Attribute-Based Encryption: Using Identity-Based Encryption for Access Control

Ari Juels and Michael Szydlo

RSA Laboratories
Bedford, MA 01730
{ajuels,mszydlo}@rsasecurity.com

17 June 2004

Abstract. We introduce an access-control technique that we refer to as *attribute-based encryption* (ABE). An extension of the cryptographic technique of identity-based encryption (IBE), our proposed ABE scheme can serve as the basis of an access-control architecture in which entities require no interaction with a trusted authority in order to gain access to sensitive data.We show how to construct any access-control policy for ABE that is expressible as monotone boolean formulae on variables describing the possession of attributes by a requesting entity. This encompasses a broad range of the policy formulations of common interest. Our system is practical: Indeed, its most attractive feature is architectural simplicity.

Key words: access control, attribute, elliptic curve, identity-based encryption

1 Introduction

The goal of an access-control infrastructure is to distribute resources to a community of users in strict accordance with individually assigned privileges. The use of *attributes* represents a natural and flexible approach to this problem. An attribute may be viewed as a form of membership among a group of entities or *principals* in a given system. For example, among the attributes assignable to a human principal in any system is that of being at least eighteen years of age. When an attribute is defined primarily in terms of a natural cluster of privileges, rather than a natural cluster of entities, it is often referred to as a *rôle*. An example of a rôle in this sense, for instance, would be United States citizenship. While access-control lists still represent the predominant form of rights-management in commercial security software, access-control based on attributes or rôles is much favored in the academic literature and gaining ground in real-world environments. This approach is often referred to as *rôle-based access control*, as a result of emphasis on rôles as a preferred form of attribute. We use the broader term *attribute-based* access control in this paper.

Existing attributed-based access-control architectures loosely fall into one of two categories: (1) *Centralized* systems, where attribute assignments are held, managed, and referenced by way of the databases of trusted entities, and (2) *Attribute-certificate-based* systems, in which an *attribute authority* (AA) issues attestations to principals of their assigned attributes, often in the form of digitally signed statements. (We describe attribute certificates in more detail in section **??**.) In both types of architecture, an entity seeking access to a resource must interact with some trusted authority (TA). For example, suppose that Alice wishes to post a document to a Web page in such a way that it is available to anyone in a European branch of her company, as indicated by an attribute labeled *Euro*. Alice would have to rely on a TA to release her document only to a requesting party that has demonstrated possession of attribute *Euro*. Thus, if Bob wanted to obtain access to the document, he would need in a centralized system to appeal to an appropriate TA with which he is registered as possessing attribute *Euro*; in an attribute-certificate-based system, Bob would need to present an attribute certificate for *Euro* to the TA.

In this paper, we present a concept that we call *attribute-based encryption* (ABE). We regard ABE as a natural extension of the cryptographic technique known as *identity-based encryption* (IBE), newly introduced in a practical form by Boneh and Franklin. By means of ABE, it is possible to envision a third category of access-control architecture involving significantly less interactivity than a centralized or attribute-certificate-based system. In an access-control system based on ABE, Alice would be able to post her document publicly or e-mail it to everyone in such a way that only those possessing attribute *Euro* could decrypt it. In particular, authorized receivers could decrypt the document *without interacting with a TA*.

Our proposed ABE approach to access control is closely aligned in its essentials to attribute-certificate-based systems. Attributes take the form of credentials issued by an authority that is in essence an AA. The special feature of identity-based encryption is its ability to create "implicit" certificates, that is, certificates for which public keys assume the form of arbitrary text strings, and therefore do not need to be looked up. ABE may similarly be viewed as an embodiment of "implicit" attribute certificates relative to which encryption may take place without any lookup or interaction. Thus, resource protection may take place in an ABE without interaction with a TA, resulting in a simplified access-control architecture – particularly in store-and-forward environments. An ABE architecture also has the feature that the burden of computation is placed on ordinary entities rather than a TA, relieving the potential bottlenecks of systems that place heavy reliance on a centralized access-control authority.

An access-control policy is a specification of which attributes an entity must possess access a resource. Our focus in this paper is on access to sensitive data. We show toward this end how ABE may be used to implement any access-control policy expressible as a monotone boolean formula on attribute values, i.e., on a set of boolean variables each of which denotes the possession or lack of a given attribute by the requesting entity. This encompasses a broad swath of the policy formulations of general interest for attribute-based access control.[1] For example, suppose that *Senior* and US are labels respectively for attributes denoting senior management status and membership in the US branch of Alice's company. Alice may use ABE to post a document to a public location in such a way that it may be decrypted only by employees in the European branch of her company, or by employees who are senior managers in the United States branch. This policy is expressible, using rather loose notation, by the following simple monotone boolean statement: $Euro \vee (Senior \wedge US)$.

A potential drawback to our proposed ABE approach is the computational overhead it imposes on entities established protected resources or seeking to access such resources. For a policy expressed as a boolean formula on k attributes, this overhead is approximately equivalent to a k elliptic-curve multiplications. For typical cryptographic security parameterizations, however, the computational cost of a single elliptic-curve multiplication is much less than for an RSA signing operation (by about an order of magnitude [11]. Thus it may be expected that for ordinary policies, the computational requirements for the ABE operations will be quite reasonable, requiring only a fraction of a second on an ordinary workstation.

1.1 Organization

In section 2, we provide some details on the workings of attribute certificates and IBE. We present the details of our proposed ABE scheme in section **??**, and give some examples of its application to store-and-forward environments like e-mail in section 4. In section 5, we offer a brief analysis of the efficiency of ABE in a practical environment and discuss some implementation considerations.

2 Background

2.1 Attribute certificates

Generally speaking, an attribute certificate is statement binding to a principal a string representing an attribute, as defined above. An attribute certificate typically accompanies a public-key identity certificate. While used only sparsely today, attribute certificates have a basis for support in several maturing standards of influence, most notably in ISO/IEC 9594-8 (ITU-T X.509) [1] and ANSI X9.45 [2]. Moreover, use of attribute certificates promises to proliferate in the wake of broadening public-key infrastructure (PKI) deployment. The purpose of an attribute certificate is to assert a privilege or set of privileges. The issuer of an identity certificate is referred to as a certificate authority (CA). As explained

[1] Note that non-monotone boolean formulae yield only degenerate expressions of policy in the context of attribute-based access control. Such formulae assign truth values based on the *absence* of attributes. The only way for an entity to demonstrate the absence of a given attestation, however, is to demonstrate the *presence* of an attestation asserting this absence.

above, the issuer of an attribute certificate is referred to as an attribute authority, abbreviated AA, and may be an entity distinct from the CA in a given system. For a good overview, the reader is directed to the survey paper of Linn and Nystrom [9].

In most deployments, an attribute certificate is linked in some manner to an identity certificate, which serves as the basis for authentication of the principal. Thus, for example, if Long John Silver works in food services, he may possess an identity certificate IC_{LJS} bound to the principal name "Long John Silver", along with an attribute certificate AC_{LJS} that is linked to IC_{LJS} and asserts membership in the food services "group". To prove membership in food services, Long John Silver need merely authenticate himself using IC_{LJS} and provide AC_{LJS} as supplementary data. If, for instance, the system allows access by all food service personnel to the corporate log of beverage deliveries, then Long John Silver can gain such access by asserting his privilege as a member of the food services group.

While attribute certificates are normally used in conjunction with identity certificates, there is no reason why this need be the case. Indeed, given an accompanying private key or protected itself as a secret (as for SAML assertions []), an attribute certificates may serve as a freestanding proof of the attribute of its possessor. This approach is often exploited for its privacy-enhancing benefits, as it enables the dissociation of attributes from identities. For example, blind digital cash systems, e.g., [7], employ certificates that may be thought of as asserting financial priviliges without revealing identities. We refer to attribute certificates in this mode of use as *freestanding*. Freestanding attribute certificates, as we discuss below, offer an especially appealing range of properties in an ABE system.

Standards like ISO/IEC 9594-8 (ITU-T X.509) do not provide formal specification of the range of policies that may be expressed in a security infrastructure employing attribute certificates. This can obviously be quite broad, combining the privileges possessed by the claimant with a host of factors, such the type of access being requested for a sensitive object, the system time, and so forth. Additionally, a policy can be crafted so as to control access, in principle, to any system resource. Our focus in this paper is on access to data resources, but since a data resource may itself consist of a cryptographic key used to access other resources, the notions we present here are extensible to a variety of situations.

2.2 Identity-based encryption

As mentioned above, the new cryptographic tool which we employ in this paper is *identity-based encryption* (IBE). An IBE scheme is a cryptosystem in which an arbitrary text string may serve as a public key. Names, dates, and email addresses, for example, may serve as public keys in an IBE system. This feature is valuable because it reduces the interaction and infrastructure required to send data securely. In particular, is possible to perform encryption under the public key of a selected entity without performing a certificate lookup or other interaction. The utility of IBE as a means to simplify PKI was noted in 1984 by Shamir, who challenged cryptographers to find a practical IBE scheme. In 2001, the first *efficient* IBE scheme was discovered by Boneh and Franklin, based on pairings defined on elliptic curves. (A more-or-less contemporaneous but somewhat less efficient scheme was devised by Cocks [8].)

Boneh *et al.* have demonstrated the practical viability of the Boneh-Franklin scheme by deploying a secure email system [4], (see `http://identicrypt.com`). This works as follows. Suppose that Alice wishes to send Bob a message, and has IBE software installed on her machine. She simply encrypts her message under the key "bob@company.com" and sends it to Bob. If Bob has ever used the system before he will have a corresponding private key, and can immediately decrypt the message. Otherwise Bob communicates with an entity that we call the *Public Key Generator* (PKG). He obtains any necessary software, and after a successful authentication in accordance with the policy of the PKG, receives the private key associated with the string "bob@company.com". The main attraction of this approach is that Alice does not have to interact with Bob or wait for him to install software or even receive his private key before she sends him a message. Because the PKG can compute any private key, there is an inherent key-escrow aspect to the system. This is something of a drawback in an e-mail encryption system, but aligns very appropriately with the trust model in our ABE system, wherein the PKG serves the rôle of an AA.

More concretely, an IBE system consists of four randomized algorithms, which we roughly summarize as follows:

- setup: The function setup is executed by the PKG on input consisting of a security parameter k. The output includes $params$, a set of data comprising a message space, a ciphertext space, and other parameters to be published by the PKG. Additionally, setup returns a private value $master - key$ to be retained by the PKG, and used for generation of private decryption keys.
- key-gen: Given input $params$, $master - key$, and some string (public-key) ID, the function key-gen returns d_{ID}: the private key corresponding to ID.
- encrypt: Given input $params$, the string (public-key) ID, and a message M, the function encrypt yields a ciphertext C.
- decrypt: Given input $params$, the string (public-key) ID, and the correct corresponding private key d_{ID}, the function decrypt returns the message M.

The algorithms are implemented by means of algebraic operations over a pre-selected elliptic curve. The private key associated with a given public key takes the the form of a single point on this curve, and is thus quite compact – roughly twenty bytes in length. We provide a sketch of the basic technique in the appendix, and refer the reader to [6] for full details and security definitions. These technical details, however, are not required for our discussion of ABE in what follows. Also described in [6] are natural and efficient ways of distributing $master - key$ among multiple entities, thus permitting a distributed AA to be realized in our proposals below.

3 Attribute Based Encryption

Having explained the mechanics of attribute certificates and IBE, we can now describe our ABE scheme. We describe two approaches with different security and use characteristics. The first, which we call *identity-bound* ABE, harmonizes with the more canonical use of attribute certificates as adjuncts to identity certificates. It has the drawback of requiring the transmitter of sensitive data to specify the identities of receivers in advance. The second approach, which we call *freestanding ABE*, makes use instead of freestanding attribute certificates as defined above. In other words, this latter approach treats attributes as distinct from the identity of their principal. A benefit of this feature is that access-control policies may be specified without reference to the identities of principals. A drawback is the transferrability of credentials, and the consequent difficult of attribute revocation. In both cases, the AA distributes IBE private keys corresponding to attribute strings; in other words, the AA serves as a PKG. In brief, an ABE system differs from a standard system employing attribute certificates in that the AA issues private keys to represent attributes, rather than signed statements or certificates.

To achieve the broadest possible scope in our descriptions, we consider the application of ABE to control distribution of a cryptographic key κ. This key κ might serve any desired purpose: It may be a symmetric key for decryption of a document, a private signing key, etc. For simplicity of presentation, we assume that κ is drawn from a large, publicly specified group $\mathcal{F}$. We further assume a single AA that has published public information `params` for an IBE scheme.

3.1 Identity-bound ABE

Consider an attribute denoted by identifier a and a principal denoted by identifier $\mathcal{P}$. We may represent a binding or assignment of the attribute a to principal $\mathcal{P}$ by the string "$\mathcal{P} \parallel a_i$". To assert this binding in our ABE scheme, then, a private IBE key is employed that corresponds to the public IBE key "$\mathcal{P} \parallel a_i$". In other words, we may think of the assignment of attribute a to principal $\mathcal{P}$ as represented in an "implicit" attribute certificate with the corresponding private key:

$$Cert_{\mathcal{P},a} = d_{``\mathcal{P} \parallel a"}. \qquad (1)$$

The AA, then, may transmit this private key to principal $\mathcal{P}$ in order to certify the attribute/identity binding "$\mathcal{P} \parallel a_i$".

Let $C_{\mathcal{P}}$ denote the public-key identity certificate of $\mathcal{P}$, and let $PK_{\mathcal{P}}$ be the corresponding public key. This certificate may be of a conventional form, e.g., an X.509 certificate. Alternatively, so as to eliminate directory lookups of identity certificates in addition to attribute certificates, $C_{\mathcal{P}}$ might itself be an "implicit" certificate generated by a CA using its own IBE parameters. In our ABE scheme, a resource owner who wishes to permit access by $\mathcal{P}$ to κ only under the condition that $\mathcal{P}$ possesses attribute a does the following. She performs a perfect splitting of κ into two shares $\kappa_1 \in \mathcal{F}$ and $\kappa_2 \in \mathcal{F}$ by, e.g., selecting κ_1 uniformly at random from $\mathcal{F}$ and letting $\kappa = \kappa_1 + \kappa_2$.. Let $E_{PK}[m]$ denote encryption under a suitable cryptosystem of plaintext m under either a conventional or IBE public key PK. The resource owner then constructs the following ciphertext:

$$c = (E_{PK_{\mathcal{P}}}[\kappa_1], E_{\text{``}\mathcal{P} \,\|\, a\text{''}}[\kappa_2]). \tag{2}$$

This ciphertext c may be transmitted to $\mathcal{P}$ by any desired means, e.g., e-mail or a bulletin board. It may be seen that only $\mathcal{P}$ should be able to decrypt κ_1, and that $\mathcal{P}$ can only decrypt κ_2 given assignment of attribute a, i.e., possession of the "implicit" certificate $Cert_{\mathcal{P},a}$. Thus the ciphertext c implements the desired access-control restriction on κ. We show how to implement more sophisticated access-control policies below.

Remark: A simpler alternative to identity-bound ABE is just to construct ciphertext $c = E_{\text{``}\mathcal{P} \,\|\, a\text{''}}[\kappa]$. In this case, it is important that the AA only transmit $Cert_{\mathcal{P},a}$ privately to $\mathcal{P}$ on proper identification by means of $C_{\mathcal{P}}$. Otherwise, κ might be obtained by a principal other than $\mathcal{P}$. Consequently, a different trust model applies in this variant: Entities assume that AA has been honest in verifying the identities of principals.

3.2 Freestanding ABE

As suggested by our discussion above, our freestanding ABE approach treats attributes as independent of identities. The aim is to permit access-control policies to be formulated with respect to attributes alone, without a necessity for awareness of the identities of participating principals. In this case, the AA assigns attribute a to a principal simply by transmitting the "implicit" certificate:

$$Cert_a = d_{\text{``}a''}. \tag{3}$$

Thus *all* players with attribute a share the same private key.

As expected in this case, to permit access to κ only to principals possession attribute a, the resource owner now just produces the ciphertext

$$c = E_{\text{``}a\text{''}}[\kappa]. \tag{4}$$

The main benefit of this approach is immediately evident. By making the ciphertext c publicly available, the resource owner permits access to κ by any holder of attribute a with no interaction and no awareness of identities. This is especially useful if the identities of target principals are not known *a priori*, or of the number of revelant principals in the system is large. A side benefit is enhanced privacy protection: The resource owner does not directly learn which entities have gained access to κ. (The resource owner may of course gain partial knowledge in this respect if capable of observing which entities seek access to c.)

There is, however, a flip side to the separation of identities and attributes characterizing this approach and the associated privacy enhancement. In particular, the fact that attribute holders are not identified on accessing a resource means that it is possible to share credentials with impunity. In addition, revocation of attributes is difficult. In the case of identity-bound ABE or standard access-control systems, attributes may be revoked by means of certificate revocation lists (CRLs). This is not possible for a freestanding ABE system. On the other hand, many system designers deprecate the use of CRLs, regarding them as cumbersome to engineer. An oft-favored alternative approach is to rely on tightly controled expiration dates on certificates. This is possible in an identity-bound ABE. The AA need simply issue an attribute a with an associated expiration date, i.e., employ public keys for attributes of the form "$a \parallel date$" in the obvious manner. This technique, described in [6] imparts temporal granularity to the access-control system.

Remark: A cryptographic means of restricting the sharing of attributes among principals is to employ the *traitor-tracing* scheme of Boneh and Franklin [5]. In brief, the idea is to construct multiple private decryption keys corresponding to a single public IBE key. If a private decryption key is found to have been shared inappropriately, it may be traced to its owner by merit of its uniqueness. Boneh and Franklin show how to accomplish efficient traitor-tracing in a standard discrete-log setting. Their idea may, however, be extended more-or-less straightforwardly to the IBE setting. (We omit details due to lack of space.) A drawback of this approach is that the scheme carries linear overhead k, where k is the minimum number of principals capable of colluding to defeat the traitor-tracing scheme.

3.3 More complex policies

To describe the concepts behind identity-bound and freestanding ABE above, we considered a simple policy involving a single attribute a. An important benefit of ABE, however, is that more complex policies are possible that rely on combinations of multiple attributes. Let X_a denote a boolean variable regarded as *true* if a requesting principal possesses attribute a, and *false* otherwise. It is possible, then, to describe a wide range of natural access policies, then, in terms of a boolean formula f on a set of variables $X = \{X_{a_i}\}$ for attributes $\{a_i\}$. Note that it only makes sense to consider *monotone* boolean formulae, i.e., those free from negations. This is because, as explained above, a principal can only assert the absence of an attribute by asserting the possession of an attribute describing this absence. Viewed another way, a principal can always claim spuriously not to possess an attribute, if it so desires.

Let $f[\kappa]$ denote, in loose shorthand, a ciphertext achieving access control on key κ under the policy represented by boolean formulat f. It is a well known fact of boolean logic that any monotone boolean formula f may be expressed in what is known as *disjunctive normal form*, i.e., as the disjunction of sets of conjunctive formulae []. In other words, f may be expressed in the form $f_1 \vee f_2 \vee \ldots f_n$ for some finite n, where f_i assumes the form $X_1 \wedge X_2 \wedge \ldots X_{n_i}$ for some integer n_i and variables $\{X_i\} \in X$. In consequence of this observation, we can show how to implement any monotone boolean policy f as follows: For policies f_1 and f_2, we show how to implement $f_1 \vee f_2$ and $f_1 \wedge f_2$. By induction, then, it is possible to implement f.

Conjunctive construction (AND): We may implement this construction using essentially the same technique proposed above for combining identities and attributes in identity-bound ABE. Let κ_1 and κ_2 be shares in a perfect $(2,2)$-sharing of κ. To implement policy $f = f_1 \wedge f_2$, we may construct the ciphertext $f[\kappa]$ as $(f_1[\kappa_1], f_2[\kappa_2])$. It may be seen that to recover κ_1 and κ_2, and thus the key κ, a principal must have attributes satisfying both κ_1 and κ_2.

Disjunctive construction (OR): To implement policy $f = f_1 \vee f_2$, we may construct the ciphertext $f[\kappa]$ as $(f_1[\kappa], f_2[\kappa])$. A principal with attributes satisfying either f_1 or f_2 can clearly recover κ.

Remark: Inductive use of the AND and OR constructions described here does not in general yield a maximally compact ciphertext $f[\kappa]$. One technique for rendering the ciphertext more compact is to use more general forms of secret sharing. In particular, suppose we wish to implement a policy in which κ is accessible to a principal satistfying j-out-of-n of the formulae in $\{f_1, f_2, \ldots, f_n\}$. Then we may perform a perfect j-out-of-n secret sharing of κ via, e.g., the well known technique of Shamir [10], yielding shares $\kappa_1, \kappa_2, \ldots, \kappa_n$. We then construct the ciphertext $f[\kappa] = \bigcup_{i=1}^{n} f_i[\kappa_i]$. An interesting problem is to devise other methods of rendering ABE ciphertexts more compact.

3.4 ABE synopsis

For convenience, we summarize here the steps taken by the Attribute Authority AA, Alice, a resource owner, and Bob, a candidate recipient in an freestanding ABE system. (The steps are similar for identity-bound ABE.)

ABE overview

1. **Setup**: The AA acts a PKG for an IBE scheme, running `Setup` to generate the private key $master - key$ and the public values $params$.
2. **Attribute List**: (Optional) Alice obtains list of standard attribute types $\{a_i\}$. (Note: no certificates.)
3. **Policy**: Alice determines her access policy f for a resource key κ in terms of the attribute set $\{a_i\}$.
4. **Encrypt**: Alice computes ciphertext $c = f[\kappa]$ using the AND and OR constructions described above.
5. **Post**: Alice broadcasts or posts the ciphertext c in a public location.
6. **Decryption**: If he possesses private IBE keys (“implicit” certificates) corresponding to attributes satisfying f, then Bob can decrypt c, yielding κ.
7. **Authorization Request**: Otherwise, Bob may request from the AA the private IBE keys (“implicit” certificates) for attributes to which Bob is entitled.

4 Example Application: E-mail

In cases where interactivity is limited, as in store-and-forward environments, attribute certificates cannot always function as an effective access-control mechanism. Current versions of S/MIME [3], for example, support the encapsulation of attribute certificates to enable a receiver to make decisions regarding requests by the sender of a piece of e-mail. It is not possible in S/MIME, however, for a sender to use attribute certificates in order to control access to the contents of a message by the receiver. To exemplify the workings of an ABE system, we provide a few examples of how it might be used to impose access-control restrictions on data transmission in a store-and-forward environment such as S/MIME. These examples may be viewed as natural extensions to the Identicrypt system [4], an existing demonstration of the practical layering of IBE on S/MIME. Just as with IBE, the benefits of ABE include a simplified PKI and less interaction on the part of the sender. We restrict our examples here to identity-bound ABE, but the reader may easily envisage variants involving freestanding ABE.

Example 1. It is October 2002, and Alice wishes to send a spreadsheet X to Bob via e-mail. Bob has been working as a contractor in the finance department, and Alice doesn’t know whether he is still working there this month and therefore authorized to read the spreadsheet. Alice’s company defines “$finance : month$” as an attribute describing employment in the finance department for a given month.

In a system involving attribute certificates, Bob would have to contact Alice in order to prove possession of an identity certificate and also present an associated attribute certificate including the attribute “$finance : Oct02$”. In contrast, in a system employing ABE, Bob would not have to contact Alice. In this situation, Alice selects a random symmetric key κ and computes $D = \epsilon_\kappa[X]$, where

ϵ denotes encryption under an appropriate symmetric-key cipher. She sends the pair $(C = (E_{``Bob \,\|\, finance:Oct02"}[\kappa], D)$ to Bob. If Bob has been assigned the monthly attribute "$finance : Oct02$", then he can decrypt the spreadsheet. ∎

Example 2. Alice wishes to send spreadsheet X to Charlie. Charlie should only be permitted to open the spreadsheet if he is in the finance department or if he is a senior manager and employed in the U.S. (We set aside the issue of timestamping in this example.) In this situation, Alice selects a random symmetric key κ and computes $D = \epsilon_\kappa[X]$; she additionally computes a random (2,2)-sharing of κ comprising the pair of keys (κ_1, κ_2). She computes

$$C = \{E_{``Charlie \,\|\, finance"}[\kappa], E_{``Charlie \,\|\, seniormanager"}[\kappa_1], E_{``Charlie \,\|\, U.S.employee"}[\kappa_2]\}.$$

Alice sends the pair (C, D) to Charlie. ∎

Example 3. It is October 2002, and Alice wishes to invite all full-time employees (as of October) to a company outing. Without ABE Alice would proceed by constructing a list of employees with the attribute "full-time". However since Alice's company supports free-standing ABE, and has issued free standing attribute certificates of the form "$full - time : month$" Alice can simply encrypt the message once with a random symmetric key κ and compute

$$C = \{E_{``full-time:Oct02"}[\kappa].$$

Alice can post the ciphertext invitation on the company website, or send it to all employees. ∎

5 Efficiency and Implementation

As illustrated in Example 1, ABE can support a certain level of temporal granularity. In many settings, however, attribute certificates are used on an ephemeral basis, sometimes with expiration times measured in minutes. In most such cases, the key distribution requirements of ABE would require a level of interaction that would negate the basic benefits of the scheme. Thus ABE is most usefully applicable to relatively long-term credentials.

Computation of C requires work linear in the number of argument instances (including repetitions of the same argument) that appear in the logical representation of f. Obviously, general (k, n)-Shamir secret sharing enables a more compact construction of C in cases where a privilege is defined according to possession of some subset of k out of n attributes. Additionally, an ABE system requires storage of a separate key for every credential associated with an individual, while, with use of attribute certificates, in contrast, it is possible in principle to aggregate multiple attributes in a single certificate. Nonetheless, for settings involving desktop machines, we feel that the overhead imposed by the need for separate attribute keys is relatively small, and justified by the resulting infrastructural simplification.

Acknowledgments

The authors wish to thank John Linn for his comments and suggestions.

References

1. ITU-T Recommendation X.509: Information technology - Open systems interconnection - The Directory: Authentication framework, 1998. ISO/IEC 9594-8:1998.
2. ANSI X9.45: Enhanced management controls using digital signatures and attribute certificates, 1999. American Bankers Association Accredited Standards Committee X9.
3. IETF RFC 2632: S/MIME version 3 certificate handling, 1999.
4. Identicrypt email system., 2003. Software and references available at www.identicrypt.com.
5. D. Boneh and M. Franklin. An efficient public key traitor tracing scheme. In *CRYPTO '99*, pages 338–353. Springer-Verlag, 1999. LNCS no. 1666.
6. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO '01*, pages 213–229. Springer-Verlag, 2001. LNCS no. 2139. Also to appear in *Siam J. Computing*. Full version available at http://crypto.stanford.edu/ dabo/pubs.html.
7. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO '82*, pages 199–203. Springer-Verlag, 1982.
8. C. Cocks. An identity based encryption scheme based on quadratic residues. In B. Honary, editor, *Cryptography and Coding, 8th IMA International Conference*, pages 360–363. Springer-Verlag, 2001. LNCS no. 226.
9. J. Linn and M. Nystrom. Attribute certification: An enabling technology for delegation and role-based controls in distributed environments. In *Proceedings of the Fourth ACM Workshop on Role-Based Access Control*, pages 121–130, 1999.
10. A. Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.
11. M. Weiner. Performance comparison of public-key cryptosystems. *CryptoBytes*, 4(1):1–5, Summer 1998.

A Elliptic Curves, Weil Pairing, and IBE

In this section we sketch the technical details underlying the IBE system of [6], based on the Weil Pairing. The following is a somewhat simplified description, but does explain how the existence of a bilinear map on Elliptic curves has been exploited to create the practical IBE scheme.

An elliptic curve, E, may be defined to be the set of solutions to a cubic $y^2 = x^3 + a$ over a finite field F_p, with one additional point, denoted O. By declaring three collinear points to sum to O, E becomes an abelian group with O as identity. Let q be a prime of bitlength k, a security parameter, which divides the order of E. Then the subgroup of order q, G, is a group for which the discrete logarithm and Diffe-Hellman problem are believed to be computationally difficult problems. G may be used analagously to El-Gamal (which uses a field F_p) to define encryption and signature schemes.

It is remarkable that for appropriate elliptic curves E, there exists a bilinear "pairing" on G, called the Weil (or Tate) pairing. Specifically there is a small integer d (approximately 6), and a mapping

$$\phi : G \times G \rightarrow (F_p^d)^*$$

which is bilinear, meaning $\phi(aP, bQ) = \phi(P, Q)^{ab}$, efficiently computable and non-degenerate, meaning for no $P_0 \neq$ is $\phi(P_0, Q)$ identically 0. This mapping shows the decisional Diffe-Hellman problem to be easy for the group G. However, the computational problem appears to be difficult for both G and $(F_p^d)^*$. The reader may refer to[6] for the precise security definition (called the Diffe-Hellman Gap assumption), and an algorithm to compute the pairing ϕ.

The computational mechanics of the basic IBE scheme may now be described in some further detail.

- **Setup** An appropriate Elliptic curve with group G is generated with bilinear map $\phi : G \times G \rightarrow (F_p^d)^*$. A base point $P \epsilon G$ is chosen. Let $s \epsilon Z_q^0$ be the master (private) key, and let $UK = sP$ be the (universal) public key point. A hash function H is also fixed.
- **Key-Gen** Let Q_{ID} be a point on G deterministically derived from the string ID. This is the public key for ID. The corresponding private key d_{ID} is computed as sQ_{ID}.
- **Encrypt** Let M be a message string, and r be a random integer (mod q). The ciphertext C is computed for ID as rP, $M \oplus H(\phi(Q_{ID}, UK)^r)$.
- **Encrypt** Let $C = U, V$ be the cipher text. The message is recovered as $V \oplus H(\phi(d_{ID}, U))$.

It is an easy excercise to verify consistency from the bilinearity of ϕ. A technique of Fujisaki and Okamoto may be used to modify this basic scheme so that it is secure against chosen ciphertext attack.