*$. Choose random generators: $g \leftarrow G$; $\quad h_0, \dots h_{t_{\mathsf{max}}} \leftarrow H$. The trustee public key is $TPK = (G, H, \mathcal{H}, g, h_0, \dots, h_{t_{\mathsf{max}}})$.

ABS.ASetup: Choose random $a_0, a, b, c \leftarrow \mathbb{Z}_p^*$ and set:

$$C = g^c; \qquad A_0 = h_0^{a_0}; \qquad A_j = h_j^a \text{ and } B_j = h_j^b \quad (\forall j \in [t_{\mathsf{max}}]).$$

The master key is $ASK = (a_0, a, b)$. The public key APK is $(A_0, \dots, A_{t_{\mathsf{max}}}, B_1, \dots, B_{t_{\mathsf{max}}}, C)$

ABS.AttrGen: On input ASK as above and attribute set $\mathcal{A} \subseteq \mathbb{A}$, Choose random generator $K_{\mathsf{base}} \leftarrow G$. Set:

$$K_0 = K_{\mathsf{base}}^{1/a_0}; \qquad K_u = K_{\mathsf{base}}^{1/(a+bu)} \quad (\forall u \in \mathcal{A})$$

The signing key is then $SK_{\mathcal{A}} = (K_{\mathsf{base}}, K_0, \{K_u \mid u \in \mathcal{A}\})$.

ABS.Sign: On input $(PK, SK_{\mathcal{A}}, m, \Upsilon)$ such that $\Upsilon(\mathcal{A}) = 1$, first convert Υ to its corresponding monotone span program $\mathbf{M} \in (\mathbb{Z}_p)^{\ell \times t}$, with row labeling $u : [\ell] \to \mathbb{A}$. Also compute the vector $\boldsymbol{v}$ that corresponds to the satisfying assignment $\mathcal{A}$. Compute $\mu = \mathcal{H}(m \| \Upsilon)$.

Pick random $r_0 \leftarrow \mathbb{Z}_p^*$ and $r_1, \dots r_\ell \leftarrow \mathbb{Z}_p$ and compute:

$$Y = K_{\mathsf{base}}^{r_0}; \qquad S_i = (K_{u(i)}^{v_i})^{r_0} \cdot (Cg^{\mu})^{r_i} \quad (\forall i \in [\ell]);$$

$$W = K_0^{r_0}; \qquad P_j = \prod_{i=1}^{\ell} (A_j B_j^{u(i)})^{\mathbf{M}_{ij} \cdot r_i} \quad (\forall j \in [t]).$$

We note that the signer may not have $K_{u(i)}$ for every attribute $u(i)$ mentioned in the claim-predicate. But when this is the case, $v_i = 0$, and so the value is not needed. The signature is $\sigma = (Y, W, S_1, \dots, S_\ell, P_1, \dots, P_t)$.

[6] As always, the universe of attributes can be further extended to $\{0,1\}^*$ by applying a collision-resistant hash having range $\mathbb{Z}_p^*$. For simplicity of presentation, we do not include this modification.

ABS.Ver: On input $(PK, \sigma = (Y, W, S_1, \ldots, S_\ell, P_1, \ldots, P_t), m, \Upsilon)$, first convert Υ to its corresponding monotone span program $\mathbf{M} \in (\mathbb{Z}_p)^{\ell \times t}$, with row labeling $u : [\ell] \to \mathbb{A}$. Compute $\mu = \mathcal{H}(m \| \Upsilon)$. If $Y = 1$, then output reject. Otherwise check the following constraints:

$$e(W, A_0) \stackrel{?}{=} e(Y, h_0)$$

$$\prod_{i=1}^{\ell} e\Big(S_i, (A_j B_j^{u(i)})^{\mathbf{M}_{ij}}\Big) \stackrel{?}{=} \begin{cases} e(Y, h_1)\, e(Cg^\mu, P_1), & j = 1 \\ e(Cg^\mu, P_j), & j > 1, \end{cases}$$

for $j \in [t]$. Return accept if all the above checks succeed, and reject otherwise.

We defer the detailed proof of security (carried out in the generic group model) to the full version.

Theorem 4. *In the generic group model, there is an ABS scheme supporting claim-predicates represented as monotone span programs, with signatures consisting of $s + 2$ group elements, where s is the size of the monotone span program.*

5 Multiple Attribute-Authorities

Our first two intantiations of ABS (indeed, our general framework) can be easily extended for use in an environment with multiple attribute-issuing authorities. Except in a centralized enterprise setting, a single user would acquire her attributes from different authorities (e.g., different government agencies, different commercial services she has subscribed to, different social networks she is registered with and so on). These different attribute authorities may not trust each other, nor even be aware of each other. Indeed, some attribute authorities may be untrustworthy, and this should not affect the trustworthiness of attributes acquired from other authorities, or of ABS signatures involving trustworthy attributes.

Apart from these mutually distrusting attribute authorities, we still require a (possibly separate) *signature trustee* to set up the various public parameters of the ABS signature scheme itself. A signature trustee does not have to trust any attribute authority. The attribute authorities use only the public keys from the signature trustee. As long as the signature trustee is trusted, then the ABS signatures are secure and leak no information about the identity or attributes of the signer. The only requirement for compatibility among attribute authorities is that they all have a mechanism for agreeing on a user's userid (say, an email address) so that a user's bundle of credentials may contain compatible attributes from several authorities.

Finally, the claim-predicate in the ABS signature must carry the identity of the attribute-authorities who *own* the various attributes (possibly as meta-data attached to the attribute description). Given this information, the statement proven in the non-interactive proof can be modified to refer to the appropriate digital signature verification keys corresponding to each attribute, including the pseudo-attribute. If one attribute authority's signatures are compromised, then an ABS verifier should not give much importance to attributes from that authority. However, the ABS signatures themselves are still valid (in that they indeed attest to the given claim-predicate being satisfied) as long as the trustee is uncorrupted.

6 Applications

We identify several natural applications of ABS schemes:

Attribute-based messaging. Attribute-Based Messaging, or ABM, (e.g., [4]) provides an example of a quintessential attribute-based system. In an ABM system, messages are addressed not by the identities of the recipients, but by a predicate on users' attributes which the recipients must satisfy. The users need not be aware of each other's identities or attributes. To provide *end-to-end* message privacy (against users whose attributes do not satisfy the sender's policy), one can use *ciphertext-policy attribute-based encryption*, as proposed by Bethencourt, Sahai and Waters [3]. However, there was no satisfactory way to achieve *authentication* (i.e., for the receiver to verify that the sender also satisfied a particular policy) in an ABM system until now. Existing cryptographic technology, including certificates and mesh signatures, would not provide an adequate level of anonymity for the senders while simultaneously preventing collusions.

In a typical ABM system, a certain degree of authorization is required to send messages to certain groups of users. That is, an attribute-based access control mechanism must decide whether to allow a messaging attempt from a sender, depending on both the attributes of the sender and the attribute-based address attached to the message. ABS can be used to authenticate a sender to the ABM system itself (as opposed to the scenario above, where the sender was authenticating to the message recipient). As the messaging system can publicly verify the ABS signature, this solution eliminates the need for the messaging system to query the attribute database to determine the sender's authorization. Indeed, the messaging system need not know the sender's identity at all.

Finally, because our construction is so readily suited for multi-authority settings, ABS is a natural choice for inter-domain ABM systems. However, there are many engineering and cryptographic challenges involved in other aspects of a truly inter-domain ABM system. For example, Chase's proposal [11] for multi-authority attribute-based encryption (originally for the schemes in [27,16], but can be extended to the one in [3]) requires all the attribute-authorities to share secret keys with a central authority, thereby requiring the central authority to trust all the attribute authorities. In contrast, our ABS system requires no such trust between the signature trustee and attribute authorities. As such, ABS is much better suited to practical inter-domain attribute-based systems than its encryption counterparts.

Attribute-based authentication and trust-negotiation. ABS can also be used as a more general fine-grained authentication mechanism. For instance, a server can publish its access policy for a particular resource along with its encryption public key. When a client wishes to access the resource, the server issues a random challenge string. The client can then generate a session key for (private-key) communication, generate an ABS signature of $(challenge, sessionkey)$ under the server's policy, and send these to the server encrypted under the server's public key. Thereafter, the client and server can communicate using the shared session key. This simple protocol is robust even against a man in the middle.

This technique can be extended to multiple rounds as a simple *trust negotiation* protocol, in which two parties progressively reveal more about their attributes over several rounds of interaction. Several recent works also consider cryptographic approaches to trust negotiation that give more privacy to users than is achieved when they simply take turns revealing their attributes [25,15]. Instead of these techniques, ABS can provide a sophisticated way to reveal partial information about one's attributes that is natural for this setting. Being able to bind a message to such a proof about one's attributes, as ABS permits, also allows one to protect the trust negotiation from outside attack, using an approach as above. At each step of the negotiation, the active party can choose an "ephemeral key" for secure (private-key) communication and sign it using ABS. This approach prevents a man-in-the-middle attacks by an adversary who has enough attributes to intercept the first few steps of the negotiation.

Leaking secrets. The classical application for which the notion of ring-signatures was developed by Rivest, Shamir and Tauman [26] is "leaking secrets," that we used as the motivating example in the opening of this paper. Ring signatures support only claim-predicates which are disjunctions. Mesh signatures are an extension of this concept which allow more sophisticated claim-predicates, but permit multiple parties to pool their attributes (atomic signatures). This is not necessarily the intended semantics in natural secret-leaking environment. ABS, on the other hand, provides the semantics that a *single* user (not a coalition) whose attributes satisfy the stated predicate attests to the secret.

References

1. 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)
2. Bellare, M., Namprempre, C., Neven, G.: Security proofs for identity-based identification and signature schemes. Journal of Cryptology 22(1), 1–61 (2009); Preliminary version appeared in Eurocrypt 2004
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
4. Bobba, R., Fatemieh, O., Khan, F., Gunter, C.A., Khurana, H.: Using attribute-based access control to enable attribute-based messaging. In: ACSAC, pp. 403–413. IEEE Computer Society, Los Alamitos (2006)
5. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
6. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
7. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
8. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. SIAM J. Comput. 32(3), 586–615 (2003)
9. Boyen, X.: Mesh signatures. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 210–227. Springer, Heidelberg (2007)
10. Boyen, X., Waters, B.: Compact group signatures without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 427–444. Springer, Heidelberg (2006)
11. Chase, M.: Multi-authority attribute based encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)

12. Chase, M., Chow, S.S.M.: Improving privacy and security in multi-authority attribute-based encryption. In: Al-Shaer, E., Jha, S., Keromytis, A.D. (eds.) ACM Conference on Computer and Communications Security, pp. 121–130. ACM, New York (2009)
13. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. ACM Commun. 28(10), 1030–1044 (1985)
14. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991)
15. Frikken, K.B., Li, J., Atallah, M.J.: Trust negotiation with hidden credentials, hidden policies, and policy cycles. In: NDSS. The Internet Society, San Diego (2006)
16. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) ACM Conference on Computer and Communications Security, pp. 89–98. ACM, New York (2006)
17. 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)
18. 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)
19. Guo, S., Zeng, Y.: Attribute-based signature scheme. In: International Conference on Information Security and Assurance, pp. 509–511. IEEE, Los Alamitos (2008)
20. Katz, J., Ostrovsky, R., Rabin, M.O.: Identity-based zero-knowledge. In: Blundo, C., Cimato, S. (eds.) SCN 2004. LNCS, vol. 3352, pp. 180–192. Springer, Heidelberg (2005)
21. Khader, D.: Attribute based group signature with revocation. Cryptology ePrint Archive, Report 2007/241 (2007), http://eprint.iacr.org/2007/241
22. Khader, D.: Attribute based group signatures. Cryptology ePrint Archive, Report 2007/159 (2007), http://eprint.iacr.org/2007/159
23. 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, New York (2010)
24. Li, J., Kim, K.: Attribute-based ring signatures. Cryptology ePrint Archive, Report 2008/394 (2008), http://eprint.iacr.org/2008/394
25. Li, N., Du, W., Boneh, D.: Oblivious signature-based envelope. Distributed Computing 17(4), 293–302 (2005)
26. 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)
27. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
28. Shahandashti, S.F., Safavi-Naini, R.: Threshold attribute-based signatures and their application to anonymous credential systems. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 198–216. Springer, Heidelberg (2009)
29. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
30. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
31. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290 (2008), http://eprint.iacr.org/2008/290