Binding Identities and Attributes Using Digitally Signed Certificates

Joon S. Park and Ravi Sandhu
Laboratory for Information Security Technology (LIST)
George Mason University
4400 University Dr., Fairfax, VA 22030
jpark@itd.nrl.navy.mil, sandhu@gmu.edu

Abstract

A certificate is digitally signed by a certificate authority (CA) to confirm that the information in the certificate is valid and belongs to the subject. Certificate users can verify the integrity and validity of a certificate by checking the issuing CA's digital signature in the certificate and, if necessary, chasing certificate chain and revocation lists. Usually, we use certificates to provide the integrity of identity or attribute information of the subject. Attributes must be coupled with the corresponding identities. In this paper, we introduce comprehensive approaches to bind identity and attribute certificates, identifying three different techniques: monolithic, autonomic, and chained signatures. We describe each technique and analyze the relative advantages and disadvantages of each.

1 Introduction

Digital certificates support integrity services by confirming that the information in a certificate has not been altered by unauthorized methods and belongs to the proper subject. Public key cryptography has been used for digital signatures on the certificates, providing scalability and non-repudiation services. Certificates are issued, signed, and maintained by the certificate authorities (CAs). Currently, there are two kinds of information supported by certificates: *identity* and *attributes*.

Identity certificates are used in conjunction with authentication services to verify the subjects of the certificates. For instance, the standard certificate format, X.509, contains the subject's public-key information, which is used to authenticate the subject (owner of the corresponding private key).

An attribute is a particular property associated with an entity, such as a role [9], access identity, group, or clearance. An attribute certificate contains the subject's attribute information; however, no authentication information, such as public key, is incorporated within an attribute certificate. Therefore, there should be a mechanism to link attributes to proper identities. We name this mechanism a *binder*. If we were to use a certificate (which contains both identity and attribute information) signed by a single CA, it would be very easy and simple to link and verify the two kinds of information for a subject. However, if we need different CAs and lifetimes for individual information, separate certificates are required.

In this paper, we analyze the high-level structure of identity and attribute certificates, and introduce comprehensive approaches to bind them, identifying three different binding techniques : *monolithic*, *autonomic*, and *chained signatures*. We describe each technique and analyze the relative advantages and disadvantages of each.

The rest of this paper is organized as follows. Next, in Section 2, we describe the technologies most relevant to our approach. In Section 3, we analyze the structure of identity and attribute certificates. In Section 4, we identify three binders and describe each one, providing relative advantages and disadvantages. Section 5 provides a summary comparison of the three binders. This is followed by our conclusions in Section 6.

2 Related Technologies

2.1 Public-Key Certificate

A public-key certificate is digitally signed by a certificate authority (a person or entity) to confirm that the identity or other information in the certificate belongs to the holder (subject) of the corresponding private key. If a message-sender wishes to use public-key technology for encrypting a message for a recipient, the sender needs a copy of the public key of the recipient. On the other hand, when a party wishes to verify a digital signature generated by another party, the verifying party needs a copy of the signing party's public key. Both the encrypting message-sender and the digital signature-verifier use the public keys of other parties. Confidentiality, which keeps the value of a public key secret, is not important to the service. However, integrity is critical, as it assures public-key users that the public key used is the correct one for the other party. For instance, if an attacker is able to substitute his or her public key for the real one, he can forge digital signatures and read encrypted messages.

ITU (International Telecommunication Union) and ISO (International Organization for Standardization) published the X.509 standard [6] in 1988, which has been adopted by IETF (International Engineering Task Force). X.509 is the most widely used data format for public-key certificates today and is based on the use of certificate authorities (CAs). An X.509 certificate is used to bind a public-key to a particular individual or entity, and it is digitally signed by the issuer of the certificate (certificate authority) that has confirmed the binding of the public key to the holder (subject) of the certificate.

2.2 Attribute Certificate

The U.S. financial industry through the ANSI X9 committee developed attribute certificates [2, 5], which have now been incorporated into both the ANSI X9.57 standard and X.509. An attribute certificate links attribute information to the certificate's subject. Anyone can define and register attribute types and use them for his or her purposes. The certificate is digitally signed and issued by an attribute authority. Furthermore, an attribute certificate is managed in the same way as an X.509 certificate. However, an attribute certificate does not contain a public key. Therefore, an attribute certificate needs to be used in conjunction with an ID certificate, such as X.509.

2.3 SPKI (Simple Public Key Infrastructure)

The SPKI [4] Working Group in IETF developed a standard form for digital certificates, focusing on authorization rather than authentication. A SPKI certificate grants specific authorization to a public key, without necessarily requiring identity of the holder of the corresponding private key. The public key can be used as a unique identifier for the key holder. Furthermore, a collision-free hash of the public key can be also used as a unique identifier for the key holder. SPKI provides simplicity using a less rich data encoding scheme than the ASN.1 notation used in X.509.

2.4 Pretty Good Privacy (PGP)

PGP (Pretty Good Privacy [11]), a popular software package originally developed by Phil Zimmermann, is widely used by the Internet community to provide cryptographic routines for e-mail, file transfer, and file storage applications. PGP is based on public-key cryptography, and defines its own public-key pair management system and public-key certificates. The PGP key management system is based on the relationship between key owners, rather than on a single infrastructure such as X.509. A proposed Internet standard has been developed [3], specifying use of PGP. It uses existing cryptographic algorithms and protocols and runs on multiple platforms. It provides data encryption and digital signature functions for basic message protection services.

2.5 Smart Certificates

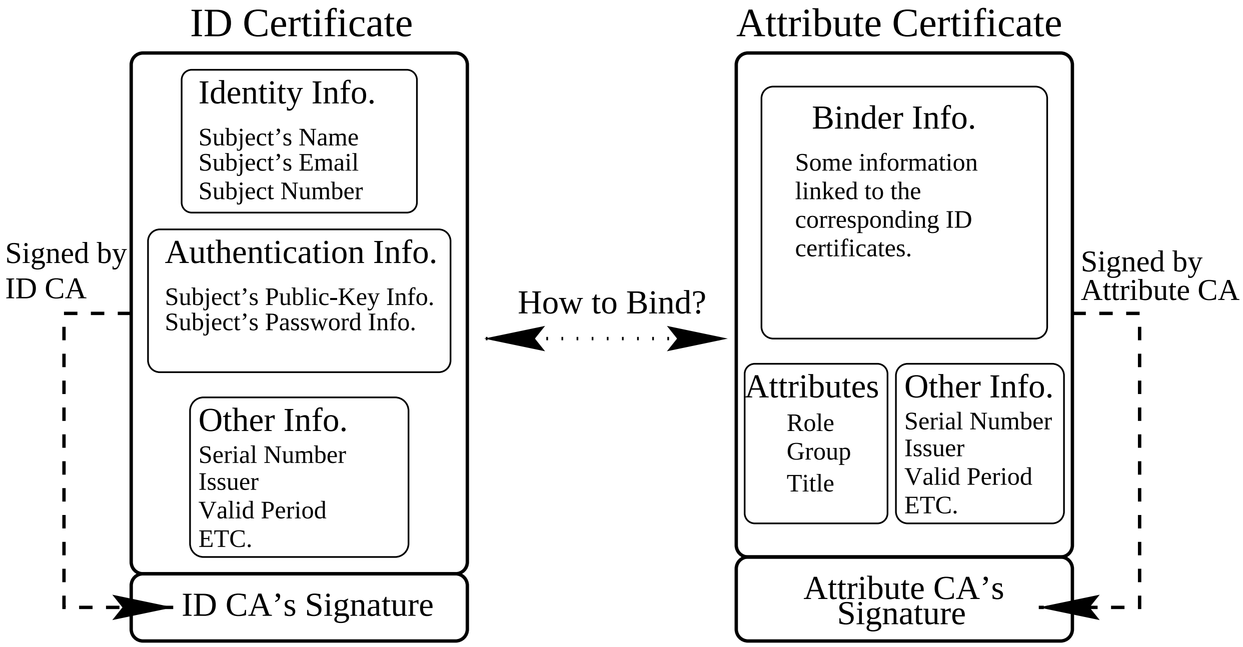
Park and Sandhu have developed *smart certificates* [7, 8, 1] by extending an existing digital certificate, X.509, with several new features. The smart certificates provide short-lived lifetimes, attributes, multiple CAs, postdated and renewable services, and confidentiality services in PKI. According to the requirements of applications, some of these new features can be selectively used in conjunction with currently existing technologies.

If we use a smart certificate, both the attributes and public-key information can be bundled in a single certificate without losing effective maintenance. This provides simplicity for both the protocol itself and for certificate administration. When we need separate authorities for attributes and authentication services, each authority signs separately the same basic certificate and corresponding extension field, which contains attribute information. This can happen multiple times on a basic certificate by different attribute authorities. Each attribute authority has independent control over the attributes he issued. Even though a smart certificate can support independent management for the public-key information and attributes, if there is one authority who controls both sets of information, the system management becomes simpler.

2.6 Secure Socket Layer (SSL)

SSL (Secure Socket Layer [10]) was introduced with the Netscape Navigator browser in 1994, and rapidly became the predominant security protocol on the Web. Since the protocol operates at the transport layer, any program that uses TCP (Transmission Control Protocol) is ready to use SSL connections. The SSL protocol provides a secure means for establishing an encrypted communication between Web servers and browsers. SSL also supports the authentication service between Web servers and browsers.

SSL uses X.509 certificates. Server certificates provide a way for users to authenticate the identity of a Web server. The Web browser uses the server's public key to negotiate a secure TCP connection with the Web server. Optionally, the Web server can authenticate users by verifying the contents of their client certificates.



Note: The content of each block depends on the policy or application.

Figure 1. The Structure of ID and Attribute Certificates

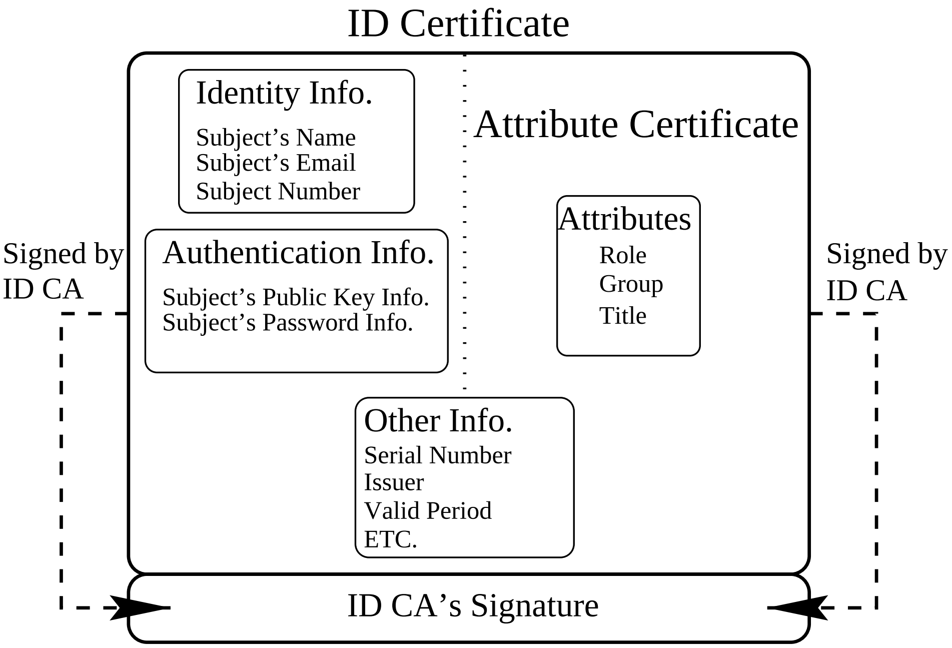
3 Basic Structure of Certificates

A certificate is digitally signed by a certificate authority (CA) to confirm that the information in the certificate is valid and belongs to the proper subject. Certificate users can verify the integrity and validity of a certificate by checking the issuing CA's digital signature in the certificate on demand. Integrity is the most critical service for certificates, since the information in a certificate should not be altered by unauthorized means. Public-key cryptography has been widely used to support digital signatures, providing scalability and non-repudiation services.

There are many different kinds of certificates as we described in Section 2. Their basic contents can be categorized in several blocks. Figure 1 shows an example of the basic contents of ID (identity) and attribute certificates. We classify the nature of information contained in the certificates, and denote them in blocks. The content of each block depends on the policy or application. The identity and authentication information is indispensable to an ID Certificate. Identity information represents the subject of the certificate with the subject's name, email address, subject number, or hashed public key (as introduced in SPKI). To authenticate the subject of an ID certificate, there must be at least one authentication service provided. For this purpose, an X.509 certificate[1] - a standard data format for public-key certificates - contains the subject's public-key information. Alternatively, we can use subject's passwords - which are encrypted or hashed. Even though using passwords is susceptible to dictionary attacks, it provides a simpler and lighter protocol than does public-key-based authentication. Additionally, an ID certificate obtains the issuer's information, certificate serial number, lifetime of the certificate, and so on. All the above information is signed by the ID CA (Identity Certificate Authority) using its private key to support integrity of the certificate, and is verified by using the ID CA's public key.

An attribute certificate contains the subject's attribute information, such as role, group, and title. However, it cannot be used by itself, since the subject of the attribute certificate should be authenticated before the attribute information is used. An attribute certificate should not rely upon unverified or invalid ID certificate. Therefore, an attribute certificate should declare how the attribute certificate is coupled with one or more valid ID certificates. Typically, an attribute certificate is linked to an X.509 certificate by the subject's name and serial number of the X.509 certificate, which contains the subject's public key. There are other alternatives to combine identity and attributes. For instance,

[1] The main purpose of X.509 is binding a public key to the holder of the corresponding private key, even though it was originally designed to contain a password instead of a public key for authentication mechanism.



Note: The content of each block depends on the policy or application.

Figure 2. Binding by Monolithic Signature

it is also possible to link an attribute certificate to the subject's public key, hashed public key, encrypted or hashed password, or other subject-related information contained in the corresponding ID certificate. Furthermore, the ID and attribute certificates can be bundled in a single credential, or separated into different credentials. A bundled certificate proffers interoperability with existing systems, which already support ID certificates, and simplifies the transactions, since it can be used for both authentication and authorization.

In this paper, we identify three major techniques of binding identities and attributes: *monolithic*, *autonomic*, and *chained signatures*. Particularly, in the case of the autonomic signature, we have many alternatives by choosing a set of information in an ID certificate as a binder to the corresponding attribute certificates. Now we describe each case in the following section.

4 Binders Between Identities and Attributes

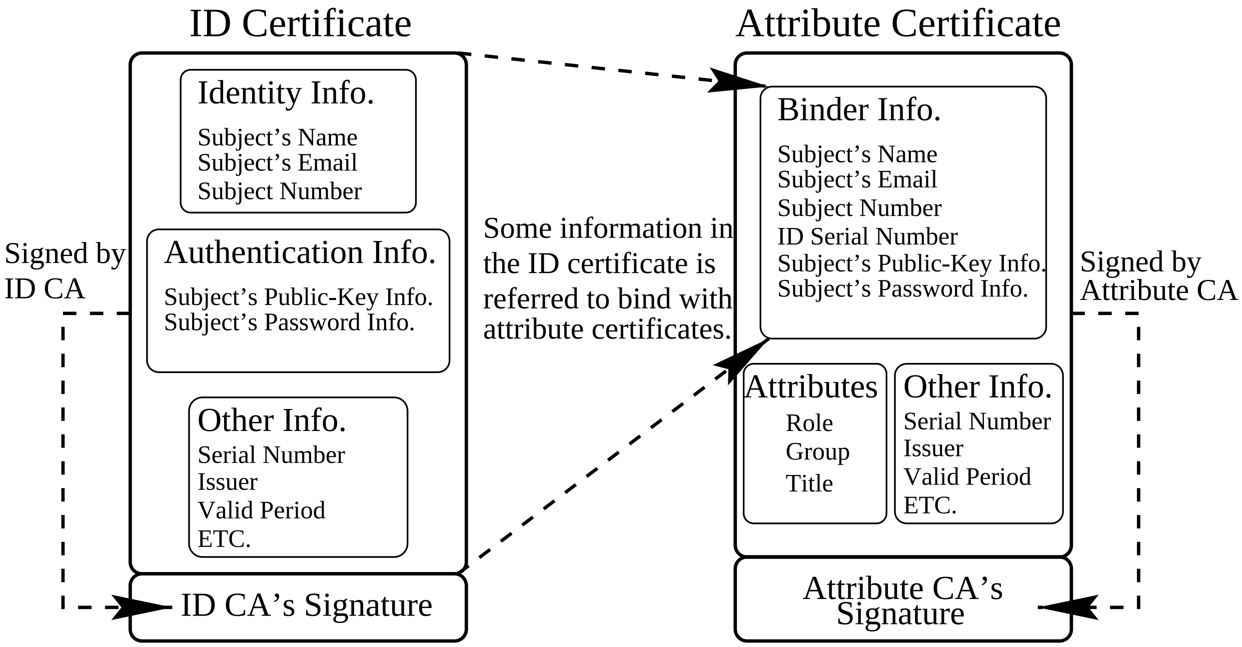
In this section, we discuss different approaches for binding identities and attributes, especially, with respect to *monolithic*, *autonomic*, and *chained signatures*. Each approach including hybrid solution can be made to work, and we provide an analysis of their relative advantages and disadvantages.

4.1 Monolithic Signatures

If there is only one authority who has control over both identities and attributes, the authority is able to sign both sets of information in a single certificate, as depicted in Figure 2. Since both identity and attributes are in a single certificate, the Other Info. block - which contains serial number, issuer, valid period, and so on - can be shared. This can be easily implemented by using X.509 and its extension fields.

The identity information and attributes are *tightly coupled* by a single signature. In other words, once a certificate is issued, the information in the certificate cannot be changed unless a new certificate is reissued. Under a simple policy, this binder is useful, since the system management is simplified. There is only one CA to trust. All the information in the certificate is verified by checking the CA's signature in the certificate and, if necessary, chasing certificate chains and revocation lists.

However, monolithic signatures do not support individual maintenance by multiple CAs. For instance, if we need independent control over each kind of information (e.g., identity, school attribute, and company attribute, etc.) by corresponding CAs, providing different lifetime of each information, the monolithic signature cannot support the requirements. This method increases the convenience of using certificates but decreases the flexibility of the mainte-



Note: The content of each block depends on the policy or application.

Figure 3. Binding by Autonomic Signature

nance.

4.2 Autonomic Signatures

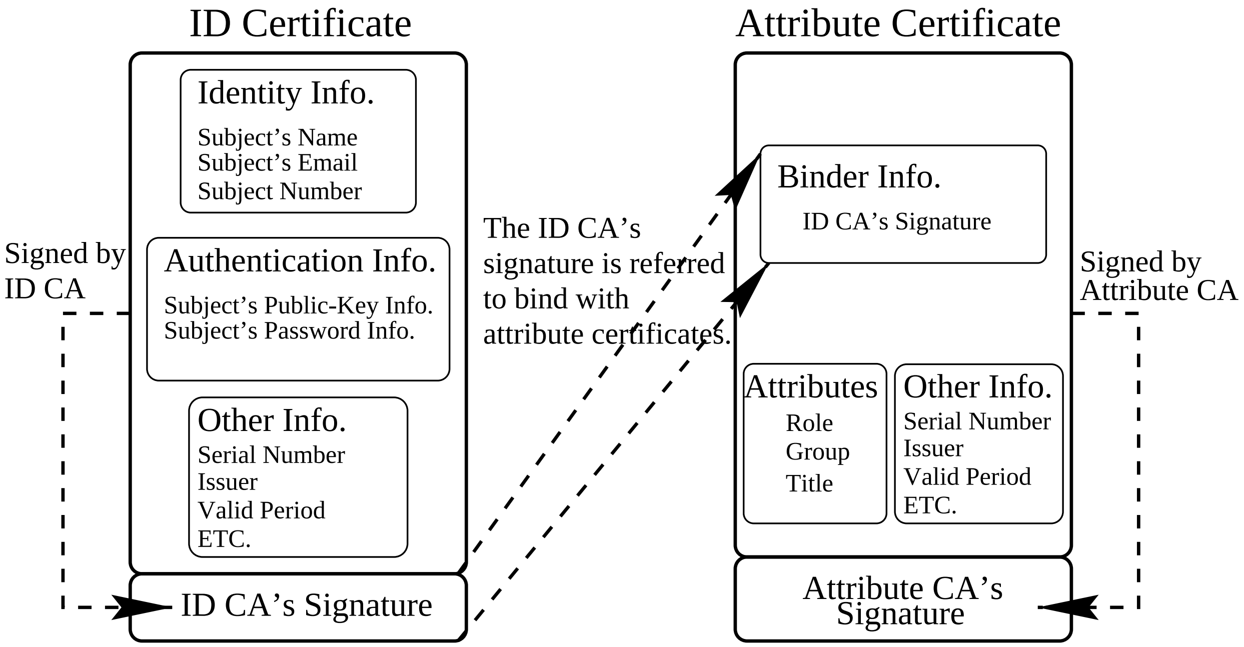
This binder supports multiple CAs and different lifetimes of identity and attribute certificates. Initially, a subject, let's say, Alice, has one or more ID certificates issued and digitally signed by different ID CAs as usual. Those certificates do not have the subject's attribute information. Subsequently, an attribute CA issues an attribute certificate for Alice, binding the attribute certificate with some information in Alice's ID certificates. Conventionally, only subject's distinguished name, or ID certificate's serial number are used as binders. However, it is also possible to use other binders, such as subject's public key, hashed public key, encrypted or hashed passwords, based on applications and domain policies. We allow applications to determine the binders within their particular domains. Many attribute certificates can be issued by many different attribute CAs in this way. Technically and physically, the attribute certificates can be bundled with an ID certificate in a single credential or separated into different certificates.

In Figure 3, the ID CA issues an ID certificate for Alice and signs it. Following the ID certificate generation, an attribute CA (e.g., school attribute CA) issues an attribute certificate, which contains Alice's school attribute (e.g., student), binder information (binder name and value), and other related information, such as lifetime of the attribute and issuer's information. The binder information includes one or some of the items in the binder block in Figure 3. The school attribute CA signs the attribute certificate, including the corresponding binder information to the ID certificate (excluding the ID CA's signature in the ID certificate[2]). For instance, if the subject's public key is used as a binder, the school attribute CA signs the attribute certificate, including Alice's public-key information in her ID certificate, and attaches its signature in the attribute certificate.

Subsequently, another attribute CA (e.g., company attribute CA) issues yet another attribute certificate, which contains Alice's company attribute (e.g., manager), binder information, and other related information. Including the binding information, the company attribute CA signs the attribute certificate. For instance, if the serial number of one of Alice's ID certificates is used as a binder, the company attribute CA signs the attribute certificate, including the serial number (excluding the ID CA's signature in the ID certificate, and attaches its signature in the attribute certificate.

Additionally, more attribute certificates can be issued by different attribute CAs, using different binders. Based on this mechanism, individual attribute certificates are allowed

[2] If the signature is included, we get the chained signatures discussed in the next subsection



Note: The content of each block depends on the policy or application.

Figure 4. Binding by Chained Signature

to refer to different information even in different ID certificates of the same subject. For instance, if Alice has multiple ID certificates, which have the same subject number, issued by different ID CAs, and an attribute certificate uses her subject number as the binder, Alice can then use any one of those ID certificates to verify that the attribute certificate belongs to her. However, if an attribute certificate uses an ID certificate's serial number (which should be unique) as the binder, then Alice must use the matching ID certificate.

Since the autonomic signature uses a subset of information in an ID certificate, excluding the ID CA's signature, it supports a *loosely-coupled* binding mechanism between identities and attributes. In other words, except for the binder, the rest of the information in an ID certificate can be changed, added, or deleted by authorized means. Furthermore, as long as they maintain the same binder, multiple ID certificates issued by different ID CAs can be used for the subject to obtain the attributes. As we mentioned earlier, if Alice's public-key information is used as the binder for her school attribute certificate, Alice can use any ID certificate to obtain her school attributes as long as the ID certificate has her public-key information, which was bound with her school attribute certificate. The rest of the information in the ID certificate, such as lifetime, serial number, and subject's name, can be changed while still maintaining the links to attribute certificates. Therefore, the autonomic signature supports higher reusability of ID certificates than do tightly-coupled mechanisms (which are supported by monolithic and chained signatures). It offers a more convenient mechanism for allowing subjects to obtain their attributes.

4.3 Chained Signatures

This binder supports multiple CAs and different lifetimes of identity and attributes like the autonomic signature, but it provides a *tightly-coupled* binding mechanism between identities and attributes. Initially, a subject, Alice, has one or more identity certificates issued and digitally signed by different ID CAs, as usually occurs. Subsequently, an attribute CA issues an attribute certificate for Alice, binding the attribute certificate and the ID CA's digital signature in one of Alice's ID certificates. Many attribute certificates can be issued by many different attribute CAs, coupled with the ID CA's digital signature in one of Alice's identity certificates. Likewise in autonomic signatures, the attribute certificates can be bundled with an ID certificate in a single credential or separated into different certificates.

In Figure 4, the ID CA issues an ID certificate for Alice and signs it. Then, an attribute CA (e.g., school attribute CA) issues an attribute certificate, which contains Alice's school attribute (e.g., student), binder information (ID CA's signature in Alice's ID certificate in this case), and other related information (such as lifetime of the attribute and issuer's information). Including the digital signature of the

	Monolithic Signature	Autonomic Signature	Chained Signature
CAs	Single	Multiple	Multiple
Lifetimes	Same	Different	Different
Binding Strength	Tightly-Coupled	Loosely-Coupled	Tightly-Coupled
Discovery	Easy	Medium	Difficult
Reusability	Low	High	Medium

Table 1. A Comparison of Binders Linking Identities and Attributes

corresponding ID certificate, the school attribute CA signs the attribute certificate.

Subsequently, another attribute CA (e.g., company attribute CA) issues another attribute certificate, which contains Alice's company attribute (e.g., manager), binder information, and other related information. Including the digital signature of the corresponding ID certificate as a binder, the company attribute CA signs the attribute certificate.

Additional attribute certificates can be issued by different attribute CAs, using the same mechanism. Since this approach uses the digital signature in a particular ID certificate, instead of the actual ID information, it supports a *tightly-coupled* binding mechanism between identities and attributes. In other words, an attribute certificate must refer to the particular ID certificate, which has the digital signature coupled with the attribute certificate. Furthermore, if the information in the referenced ID certificate is changed, the links between the ID and attribute certificates are broken, since the digital signature in the ID certificate should be changed.

5 Discussions

A summary of the binders linking identities and attributes is shown in Table 1. For some applications and domains, a monolithic binder is appropriate, but for other applications and domains, autonomic or chained binders will be preferred. The attribute certificate can be bundled in an ID certificate or separated.

The monolithic signature is the simplest binding mechanism under the policy, which requires a single CA and the same lifetime for both identity and attributes. Technically, identity and attributes with different lifetimes can be stored in one credential and signed by a single CA. However, in this case, to renew the short-lived information (usually attributes), the whole certificate including identity should be re-issued, while autonomic and chained signatures allow us to renew only attribute certificates independently. Therefore, if we need individual CAs or different lifetimes for identity and attributes, either the autonomic or chained signatures are recommended.

The monolithic and chained signatures provide a tightly-coupled binding mechanism; if any content of an ID certificate is changed, its links to attribute certificates are broken. On the other hand, the autonomic signature offers a loosely-coupled binding mechanism; as long as it maintains the same binder information, the content of an ID certificate can be changed without losing the links to attribute certificates.

When identity and attributes are stored in a single credential, discovery of matching certificates is not necessary. When identity and attributes are stored in separated credentials, discovery of matching certificates is required. Therefore, the monolithic signature supports easy discovery of matching certificates, since identity and attributes are always in a single credential. However, it has low reusability, because changing information either in ID or attribute certificates requires issuing a new certificate. On the contrary, the discovery of matching certificate is relatively difficult in chained signature scheme unless ID and attribute certificates are bundled in a single credential (since each attribute certificate is linked to a particular certificate), but changing attribute certificate does not require issuing the corresponding ID certificate (not vice versa).

The difficulty of matching certificate discovery in autonomic signature scheme is between chained and monolithic signatures, because an attribute certificate can refer to multiple identity certificates - any one of those matching ID certificate can be presented with the attribute certificate, but the user needs to decide which matching certificate he or she will use - and vice versa, while chained binding mechanisms requires to discover a particular identity certificate. The autonomic signatures supports high reusability, because either ID or attribute certificates can be changed without breaking the links between them, as long as they maintain the same binder information.

6 Conclusions

In this paper, we have analyzed the high-level structure of identity and attribute certificates, and identified three techniques to bind them: *monolithic*, *autonomic*, and *chained signatures*. We described the relative advantages and disadvantages of each case. The selection of these techniques, including hybrid solutions, depends on the application and given policies.

References

[1] G.-J. Ahn, R. Sandhu, M. Kang, and J. Park. Injecting RBAC to Secure a Web-based Workflow System. In *Proceedings of 5th ACM Workshop on Role-Based Access Control*. ACM, Berlin, Germany, July 26-28 2000.

[2] American Bankers Association. *Enhanced Management Controls Using Digital Signatures and Attribute Certificates*, 1999. Accredited Standards Committee X9, X9.45-1999 (ANSI X9.45).

[3] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. *OpenPGP Message Format*, November 1998. RFC 2440.

[4] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. *SPKI (Simple Public Key Infrastructure)*, September 1999. RFC 2693.

[5] S. Farrell. *An Internet Attribute Certificate Profile for Authorization*, March 2000. draft-ietf-pkix-ac509prof-02.txt.

[6] ITU-T Recommendation X.509. *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework*, 1993. ISO/IEC 9594-8:1993.

[7] J. S. Park and R. Sandhu. RBAC on the Web by Smart Certificates. In *Proceedings of 4th ACM Workshop on Role-Based Access Control*, pages 1–9. ACM, Fairfax, Virginia, October 28-29 1999.

[8] J. S. Park and R. Sandhu. Smart Certificates: Extending X.509 for Secure Attribute Services on the Web. In *Proceedings of 22nd National Information Systems Security Conference*, Crystal City, Virginia, October 1999.

[9] R. Sandhu. Role-Based Access Control. *Advances in Computers*, 46, 1998.

[10] D. Wagner and B. Schneier. Analysis of the SSL 3.0 Protocol. In *Proceedings of the Second UNIX Workshop on Electronic Commerce*, November 1996.

[11] P. R. Zimmermann. *The Official PGP User's Guide*. MIT Press, 1995.