Helsinki University of Technology Laboratory for Theoretical Computer Science

Research Reports 61

Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 61

Espoo 2000 HUT-TCS-A61

PRIVACY AND ACCOUNTABILITY IN CERTIFICATE SYSTEMS

Tuomas Aura and Carl Ellison

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Helsinki University of Technology Laboratory for Theoretical Computer Science
Research Reports 61
Teknillisen korkeakoulun tietojenkäsittelyteorian laboratorion tutkimusraportti 61
Espoo 2000 HUT-TCS-A61

PRIVACY AND ACCOUNTABILITY IN CERTIFICATE SYSTEMS

Tuomas Aura and Carl Ellison

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory for Theoretical Computer Science

Teknillinen korkeakoulu
Tietotekniikan osasto
Tietojenkäsittelyteorian laboratorio

Distribution:
Helsinki University of Technology
Laboratory for Theoretical Computer Science
P.O.Box 5400
FIN-02015 HUT
Tel. +358-0-451 1
Fax. +358-0-451 3369
E-mail: lab@tcs.hut.fi

ISBN 951-22-5000-4
ISSN 0783-5396

Picaset Oy
Helsinki 2000

ABSTRACT: Discretionary access right management on the Internet and in other distributed communications systems is increasingly based on public-key identity and authorization certificates. The certificates pose a threat to privacy because they identify the owners and reveal the authorization relations between them. This paper overviews the privacy concerns and describes techniques for minimizing the amount of confidential information leaked about individuals and organizations. We also show how identity escrow certificates can ensure individual accountability without identity authentication. All the techniques can be implemented with SPKI certificates.

KEYWORDS: privacy, anonymity, PKI, certificates, SPKI

CONTENTS

1 Introduction 1

2 Certificates and privacy 1

2.1 Identity certificates 2

2.2 Authorization certificates 4

2.3 Trust chains and thresholds 4

3 Anonymity techniques 6

3.1 Key-oriented access control 7

3.2 Certificate reduction 8

3.3 Temporary and task-specific keys 10

3.4 Open threshold certificates 11

4 Ensuring individual accountability 12

4.1 Encrypted identity certificates 12

4.2 Identity escrow 13

5 Conclusion 15

References 16

1 INTRODUCTION

Although public-key certificate standards such as X.509 [8, 25] have existed for more than a decade, the use of certificates for authentication and access control has become wide-spread only in the last years. This development is due to two factors. First, the increased speed of microprocessors has made public-key signatures acceptable for a wide range of devices and applications. Second, the growing complexity and distribution of the communications systems has created a need for powerful access control mechanisms. The public-key certificates can provide scalable and fully distributed solutions for access rights management.

The global access to services and efficient authentication of the users, however, does not come without a cost. It is easier than ever to gather intelligence about the activities of individuals and organizations. In the future, our continuous presence in mobile communications networks through ubiquitous computing devices will further amplify the privacy concerns. The reliance on identity certification is contributing greatly to the problem because the certificates explicitly reveal the identities of the key owners.

The key-oriented access control systems such as SPKI [14, 15, 16, 17], SDSI [23] and PolicyMaker and KeyNote [5, 4] promise some improvement because they can identify an entity by its signature key without knowing its name. Their certificates, on the other hand, convey authorizations that are often much more sensitive information than the mere identities of individuals. Thus, the key-oriented access control both helps to solve privacy issues and creates new ones.

This paper addresses the threats to privacy that arise from certificate-based authentication and access control and ways of alleviating them. We explain in Sec. 2 how public-key certificates are used and what kind of information they may reveal. Sec. 3 details techniques for minimizing the amount of information that is leaked by the certificates. In Sec. 4, we show how identity escrow makes it possible to hold an individual accountable for his actions without compromising the privacy of honest users. Sec. 5 concludes the paper. It turns out that careful system design can greatly reduce the number of parties that are able to identify an individual and the amount of information that is revealed.

2 CERTIFICATES AND PRIVACY

This section introduces public-key certificates and discusses privacy concerns that arise in their use. We consider basic identity and authorization certificates, authorization chains, and threshold certificates. The threats are twofold: information may be collected from the certificates by outsiders, and by parties with legitimate access to the certificates.

Another much-studied threat to privacy is traffic analysis. The problems identified here are independent of traffic analysis although intelligence from both sources can be combined for greater damage. That is, the problems discussed in this paper exists even if there is enough traffic so that an individual user's access patterns can blend into the crowds, possibly with the help of random grouping of requests [22], and the source and destination addresses of all communication are blinded from unwanted eyes with reliable techniques like mixes, onion routing and chained forwarding of connections [11, 24, 18].

2.1 Identity certificates

A certificate is a proof of identity or a letter of authorization signed by an authoritative entity. We call the signing authority the *issuer* and the entity that receives the rights the *subject*. The issuer signs the certificate with a public-key cryptographic algorithm such as RSA or DSS [21].

The most common type of certificate used today is the X.509 identity certificate that has been standardized by CCITT/ITU-T [8, 25]. It is used, for example, in the SSL/TLS user and site certificates on the World Wide Web [13]. The identity certificates are issued by trusted certification authorities (CAs). In the SSL model, there are many mutually competing global CAs.

The main privacy concern with identity certificates is that they reveal the name of the subject to anyone who sees the certificate. If the communication is not encrypted, the certificates make it extremely easy for an outside listener to identify the communicating parties and to automatically track individual users. Unfortunately, the confidentiality of most traffic in the computer networks is not sufficiently protected by encryption. For example, SSL exchanges the certificates in plaintext before changing into ciphered mode.

Even if there happen to be no outsiders listening to the communication, an insider threat remains. The party verifying the certificates might misuse the identity information in them. This may sound like an awkward concern since the sole purpose of an identity certificate is to convey the name of the subject to the verifier. It is, nevertheless, one of the reasons why the SSL client certificates have not become popular on the web. The users prefer to exercise strict control over to whom they disclose their identities. Given the recent controversies about HTTP cookies and unique microprocessor ID numbers, it seems unlikely that the client certificates will ever be widely accepted by Internet users. We will argue below that there is a less compromising technical alternative to identification: authorization.

The concern about disclosing ones identity is amplified by the fact that the names in the certificates are intended to uniquely pinpoint the individual. For instance, X.509 is originally designed to use *distinguished names* whose owner can be found without ambiguity from an X.500 directory.

We therefore claim that it is not at all naive be worried about the following:

- **Privacy problem 1:** Identity certificates disclose identities.

Even after the certificate exchange, the authenticated communication still leaks clues to outsiders about the activity that is taking place and about its participants. Cryptographic keys and their hashes are unique by nature. Thus, they are much more reliable identifiers than names that tend to become ambiguous unless selected with care. An eavesdropper only needs to find out once the keys of its targets. After that, it can recognize any occurrences of the keys and messages signed with them. Correlating the keys with the owners is easy in a fixed network where the sender and receiver addresses effectively identify the communicating parties. And even before figuring out the names, the observer can link the occurrences of a single key to each other. Furthermore, the keys often have long lifetimes in the order of years. This makes tracking them attractive regardless of the extra work. The insider threat exists also here: the legitimate receivers of the certificates might not only verify them but also remember the keys to spy on their other activities.

- **Privacy problem 2:** Signature keys are uniquely recognizable.

Most electronic commerce today either relies on password authentication or has no proper client authentication. The SSL client certificates have not been accepted by consumers. Besides the privacy concerns, the reason is that the centralized model does not scale well to individual users. The certificates are difficult to obtain and their cost is too high for the majority of consumers. The server authentication is mostly based on SSL server certificates. It provides limited assurance because the users often do not know the name of the server. Furthermore, it is unclear to an average web user which certifiers are actually trustworthy. Most users and companies never review the growing list of CAs the browser software trusts. The centralized trust model is also unsuitable for private organizations that want to control their own resources and cooperate with each other. Clearly, a more structured system is needed for providing certificates to individuals, to lower levels of organizational hierarchies, and for access across organizational boundaries.

The X.509 standard originally specifies a hierarchy of certification authorities. All CAs in the hierarchy are selected by upper level authorities and are globally trusted. PGP [26] relies instead on a network of personal trust that the users express with certificates.

A structured system of CAs is, however, dangerous from the privacy point of view because the CA hierarchy and the certificates issued by it will mirror the structure of the issuing organization. Even the PGP web of trust is an image of the personal relations between people. The disclosure of organization structure is a problem for businesses that often deny public access to even their telephone directories. Military organizations have the same kind of reservations.

- **Privacy problem 3:** CA hierarchies and networks mirror organization structures and personal relations.

One place that is especially attractive for someone who wants to gather data about the organization is the CAs themselves. It is well-known that the CAs represent a weak point for the system integrity because a compromised CA could sign false identity certificates. But all the sensitive certificate data is also available to the CAs. They get to see the certificates that they sign and they need access to enough organizational information to verify the data before signing. A CA can (without violating any integrity constraints) leak a detailed map of the organization that uses it. Therefore, the CAs are also a weak point for confidentiality.

- **Privacy problem 4:** CAs have access to confidential organizational information.

2.2 Authorization certificates

Certificates can also be used for granting access rights. By signing an authorization certificate, the issuer authorizes the subject to the services listed in the certificate. In a distributed access control system, the certificate may be held by the subject until it wants to use the rights. At that time, the server verifies the signature on the certificate, compares the authorization to the service request, and makes sure that the issuer itself has the authority to grant the rights.

In SPKI, the subject of an authorization certificate may be specified by a name but, more commonly, the rights are granted directly to a signature key (see Sec. 3.1). The authorization has a validity period that is specified on the certificate. The validity of an authorization is often much shorter than that of an identity certificate.

Like identity certificates, the authorization certificates may disclose confidential information. The authorization field in a certificate contains detailed data about the business the subject wishes to conduct. An outside observer with access to the certificates can track the relations between issuers and subjects and the access rights that are attributed to them. The servers may also gain knowledge about the access rights of their clients beyond what they need to know. This happens when a certificate grants wider rights than are needed for a particular service request.

- **Privacy problem 5:** Authorization certificates reveal business and personal relationships.

2.3 Trust chains and thresholds

The subject of a certificate can, if not forbidden by the certificate or by the verifier's policy, redelegate the access rights by issuing another certificate. The redelegation is a key component of SPKI and other key-oriented public-key infrastructures. However, the idea has previously been applied in systems that rely on identity certificates and shared-secret authentication [19, 7].

In Fig. 1(a), the service provider S issues a certificate that authorizes the customer organization A to use its services. The organization in turn authorizes its employees. User B, wary of his notebook computer being stolen, does not want to store his master key on it but grants the authority to the notebook only temporarily. In each step, an identity certificate binds the subject name to a signature key and, thus, becomes a part of the trust chain. As a result of the three steps of authorization, the notebook C has the right to use the services of S. In the end, the notebook issues requests to the server, which can be construed as granting the rights further to the specific message.

Interestingly, the access rights are finally verified by the same server from which the chain of trust originates. This kind of situation, called *authorization loop* [7, 17] is common when access rights are distributed through a chain of entities. In addition to the signed service request, the verifier needs to see all the certificates in the chain and identity certificates for all the named entities. C attaches all these certificates to the request or they are conveyed to the verifier or its agent by some other means. In order to be

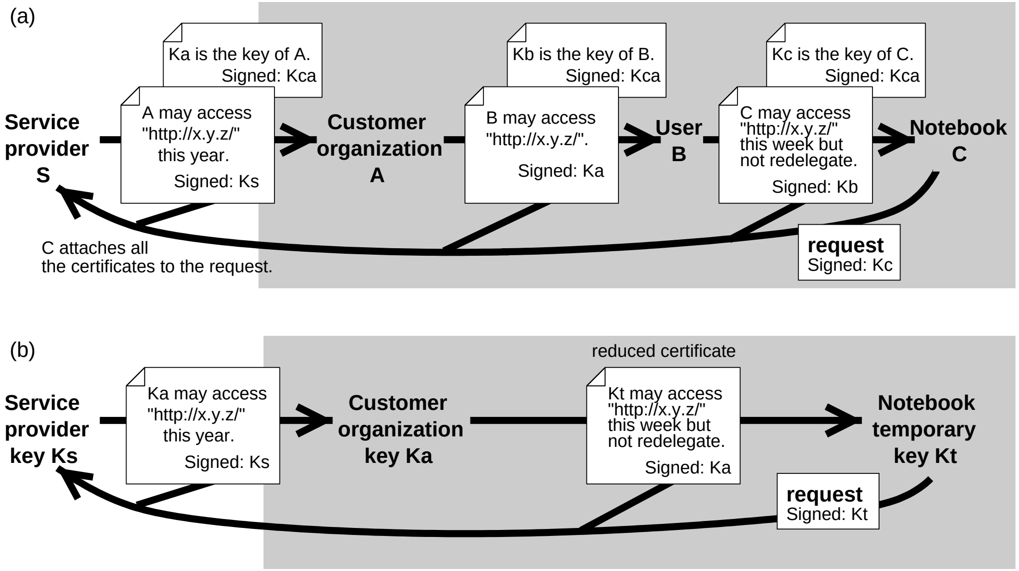
(a)
Ka is the key of A.
Signed: Kca
Kb is the key of B.
Signed: Kca
Kc is the key of C.
Signed: Kca
Service provider S
A may access "http://x.y.z/" this year.
Signed: Ks
Customer organization A
B may access "http://x.y.z/".
Signed: Ka
User B
C may access "http://x.y.z/" this week but not redelegate.
Signed: Kb
Notebook C
C attaches all the certificates to the request.
request
Signed: Kc
(b)
reduced certificate
Service provider key Ks
Ka may access "http://x.y.z/" this year.
Signed: Ks
Customer organization key Ka
Kt may access "http://x.y.z/" this week but not redelegate.
Signed: Ka
Notebook temporary key Kt
request
Signed: Kt

Figure 1: A certificate chain and privacy enhancements

accepted, *C*'s request has to be allowed by every certificate in the chain. The access rights that *C* gets are therefore the intersection of the rights listed in the certificates and their validity period is the intersection of the validity periods of the certificates.

The certificate chains and the possibility of restricting the granted rights in each step may be used in innovative ways. For example, [3] suggests the use of authorization certificates for access control in intelligent networks (IN) where code modules from competitive service providers need to co-exist and cooperate on switching platforms. The same ideas extend to certifying mobile code and its authors. However, such advanced applications of certificates do not come without a cost to privacy. When complex relations between entities are expressed with the certificates, the information becomes available to any part of the infrastructure that handles them. The authorizations fields and validity periods in certificates may reveal the exact nature and duration of contracts between companies. The certificate chains thus become sources of intelligence about the entire value chain from network operators to service providers and to end users.

- **Privacy problem 6:** Like CA hierarchies, chains of authorization certificates mirror organization structures and business processes.

Hence, if one uses a credential issued by one business associate to do business with another one, there is a danger of confidential information being leaked. This seems to lead to the extension of the Chinese-Wall security policy [6] from the sharing of information to the sharing of access credentials. Such limitations are obviously too inflexible for many applications and we must look for other ways of alleviating the privacy problems.

Another advanced feature supported by SPKI is the *threshold certificate* (Fig. 2(a)). It differs from a normal authorization certificate in that it has several subjects. All or a specified number of them must cooperate to use the rights given by the certificate. The certificate contains the list of subjects and

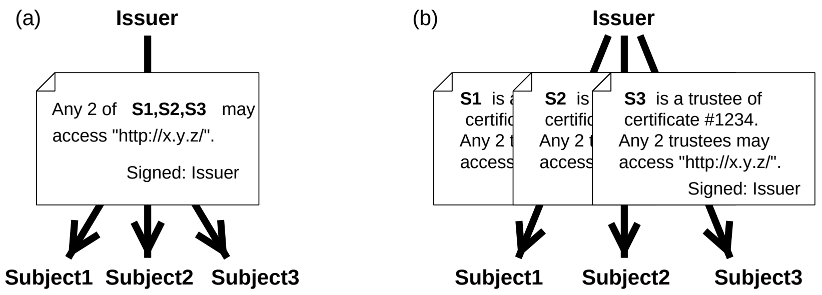

Figure 2: Threshold and open threshold certificate

the threshold number. The threshold certificates add significant flexibility to trust management in some applications. The rights can be issued to a group of trustees. The trustees use the rights by jointly signing an access request or by authorizing a single subject to act on their behalf.

Copies of a threshold certificate are normally sent to each of the subjects and they may be distributed via other channels. Therefore, the threshold certificate discloses the names or public keys (whichever are used to specify the subjects) of the trustees to each other and potentially to outsiders. This makes the subjects vulnerable to pressuring, bribery and extortion. Additionally, those subjects who decline to cooperate in using the access rights cannot remain anonymous. Their keys or names can be seen from the certificate.

- **Privacy problem 7:** Threshold certificates reveal the lists of subjects.

In Sec. 3.4, we will describe an alternative threshold certification scheme that avoids these problems.

3 ANONYMITY TECHNIQUES

Anonymity means that a client is able to access services without revealing its identity. Anonymity is needed by businesses, military, law-enforcement and governmental organizations to shield them against intelligence and to protect their staff from personal exposure to threats. It is increasingly needed by individuals against both targeted electronic snooping and mass collection of data about their lives.

Ideally, anonymity should be the basic assumption and authentication an optional service. In practice, reliable anonymity mechanisms such as onion routing and zero-knowledge proofs are considered too expensive to implement. Moreover, most security systems build on top of the classical model where the entities first identify and authenticate each other and then use the identifiers for deciding what kind of access to allow. The explicit use of unique names or other permanent identifiers makes is difficult to implement additional anonymity services.

An alternative approach is to avoid the use of identity whenever possible. This means that the basic security architecture should not depend on unique entity names or identifiers. Although it is not impossible to track individuals in such a system by analyzing the traffic and behavioral patterns, it is far more difficult than if explicit identifiers are available. Both strong authentication and enhanced anonymity will be optional services that one has to pay for. That way, users can select combinations of services best suited for their goals.

This section explores low-cost techniques for avoiding the explicit use of names and introduces some enhanced anonymity techniques that become possible when disclosing the identity is not mandatory. Our goal is to present solutions for the problems identified in Sec. 2.

A parallel can be drawn between the techniques of this section and popular anonymity services on the Internet. Key-oriented access control (Sec. 3.1) allows an entity to use a public key as its identifier while a privacy-friendly Internet server allows clients to be just IP or email addresses without asking for user names. Certificate reduction (Sec. 3.2) means that the access rights and requests are communicated through a proxy. This is akin to the anonymous access provided by web anonymizers [1] and gateway proxys of large organizations. Temporary keys (Sec. 3.3), on the other hand, resemble the multiple identities provided by many web-based discussion boards and various other services. In SPKI, support for all these mechanisms has been built into the public-key infrastructure.

3.1 Key-oriented access control

The subject of an authorization certificate may be specified either as a public signature key or as a name. If the subject is a name, is must be bound to a signature key at the verification time. This leads to the use of identity certificates and to the associated privacy problems. For example in Fig.1(a), all the subjects need identity certificates. It is therefore beneficial to issue the authorization certificates directly to the public signature keys of the subject entities as in Fig. 1(b). SPKI and a number of other distributed access control systems are called *key-oriented* because their primary subjects are keys and not names.

For access control, the key-oriented approach offers several advantages. The direct authorization of a key is convenient because the key will also sign the access requests. The ownership of the key can be verified directly without a trusted third party. The signature keys can, in fact, be thought of as identifiers because they are unique by nature. When two entities create a trust relationship between themselves, they pass their public keys to each other. Since names are not used, the key-oriented systems do not need a global naming scheme or trusted CAs to support it. Hence, there are no central authorities that could supervise activities in the system and compromise its integrity or confidentiality. When the CAs are left out of the trust chain, the system becomes simpler, less centralized and more reliable.

The key-oriented access-control systems have shifted the focus from identity authentication to authorization. The keys authorize keys directly without reference to the names of the key owners. For this reason, the users can remain nameless without any special measures. When the names are not explicitly advertised, one must systematically collect intelligence data about the system and analyze it in order to identify individual entities and their relations. In Fig.1(b), the nature of the owner of key Kt may remain obscure to the server S while the names of B and C in Fig.1(a) reveal much more.

Based on the above, one can ask if identity certificates are really a good idea at all. There are many situations where the communicating parties do not need to know each other's identities. This is apparent on the World Wide Web where users prefer to remain anonymous or use pseudonyms. For a service provider, it is more important to verify the access rights of its clients than to know who they are. For the client, the critical question is often not the name of the server but an assurance that the server is the same one the client has learned to trust. Thus, the real issue is access control, not identity authentication. It should be permissible for an individual or organization to change its identifier and create multiple identities. The price of creating a new virtual identity should be much lower than the service charges of current commercial and governmental certifiers, preferably as low as generating a new cryptographic key. What is needed is a flexible system for credential management so that a physical entity can take advantage of the access rights of all its virtual identities. Even payment protocols could leave the customer anonymous [9]. The merchants only need to be able to verify that they receive payments from those whom they serve.

The type of nameless access control provided by SPKI and other key-oriented public-key infrastructures becomes particularly useful in mobile systems where users roam in untrusted foreign networks. The risk of being exposed to hostile intelligence is greater for a mobile user than for a fixed one. Luckily, hiding the identity in a mobile network is also easier because the location does not give much information about it. It is often unnecessary for a network operator to know who exactly is using the network as long as the access rights are checked. In particular, the user identity is difficult to discover when the user is accessing services local to the foreign network without connecting to a home network.

Nevertheless, several of the privacy issues listed in Sec. 2 remain in key-oriented systems. The relations between cryptographic keys are documented explicitly in the certificates and the keys may be mapped to real-world entities by careful analysis of access patterns. By observing the certificates, outsiders can learn more about the structure and workings of system than would be desirable. The following sections describe ways of alleviating these problems. Luckily, the absence of explicit names makes it easier to implement additional privacy protection.

3.2 Certificate reduction

The two main techniques for preventing the tracking of signature keys are *certificate reduction* and *temporary keys*. The former will be discussed here and and the latter is the topic of the next section.

SPKI certificate chains can be reduced into single certificates. When A issues an authorization certificate to B (without restricting redelegation) and B grants the rights with another certificate to C, the two certificates imply a direct authorization from A to C. Upon request, A may issue the reduced certificate to C. When the subjects are keys, the reduction is a straight-forward syntactic operation. A simply verifies the certificates in the chain, computes an intersection of the access rights and validity periods, and signs the reduced certificate.

Although reduction is mainly used for improving performance, it also has the effect of hiding the intermediate keys in a chain of certificates. Thus, the reduced certificate carries less information about the trust chain. Certificate reduction should be used as a privacy protection mechanism by organizations that do not want to reveal their internal structure and relations to outsiders. They can do that by using only one or a few keys to communicate access rights with the outside world. All certificate chains inside the organization are reduced into single certificates issued by these keys.

In Fig.1(b), the two certificates internal to the customer organization have been reduced into one. C (the owner of Kt) sent the two certificates to A (the owner of Ka) and asked it to reduce them. As a consequence, the key or name of B does not appear in the certificates that go to the server S.

The protection could be taken one step further by letting a single or a few proxy keys sign all the requests made by the organization. The actual client C would redelegate its rights to a proxy that acquires a reduced certificate from the first entity A that belongs to the organization thus hiding both B and C. A more straight-forward but probably not as scalable solution would be would be to ask A to sign the requests directly.

Another possible enhancement would be to minimize the amount of information the server gets from the authorization field of a certificate. If the reduced certificate allows operations that the client does not request from the server, knowledge of the unused rights is unnecessarily leaked to the server. The leak can be fixed by including in the reduced certificate only the subset of rights that is needed for the particular request. (This can be implemented with standard SPKI reduction, without special support in the reducing host. The client may redelegate the necessary rights to its own key and have the chain, including the certificate to itself, reduced.) Consequently, the server gets only as much information about the client's rights as it absolutely needs to know. In fact, the minimal authorization does not tell the server anything that it could not read from the access request. The cost is reasonable if the certificates are reduced in any case and if the needed subset of rights does not change frequently.

The reduction requires the cooperation of the first key in the reduced part of the chain. That key signs the reduced certificate. The signer gets to see all the certificates in the chain and has to be trusted with that information. For a chain of length n, the reduction process takes n steps of certificate verification and one signing. It is done on-line and the on-line server must be secure enough to hold the signing key. On the other hand, the reduction saves communications bandwidth and work later in verifying the certificates.

Some questions still remain. The key that makes the access request cannot be hidden with reduction. Furthermore, individual consumers may not be members of any organization that provides reliable reduction service but they nevertheless want to protect their privacy. Therefore, reduction does not completely solve the issue of recognizable public keys. Some of these problems can be solved with temporary keys.

3.3 Temporary and task-specific keys

Another way to discourage the tracking of signature keys is to change keys often and to create new keys for each new task. Frequent key changes make it difficult for an observer to correlate the actions of a single user over time. Using separate keys when communicating with different entities or for each unrelated task prevents the easy combination of gathered information from the many roles of a single entity.

Temporary and task-specific keys are an excellent way for individuals to hide their sensitive rights and for consumers to prevent the collection of data on their behavior. The entity that wants privacy can itself decide to use a temporary key. It does not need to trust others to protect its identity. Temporary keys can also be used by organizations. They are most efficient, however, when the users owning the keys are themselves interested in the benefits of the improved privacy.

The cost of temporary keys is mostly paid by the individual who wants the protection. It must generate new keys and acquire the access rights for them. No special support is needed from the servers as long as there are no policies against multiple keys and simultaneous virtual identities. Communication and load on other entities may increase if keys are changed frequently and relationships between them are short. There is a trade-off between cost and privacy: keys can be reused for a few times or for a certain number of purposes. An additional advantage of task-specific keys is that they help to keep the roles of a single entity apart from each other.

It should be noted that although a level of privacy could be achieved by frequently changing pseudonyms in a name-based system, the cost of creating new names is higher than that of generating new keys. New keys can be created and distributed locally while a new name must be certified and distributed through the name service infrastructure. In addition to the cost of their use, the centralized services would be weak points for anonymity protection.

Problems can arise with changing keys and electronic payments since all popular payment methods identify the client. An organization can work around this by providing anonymous financial transactions to its members. It may set up an agent that makes all the payments and delegates the rights to the temporary user keys. That way, a series of transactions cannot be linked to the individual user. A trusted server (e.g. a bank) can provide the same kind of service for individual consumers. Naturally, an anonymous digital payment scheme would solve the problem.

Probably the best privacy protection in key-oriented access control is created by combining temporary keys with certificate reduction. The user can acquire its rights with a permanent key, delegate them to a temporary one, and have the certificates reduced so that its permanent key is not visible in the reduced chain. In Fig. 1(b), the notebook key Kt is a temporary key that is regenerated as often as the user renews the delegation of his rights to the notebook. With the temporary key and the reduction of the certificate chain, the identity of the user B is effectively hidden from both the server S and from any foreign networks B might visit to access S. The contrast to Fig. 1(a) where the names of B and C is readily available to all parties is dramatic.

3.4 Open threshold certificates

The privacy problems with threshold certificates can be solved in a satisfactory way by using so called *open threshold certificates* [2]. The trick is to replace the single certificate with several *subcertificates* as illustrated in Fig. 2(b). The issuer signs a separate subcertificate for each subject. All the subcertificates contain the same certificate identification number and the threshold value k. The issuer must make sure that it uses a new certificate identification number for each new threshold scheme. The number could, for example, be a random bit string of sufficient length. Any k trustees can use the access rights or authorize another entity to act on their behalf. They must attach their subcertificates to the service request. The verifier will accept the collection of k subcertificates if they all have the same certificate identification number and differ only in the subject field.

The open threshold certificates have two advantages over the normal kind. First, they add flexibility to certificate management. The subcertificates may be processed and distributed together or individually. Trustees can be added by issuing new shares without redistributing the certificates to all holders. Second, the shares may be kept secret by the trustees. The names or keys of the trustees that remain silent about their shares and do not cooperate to use the rights may remain anonymous forever. Thus, the subjects are protected from unwanted attention. Of course, nothing prevents the issuer from publishing the subcertificates or the list of subjects if that is wanted.

It should be noted that the increased number of certificates does not require additional storage or communications capacity when they are distributed to the trustees. This is equivalent to sending them copies of a normal threshold certificate. It is also straightforward to extend the reduction of normal threshold certificates (defined in [17]) to the open threshold ones. After the threshold number of trustees have signed a service request or granted the rights to a single entity, the certificates maybe reduced into a direct authorization to the request or to the single entity.

There are several ways of implementing the open threshold certificates. In PolicyMaker where the authorizations are programmable, it would be straightforward to implement the threshold rule as a piece of code embedded in a certificate. In SPKI, we may encode the new kind of threshold scheme into the application-specific authorization field. The verifiers and reducers must be programmed to support it.

There is, however, a way around this inconvenience in SPKI. The trick is to create a new temporary key for each subject and to issue a normal threshold certificate with the threshold k to the temporary keys. Each of the temporary keys is then used to sign an authorization certificate to redelegate the rights to one of the actual subjects. After this, the private temporary keys are destroyed. Every subject gets a copy of the threshold certificate and the redelegation certificate that was issued to it. According to the SPKI certificate semantics, this arrangement is equivalent to an open threshold certificate issued directly to the subjects except that new subjects cannot be added later. Any k subjects may combine their certificates and start using the access rights without knowing who the other subjects are. The generation of the temporary keys, of course, is an extra cost. A final solutions would be to include the open threshold certificates in the certificate standard e.g. as a variant of the subject field syntax.

The open threshold certificates provide approximately the same functionality and level of privacy as threshold signatures [12] that use cryptographic techniques for sharing between several entities the power of creating a single signature. Threshold certificates require more storage space and are less efficient because the verifier must receive and verify a set of certificates rather than just one signature. They are, however, more convenient in practice because they don not depend on the choice of cryptographic primitives and work with any standard signature algorithm. Group signatures [10] give a stronger protection of privacy when the threshold is 1 at the cost of requiring special cryptographic techniques for both the signers and the verifier.

4 ENSURING INDIVIDUAL ACCOUNTABILITY

The reservation security administrators commonly have with anonymity is that, when something goes wrong, they want to be able to find the person responsible for that. The first step of misuse prevention is usually to log the identities and actions of both users and administrators. In a certificate-based system, this would mean identifying the physical entity that misuses a service and those who authorized it. Privacy protections do not interact favorably with such accountability requirements. In particular, the key-oriented approach makes it difficult for an auditor to find out the owners of the keys.

- **Privacy problem** 8: Individual accountability conflicts with anonymity protection.

The purpose of this section is to find accountability mechanisms that minimally interfere with the privacy of the entities that are being monitored.

4.1 Encrypted identity certificates

We first consider the kinds of audit data that may be collected about certificate chains. In the certificate-based system, it is not possible to take a snapshot of the state of the access control matrix and to analyze who has what rights at a given moment. The certificates are issued locally and distributed to client hosts for storage. Their existence or creation cannot be audited without mandatory controls imposed on the issuers. Therefore, we must concentrate on monitoring the use of certificates.

The only effective place for the monitoring is where the certificates are ultimately verified. The verification is usually done by the server or its agent to check the validity of an access request but it may also take place when certificates are reduced. Thus, audit data must be collected by the servers and the hosts that reduce certificate chains.

The data that can be collected includes the access requests, authorizations, public keys and names of the clients, and the delegation chains through which the clients have obtained the access rights. A key itself is not very informative unless it can be reliably linked to the owner. The basic idea of the mechanisms we consider here is a trusted agency that ensures that the identity of the key owner can be recovered. The agency issues a certificate to the key to indicate this. Only keys approved by the trusted agency are accepted when the access rights are verified.

The first solution would be to ask all issuer and subject keys to have identity certificates from a trusted CA. In addition to the loss of privacy and the need for a CA, the cost is that the verifier of the access rights must verify all the identity certificates before allowing the access. Otherwise, there is no guarantee that the certificates are authentic.

The accountability requirement may be relaxed in several ways whose usefulness depends on the application. The identity certificate could be required only for the final client that signs the service request, for the first subject key, the last key, or any key in the authorization chain. The choice depends on the type of accountability that one wants to support. It may be enough to know one person or entity that is responsible for any damage. In particular for financial responsibility, it suffices to have only a single payer.

It is important to notice that a name or other unique identifier is not enough for holding a person or other entity accountable. There must be some way of finding the physical key owner and making it financially, legally or socially responsible for the actions of the key. That is, the important part is not knowing the name but that one cannot escape the liabilities. What the CA is actually promising in an identity certificate is that the key owner can be found and made responsible for anything signed by the key. This resembles a nonrepudiation service.

The next step is to consider how the auditing capabilities can co-exist with anonymity. Tracking the identities should only be possible for selected authorities. The names should still be kept secret from those who lack the need to know. A simple improvement is to encrypt the identity certificates. It is often known early who will verify the access rights. In that case, the identity certificates can be encrypted with the public key of the verifier (who is often the issuer of the first certificate in the chain).

Blinding the certificates may create risks for the intermediate subjects in the authorization chain. They cannot immediately verify that the credentials are acceptable by the server or have them reduced. Furthermore, the need to encrypt the certificates separately for each potential verifier creates extra work. The private decryption key for each server must be stored in the server or its agent that does the verification. It is more difficult to store a private key reliably than to store only public keys.

A better solution is to replace the identity certificates with identity escrow as will be explained next.

4.2 Identity escrow

An identity escrow agent stores the binding between a key and the owner's identity. The identity is revealed only in some well-defined circumstances. Unless these conditions are met, the key owner's identity will be known only by the escrow agent. The escrow agent issues an escrow certificate to the key as a proof that the identity of the key owner has been escrowed. The key owner will use this certificate to prove that it has submitted to the escrow. The conditions for revoking the anonymity are stated on the escrow certificate and

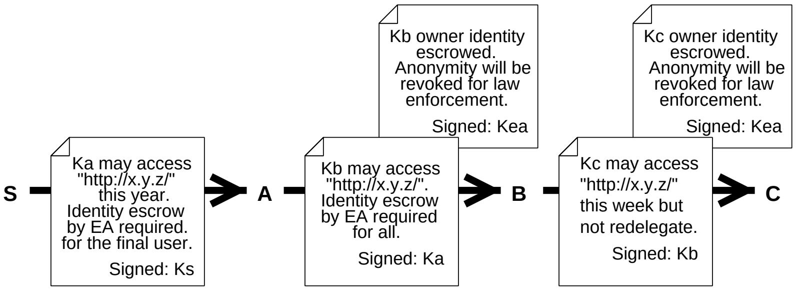
S
Ka may access "http://x.y.z/" this year. Identity escrow by EA required. for the final user.
Signed: Ks
A
Kb may access "http://x.y.z/". Identity escrow by EA required for all.
Signed: Ka
Kb owner identity escrowed. Anonymity will be revoked for law enforcement.
Signed: Kea
B
Kc may access "http://x.y.z/" this week but not redelegate.
Signed: Kb
Kc owner identity escrowed. Anonymity will be revoked for law enforcement.
Signed: Kea
C

Figure 3: Identity escrow certificates

they may vary according to the application. There could be several levels of escrow agents depending on how trustworthy they need to be from the key owner's or for the auditor's point of view.

The idea is based on the observation that it is more important to be able to track down and make financially responsible the key owner than it is to know the key owner's name. SPKI locator certificates [15] provide a guarantee that a responsible entity can be found while allowing that entity to remain anonymous until a condition specified on the locator certificate is met. The condition could be as simple as "court order" or "$100 fee". Cryptographic techniques for identity escrow were presented by Kilian and Petrank [20]. Our implementation is, however, simpler and independent of any specific cryptographic primitives.

Like identity certificates, the escrow certificates are attached to the access request. The escrow policy is enforced by the entity that verifies the certificates.

The escrow requirements are additional conditions for access and the client must somehow know the conditions to be able to fulfill them. The logical solution is to encode the escrow requirements into the authorization certificates. That way, a set of certificates itself will contain enough information to decide whether all the conditions for access have been satisfied or not. Each authorization certificate may state the public key or name of an escrow agent and a requirement to provide escrow certificates for its immediate subject, for the last subject, for any responsible key, or for all subsequent keys in the authorization chain. That is, the issuer of a certificate decides what level of escrow is needed. In an authorization chain, the escrow requirements may accumulate if many issuers ask for escrow and if they trust different escrow agents. To keep the semantics of the certificates simple, the escrow certificates themselves should not contain escrow requirements. Currently in SPKI certificates, the requirements must be encoded into the application-dependent authorization field. Eventually, they could be included in the standard and implemented in the same way as the on-line tests.

In Fig. 3, the server S requires identity escrow for the final user of the access rights and the organization A for all its affiliates. Both trust the same escrow agent EA. The figure shows the certificates that are needed for an access request by C.

Escrow agents and certificates are applicable anywhere where both anonymity and individual accountability are needed. The escrow requirements in the certificates are a kind of conditional redelegation. The cost is not much higher than that of identity certificates. The verifier must verify all the escrow agent signatures before accepting the authorization certificates. The escrow agents must follow more careful procedures than normal CAs because their mistakes cannot be detected by public scrutiny and they need to store the secret identity information reliably.

Naturally, there must be an agreement about the mechanism for holding the key owners responsible for their actions after the escrow agent revokes the anonymity. Sometimes it is not even necessary to reveal the name of the key owner if a certain level of financial liability is guaranteed. For example, the key owner may have posted a bond sum, the escrow agent may be an insurance carrier, or it may act as a representative for the key owner in solving disputes. Such financial guarantees can be expressed with SPKI insurance certificates [15] that are, like locator or escrow certificates, simply one application of the general SPKI certificate structure.

5 CONCLUSION

We analyzed the potential dangers to privacy in certificate-based access control systems and showed that authentication and privacy are not mutually exclusive. When the emphasis is shifted from identity authentication to access-right management, the identities of individuals and structural information about organizations can be protected reasonably well. The techniques described include key-oriented authorization certificates, certificate reduction, temporary and task-specific keys and a new kind of threshold certificate. Individual accountability can be enforced at a server by requiring identity escrow certificates for otherwise anonymous clients. All the suggested techniques are either a part of the SPKI specification or can be implemented with SPKI certificates.

There is a great need for improved privacy protection on the Internet. However, anonymity conflicts with the interests of many service providers who want to collect data about their clients. That is why the protection should be implemented at the basic architectural level and identification should be an additional service. This has been one of the driving forces in developing the SPKI certificate infrastructure.

ACKNOWLEDGMENTS

Tuomas Aura's work was funded by Helsinki Graduate School in Computer Science and Engineering (HeCSE) and Academy of Finland projects #44806 and #47754.

References

[1] Anonymizer, Inc. `http://www.anonymizer.com/`, January 2000.

[2] Tuomas Aura. On the structure of delegation networks. In *Proc. 11th IEEE Computer Security Foundations Workshop*, pages 14–26, Rockport, MA USA, June 1998. IEEE Computer Society Press.

[3] Tuomas Aura. Distributed access-rights management with delegation certificates. In *Secure Internet Programming – Security Issues for Distributed and Mobile Objects*, volume 1603 of *LNCS*, pages 211–235. Springer, 1999.

[4] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The role of trust management in distributed systems security. In J. Vitek and C. Jensen, editors, *Secure Internet Programming: Security Issues for Distributed and Mobile Objects*, LNCS. Springer, 1999.

[5] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proc. 1996 IEEE Symposium on Security and Privacy*, pages 164–173, Oakland, CA USA, May 1996. IEEE Computer Society Press.

[6] David F. C. Brewer and Michael J. Nash. The Chinese wall security policy. In *Proc. 1989 IEEE Symposium on Research in Security and Privacy*, pages 206–214, Oakland, CA USA, May 1989. IEEE Computer Society Press.

[7] John A. Bull, Li Gong, and Karen R. Sollins. Towards security in an open systems federation. In *Proc. ESORICS*, volume 648 of *LNCS*, pages 3–20, Toulouse, France, November 1992. Springer.

[8] *Recommendation X.509, The Directory - Authentication Framework*, volume VIII of *CCITT Blue Book*, pages 48–81. CCITT, 1988.

[9] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *Advances in Cryptology - Proc. CRYPTO '88*, volume 403 of *LNCS*, pages 319–327, Santa Barbara, CA USA, August 1988. Springer-Verlag.

[10] David Chaum and Eugene van Heyst. Group signatures. In *Advances in Cryptology - Proc. EUROCRYPT '91*, volume 547 of *LNCS*, pages 257–265. Springer-Verlag, 1991.

[11] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.

[12] Yvo G. Desmedt. Threshold cryptography. *European Transactions on Telecommunications*, 5(4):449–457, July–August 1994.

[13] T. Dierks and C. Allen. The TLS protocol. RFC 2246, IETF Network Working Group, January 1999.

[14] Carl Ellison. SPKI requirements. RFC 2692, IETF Network Working Group, September 1999.

[15] Carl Ellison, Bill Franz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylönen. SPKI examples. Internet draft, March 1998.

[16] Carl Ellison, Bill Franz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylönen. Simple public key certificate. Internet draft, July 1999.

[17] Carl Ellison, Bill Franz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylönen. SPKI certificate theory. RFC 2693, IETF Network Working Group, September 1999.

[18] Ian Goldberg and David Wagner. TAZ servers and the rewebber network enabling anonymous publishing on the world wide web. *First Monday*, 3(4), April 1998. `http://firstmonday.dk`.

[19] Li Gong. A secure identity-based capability system. In *Proc. 1989 IEEE Symposium on Research in Security and Privacy*, pages 56–63, Oakland, CA USA, May 1989. IEEE Computer Society Press.

[20] Joe Kilian and Erez Petrank. Identity escrow. In *Advances in Cryptology - Proc. CRYPTO '98*, volume 1462 of *LNCS*, pages 169–185, Santa Barbara, CA USA, August 1998. Springer.

[21] Alfred J. Menezes, Paul C. van Oorschot, and Scot A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[22] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[23] Ronald L. Rivest and Butler Lampson. SDSI — A simple distributed security infrastucture. Technical report, April 1996.

[24] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Anonymous connections and onion routing. In *Proc. 1997 IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, CA USA, May 1997. IEEE Computer Society Press.

[25] International Telecommunication Union. ITU-T recommendation X.509 (11/93) - the directory: Authentication framework, November 1993.

[26] Philip Zimmermann. *The Official PGP User's Guide*. MIT Press, June 1995.

HELSINKI UNIVERSITY OF TECHNOLOGY LABORATORY FOR THEORETICAL COMPUTER SCIENCE
RESEARCH REPORTS

HUT-TCS-A48	Tuomas Aura On the structure of delegation networks. December 1997.
HUT-TCS-A49	Tomi Janhunen Non-Monotonic Systems: A Framework for Analyzing Semantics and Structural Properties of NMR. March 1998.
HUT-TCS-A50	Ilkka Niemelä (Ed.) Proceedings of the HeCSE Workshop on Emerging Technologies in Distributed Systems. March 1998.
HUT-TCS-A51	Kimmo Varpaaniemi On the Stubborn Set Method in Reduced State Space Generation. May 1998.
HUT-TCS-A52	Ilkka Niemelä, Torsten Schaub (Eds.) Proceedings of the Workshop on Computational Aspects of Nonmonotonic Reasoning. May 1998.
HUT-TCS-A53	Stefan Rönn Semantics of Semaphores. 1998.
HUT-TCS-A54	Antti Huima Analysis of Cryptographic Protocols via Symbolic State Space Enumeration. August 1999.
HUT-TCS-A55	Tommi Syrjänen A Rule-Based Formal Model For Software Configuration. December 1999.
HUT-TCS-A56	Keijo Heljanko Deadlock and Reachability Checking with Finite Complete Prefixes. December 1999.
HUT-TCS-A57	Tommi Junttila Detecting and Exploiting Data Type Symmetries of Algebraic System Nets during Reachability Analysis. December 1999.
HUT-TCS-A58	Patrik Simons Extending and Implementing the Stable Model Semantics. April 2000.
HUT-TCS-A59	Tommi Junttila Computational Complexity of the Place/Transition-Net Symmetry Reduction Method. April 2000.
HUT-TCS-A60	Javier Esparza, Keijo Heljanko A New Unfolding Approach to LTL Model Checking. April 2000.
HUT-TCS-A61	Tuomas Aura, Carl Ellison Privacy and accountability in certificate systems. April 2000.

ISBN 951-22-5000-4
ISSN 0783-5396