# Reasoning about Accountability within Delegation

Bruno Crispo[1] and Giancarlo Ruffo[2]

[1] Cryptomathic S.p.A.
Corso Svizzera, 185 - 10149 Torino, Italy
crispo@cryptomathic.it

[2] Dipartimento di Informatica - Università di Torino
Corso Svizzera, 185 - 10149 Torino, Italy
ruffo@di.unito.it

**Abstract.** We propose a framework for the analysis of delegation protocols. Our framework allows to analyse how accountability is transferred (or kept) by delegator when he transfers some of her rights to delegate. The ability to trace how accountability is distributed among principals of a system is crucial in many transactions that have a legal value, because accountability is usually a prerequisite to guarantee other well known security properties (e.g., non repudiation). Our approach starts from the notion of "provability" to formalise accountability. Then, we introduce new specifications for the analysis of delegation protocols and the distribution of credentials necessary to exercise delegated rights.

## 1 Introduction

In many e-commerce applications, as in the real-life, electronic transactions must be able to guarantee at least the same degree of accountability provided by conventional transactions. For example, let us consider the case of a manager delegating her system administrator to backup her files containing important documents. In case something goes wrong and the documents will be unrecoverable, it would be useful for both, the manager and the system administrator, to have mechanisms that help them to prove to a third party their behaviour and doing so to determine accountability of facts. From this example it is clear the importance of the property of accountability that we define as:
*the property whereby the association of a principal with an object, an action or a right can be proved to a third party*
This paper provides an original contribution to the problem of the analysis of protocols that requires accountability. Among all the protocols that require this property we will focus our attention to delegation protocols.

This is motivated by the fact that delegation is usually the general mechanism used to transfer accountabilities among principals. Besides a lot of work has been done on the analysis of protocols, but few of these analysis have considered delegation protocols.

## 2 Accountability

For our analysis we start from a framework introduced by Kailar [5] specifically for analysing and describing accountability in order to analyse this property in delegation protocols.

This framework is based on the notion of *provability*, that is the ability of participants in a protocol to *prove* a statement to a third party, that is the basis for accountability. A participant can prove a statement to any other principal if he can convince the latter about the statement. The proof of a statement $x$ is generically defined as the ability starting from known assumptions, to produce a set of statements that can convince any other principal about $x$. In practice it is enough (and easier) to convince a particular third party (a judge) rather than all the other principals that did not participate to the protocol.

We agree with Kailar that his approach is more suitable to analyse accountability rather than other approaches based on *belief* [1] and its evolution within the protocol, because these approaches focus on what can be proved only by the participants of the protocol, while the point of view of external observers is essential to accountability.

In this section, we provide a short review of the basics of the adopted framework, referring to [5] for a more detailed description.

In section 2.4, we will introduce new postulates that permit to analyse delegation of accountability in communication protocols.

Finally, in section 3, we will analyse two communication protocols with support for delegation: the SPX protocol [8] and the Delegation of Accountability protocol [2]. Our analysis will show the usability of our approach.

### 2.1 Symbols and Concepts

In a generic communication protocol, we have a group of *principals* ($A$, $B$, ...), that exchange *messages* within each other. During the analysis of a protocol, we want to focus on the ability of principals to prove the origin of these messages. The *statement* made by each message is the message interpretation; statements are denoted by lower-case letters ($x$, $y$, ...). A

*proof* of a statement $x$ is something that convinces another principal of statement $x$. We are not worrying about the steps of a proof, because they largely depend on the environment where the protocol is designed to work.

Considering that we defined accountability as the property whereby the association of a principal and an object/action/right can be proved to a third party, we need to introduce objects, actions and rights into our language. We will denote a *set of rights* with a greek upper-case letter ($\Omega, \Delta, ...$). Observe that in the rest of a paper, the term *right* is used also to indicate an object (right to *use* a given object) or an action (right to *do* a given action).

To improve the readability of the paper, we will avoid to introduce and use new mathematical symbols. Instead, we will use common phrases like "can prove" or "can exercise" written concatenated (i.e., "CanProve" or "CanExercise"). Moreover, we will introduce only concepts and postulates that are needed to understand the paper and the process of analysing a delegation protocol for proving the accountability property. Informal descriptions of these concepts are given below:

$A$ **CanProve** $x$**:** Principal $A$ can prove the statement $x$ to any third party $B$. This implies that $A$ is able to perform a sequence of operations that lead to prove statement $x$ to a principal $B$, whoever is $B$. This proof does not reveal any secret $y \neq x$.
This is a *Strong proof*, because a principal $A$ can prove the given statement to everyone. We talk about *Weak proof* if the ability of the prover permits to prove the given statement only to another principal. In this case, we can write "$A$ CanProve $x$ to $B$", where $A$ and $B$ are involved principals. In this paper we will use only strong proof.

$K$ **Authenticates** $A$**:** The key $K$ can be used to authenticate the signature of principal $A$. As a consequence, we can associate $A$ to any statement encrypted with $K^{-1}$. $K$ and $K^{-1}$ are public and private counterparts of an asymmetric key pair.

$x$ **in** $m$**:** $x$ is the interpretation of a (group of) field(s) in message $m$. This interpretation is protocol specific.

$A$ **Says** $x$**:** Principal $A$ is responsible of statement $x$. In other words, $A$ is accountable of $x$. As a consequence, $A$ is accountable for every statement implied by $x$. Moreover, if $A$ says any statement composed by two or more parts, she is accountable for each part:

$$\frac{A \text{ Says } (x, y)}{A \text{ Says } x}$$

$A$ **Receives** $m$ **SignedWith** $K^{-1}$**:** This tells that principal $A$ receives a message $m$ signed with a private key $K^{-1}$. If $x$ is the message interpretation of $m$ (of the interpration of one of the fields of $m$), we can use the following postulate:

$$\frac{A \text{ Receives } m \text{ SignedWith } K^{-1}; x \text{ in } m}{A \text{ Receives } x \text{ SignedWith } K^{-1}}$$

$A$ **isTrustedOn** $x$**:** Principal $A$ is trusted on statement $x$, i.e., $A$ has the authority to endorse $x$ and is liable for making $x$. If principal $A$ is *globally* trusted, then $A$ is trusted on $x$ by all principals[1].

The following definitions of "CanExercise" appear for the first time in this paper.

$A$ **CanExercise** $\Omega$**:** This denotes the fact that principal $A$ can exercise the rights listed in $\Omega$. In an access control environment, a principal can exercise a right under some requirements[2]. We use "CanExercise" only to associate principals to rights, in accordance to our definition of accountability.

$A$ **CanExercise** $\Omega$ **with** $K$**:** If we want to specify the authentication key that a principal uses to exercise her rights, we can use this statement. Here, a principal $A$ CanExercise the set of rights $\Omega$ using $K$ as her authentication key. Of course, the statement "$K$ Authenticates $A$" must be provable.

### 2.2 Assumptions

Some assumptions related to security constraints must be respected before validating analysis results. The digital signature scheme considered in this framework is public-key encryption paradigm based. *Signature algorithms* are assumed to be strong enough: (1) to be undisputably associated with a single user; (2) to resist against the search of another principal's private key, independently by the available computing power, for a sufficient

[1] If a principal is trusted on a statement by only another (or a group of) principal(s), we use the notion of *non global trustness*. In this paper we will only talk about global trust, even if more loosely: principal $A$ is trusted on $x$ by all principals in the intended audience of a proof.

[2] For example, a principal can read/write the files of directory */Doc/Sec* only if she belongs to the *Security* group. We are not concerning on aspects of how this requirements are checked or how this rights are assigned to principals by a system administrator. These aspects are strongly dependent on a given environment.

period of time; (3) to withstand birthday attacks. Moreover, signature algorithms are assumed to provide message origin authentication, message content integrity and message sender non-repudiation. Finally, signature algorithms do not require the consent of the signer.

Another important group of assumptions is related to *Trustness*: principals are trusted not to share their private keys with other principals with whom they do not wish to be accountable, i.e., we trust principals that use caution to share their keys. Moreover, a principal is assumed to trust a statement if she is an authority of the given statement, or if she is convinced on the validity of the statement by a trusted party.

Other important assumptions are about message integrity, availability of services and certificate revocation. It is not possible to fake a signed message or to compute another private key that can be accepted as the authentic signature (*message integrity*); if $A$ *CanProve* $x$, then we assume that, independently of the availability of the communication service, we can assure that $A$ has the ability to send all the messages for proving $x$ (*availability of service*); finally, statements proved by revoked public keys are considered valid only if the statements were signed when the related certificates were also valid (*certificate revocation*).

### 2.3 Postulates

Postulates introduced here are applicable to the analysis of accountability properties in electronic communication protocols. All postulates are given in the form:

$$\frac{P; Q}{R}$$

where $P$ and $Q$ are the premises of the rule: if they hold simultaneously, then the consequence statement $R$ is true.

**Conjunction:** If $A$ can prove that $x$ is true and she can also prove that also $y$ is true, then $A$ can prove that the conjunction $x \wedge y$ is true.

$$\textbf{Conj: } \frac{A \text{ CanProve } x; A \text{ CanProve } y}{A \text{ CanProve } (x \wedge y)}$$

**Inference:** If $A$ can prove statement $x$ and if $x$ implies $y$, then $A$ can prove that $y$ is true.

$$\textbf{Inf: } \frac{A \text{ CanProve } x; x \Rightarrow y}{A \text{ CanProve } y}$$

**Accountability property of digital signatures:** The following postulate can be used to prove that principals are accountable for messages they signed.

$$\textbf{Sign:}\ \frac{\begin{array}{c}A \text{ Receives } m \text{ SignedWith } K^{-1}; x \text{ in } m; \\ A \text{ CanProve } (K \text{ Authenticates } B)\end{array}}{A \text{ CanProve } (B \text{ Says } x)}$$

That is, when principal $A$ receives a message $m$ signed with a key $K^{-1}$ and $A$ can prove that this key belongs to $B$, as a consequence $A$ can prove that $B$ is accountable for any statement $x$, where $x$ is a message interpretation of $m$.

**Trust relationships:** In digital signatures schemas, a proof of a statement $x$ can be given also by showing that $x$ has been endorsed by a trusted authority of $x$, i.e., $A$ is an authority on $x$ and she says $x$. As a consequence, $A$ can prove that $x$ is true. This is based on what we said in section 2.2: if $A$ is trusted on a given statement then is able to prove it to another principal.

Morevover, if a principal $A$ can prove that another principal is able to prove a statement $x$, then $A$ can prove $x$.

**Trust** postulate is a corollary of the previous considerations:

$$\textbf{Trust:}\ \frac{\begin{array}{c}A \text{ CanProve } (B \text{ Says } x); \\ A \text{ CanProve } (B \text{ isTrustedOn } x)\end{array}}{A \text{ CanProve } x}$$

## 2.4 A specification of the Framework: the *CanExercise* Postulates

This section introduces the formalization of the concept of a principal that can exercise a right (or a set of rights).

A principal can exercise a right if another principal gave her the related permissions. These permissions can be given by a trusted authority (i.e., a system administrator), and can be delegated to another principal, whom, after delegation, can exercise the transferred rights.

In a generic delegation, principal $A$ can delegate another principal $B$ to exercise the set of rights $\Omega$ only if $A$ has the ability to exercise them. Moreover, $A$ must be accountable for having delegated $B$ to exercise $\Omega$, and, finally, $B$ must be authenticated when she exercise $\Omega$. The following

postulate formalizes these ideas:

$$\textbf{CanExercise1:}\ \frac{\begin{array}{c}A \text{ CanExercise } \Omega;\\ A \text{ Says (delegation of } \Omega \text{ to } B);\\ (K_{Del} \text{ Authenticates } B);\end{array}}{B \text{ CanExercise } \Omega \text{ with } K_{Del}}$$

That is, principal A can exercise the set of rights $\Omega$ and she delegates $B$ to exercise these rights. Key $K_{Del}$ authenticates principal $B$: when $B$ will exercise $\Omega$, she will be authenticated using $K_{Del}$.

to describe the power of a principal to exercise a given set of rights, we can omit the specification of this key. The following postulate relates both ways to use "CanExercise" clause.

$$\textbf{CanExercise2:}\ \frac{A \text{ CanExercise } \Omega \text{ with } K}{A \text{ CanExercise } \Omega}$$

In our analysis, we want to proof the accountability of a principal on a set of rights that have been delegated by another principal. In other words, the goal of such a proof is to show that:

*delegate* CanProve (*delegate* CanExercise $\Omega$ with $K_{Del}$)
where $K_{Del}$ is the delegation key of the given protocol.

During analysis of delegation protocols, we will use postulates **CanExercise1** and **CanExercise2** in conjunction with **Inf** postulates, in order to unify "CanProve" and "CanExercise".

Another important goal to verify during analysis of a delegation protocol is the ability for delegator to prove that she is not associated with delegate's actions. When a set of rights $\Omega$ has been transferred from $A$ to $B$ and principal $B$ is exercising $\Omega$ using delegation key $K_{Del}$, then principal $A$ is not accountable for this $B$'s activity. This second generic goal can be formalized with the following statement:

*delegator* CanProve ($K_{Del}$ Authenticates *delegate*)

## 3 Analysis of Delegation Protocols

In this section, we show some examples of protocols analysis. In particular, we apply our analysis framework to SPX [8] and to the Delegation of Accountability protocol [2]. In these two analyses it will be possible to show the difference between two different delegation's philosophies: SPX permits grantor to delegate grantee the possibility to act on grantor's behave; in the other approach, grantor transfers the accountability on a set of rights of her own.

### 3.1 SPX with support for Delegation

**Protocol description** In SPX [8], principals use *authentication tokens* to authenticate each other. The authentication token permits the secure exchange of a session key. A simplified version of SPX is analysed in [5] in order to verify accountability properties. In this section, we summarize the content of the previous analysis and we will show that this protocol doesn't allow accountability on a set of transferred rights.

Involved principals are: a claimant (C), a certificate distribution center (CDC), and a server (S). Moreover, we have also principals $TA_1$ and $TA_2$, that, together with $CDC$, play the role of trusted authorities.

The goal of the protocol is for $S$ to securely receive a delegation key from $C$. In this delegation context, principal $C$ authorizes another principal ($S$) to act on her behalf by sharing a set of rights with $C$ for a given period of time. The protocol is not designed for delegation of accountability, because the transferred rights will be still accountable to $C$.

The protocol description is the following:

1. $C \rightarrow CDC : S$
2. $CDC \rightarrow C : K^{-1}_{CDC}(K^{-1}_{TA_1}(S, K_S, TA_1))$
3. $C \rightarrow S : K^{-1}_{C}(K_{Del}, T), K_S(K_{des}), K_{des}(K^{-1}_{Del})$
4. $S \rightarrow CDC : C$
5. $CDC \rightarrow S : K^{-1}_{CDC}(K^{-1}_{TA_2}(C, K_C, TA_2))$
6. $S \rightarrow C$ : *Response (accept/reject)*

Server $S$ plays the role of the verifier of the claimant's credential. The protocol starts with the request of $C$ for $S$'s public key (message 1). This request is send to the certificate distribution center, that replies (message 2) with a certificate of $S$, issued by the trusted authority $TA_1$. This certificate is encrypted with $CDC$'s private key. $C$ sends her delegation public key ($K_{Del}$) to $S$ (message 3), signing it with her authentication key ($K^{-1}_{C}$). $K_{Del}$ is valid for a period of time $T$. Moreover, $C$ sends to $S$ a symmetric session key ($K_{des}$) encrypted with $S$'s public key ($K_S$). $C$ encrypts the private part of the delegation key with the session key $K_{des}$ and she also sends it to $S$. Finally $S$ asks for $C$'s certificate to $CDC$ (message 4 and 5) and after receiving the certificate, $S$ verifies $C$'s credentials and replies to $C$ the response (message 6).

**Reformulating the Protocol** The protocol has been reformulated with the adopted notation by Kailar in [5]. We report here the protocol message interpretation described in the previous analysis. Only messages 2, 3 and 5 were considered relevant to the analysis:

2. $C$ Receives ( ( ($K_S$ Authenticates $S$)
SignedWith $K_{TA_1}^{-1}$) SignedWith $K_{CDC}^{-1}$)
3. $S$ Receives ( ( ($K_{Del}$ Authenticates $C$ during $T$)
SignedWith $K_C^{-1}$))
5. $S$ Receives ( ( ($K_C$ Authenticates $C$)
SignedWith $K_{TA_2}^{-1}$) SignedWith $K_{CDC}^{-1}$)

**Protocol Analysis** As we reminded at the beginning of this section, the delegation goal pointed by the Kailar's analysis was to verify the delegate's ability of proving that the delegation key authenticates delegator. In other words, the goal of the analysis showed by Kailar was:

**[Goal]** $S$ CanProve ($K_{Del}$ Authenticates $C$)

Principal $C$ can exercise the transferred set of rights, but $S$ will still be accountable for them, because $K_{Del}$ authenticates her.

As we said in section 2.4, we wish to show that: $C$ CanProve ($K_{Del}$ Authenticates $S$),
in order to give $C$ the possibility to prove her independency by delegate's actions. If we would be able to show the previous statement, it will be true together with **Goal** statement proved by Kailar, meaning that $K_{Del}$ authenticates both delegator and delegate. In this case, we lose accountability property. As a consequence, SPX protocol does not support delegation of accountability.

### 3.2 The Delegation of Accountability Protocol

**Protocol description** The protocol we propose is based on delegation tokens (Gasser *et al.* [4], Sollins [7], Low *et al.* [6]).

This protocol allows principals to delegate their own accountability to any other principals. It assumes that each principal can generate public-key pairs and has access to a digital signature service. Moreover, it assumes that each principal can get the public key(s) needed to verify digital signatures that she may receive, included the keys used for authentication purposes. We do not specify in our description the part concerned with authentication of principals, which we assume already done when the delegation protocol starts. The delegation protocol is specified as follows:

1. G $\rightarrow$ G': G, G', M, $K_G^{-1}$(M)
   where M=[G wishes to delegate to G' accountability for $\Omega$]
2. G' $\rightarrow$ G: G', G , M', $K_{G'}^{-1}$(M')
   M'=[G' accepts $\Omega$ and she will exercise $\Omega$ using $K_{Del}$]
3. G $\rightarrow$ G': T = [G, G', M'', $K_G^{-1}$(M'')]
   M''=[$\Omega$, LS, $K_G, K_{Del}$]

where $G$ is the grantor, $G'$ is the grantee, $\Omega$ is the set of delegated rights and $LS$ is the time span of delegation token $T$. $(K_G, K_G^{-1})$ and $(K_{G'}, K_{G'}^{-1})$ are respectively the authentication key pairs of grantor and grantee, $(K_{Del}, K_{Del}^{-1})$ is the delegation key pair that grantee will use to exercise $\Omega$. In message (3), a key rather than a name is used to identify the grantor so if an attacker succeeds to masquerade as the grantor he cannot fraudulently delegate grantor's accountability because he still does not know the key $K_G^{-1}$ necessary to be able to do it.

The grantor is the only one that can enable the grantee to use $\Omega$: the delegation token contains $M''$, which specifies the characteristics of the present delegation, and also it contains $M''$ signed by the grantor. When the grantee wishes to use the delegated rghts she must present [T, $K_{Del}^{-1}$(T)] to the end-point, followed by the request of the specific service she wants[3]. The end-point will check the privileges carried in the delegation token against her access control policy. The end-point can be any principal of the system because the token is verifiable by all the components of the system[4]. Thus all the principals can verify the correctness of the delegation token after they get the grantor's and grantee's public key from the authentication service in order to authenticate them in the first two messages of the protocol.

**Reformulating the Protocol** The protocol can be reformulated in terms of the described notation:

1. $G'$ Receives ( (G wishes to delegate to G' accountability for $\Omega$) SignedWith $K_G^{-1}$)
2. $G$ Receives ( ($K_{Del}$ Authenticates $G'$) SignedWith $K_G'^{-1}$)
3. $G'$ Receives ( (delegation of $\Omega$ to $G''$) SignedWith $K_G^{-1}$)

Now we have to list the implicit assumptions and apply the inference rules of the adopted logic to the assumptions and to the messages of the protocol in order to prove our goal: the delegate is accountable to exercise transferred rights.

**Goal and Initial State Assumptions** Our primary goal is:

**[Goal1]** $G'$ CanProve ($G'$ CanExercise $\Omega$ with $K_{Del}$)

Let us observe that, if we prove the goal, with the application of **CanExercise2** and **Inf** postulates, we can show the more general fact that:

[3] In such a framework, if grantee $G'$ is not honest and step 3 does not take place, she cannot exercise the delegated rights with only delegation key $K_{Del}$ because she misses the delegation token.

[4] We are assuming that an authentication service is available.

$G'$ CanProve ($G'$ CanExercise $\Omega$)

We wish also to show that grantor is able to prove that the delegation key authenticates grantee: if $G'$ will exercise $\Omega$ using $K_{Del}$, $G$ cannot be accountable for this. As a consequence, the second goal of our analysis is:

**[Goal2]** $G$ CanProve ($K_{Del}$ Authenticates $G'$ )

The initial state assumptions follow here:

**[A1]** $G$ CanProve ($K_G$ Authenticates $G$);

**[A2]** $G'$ CanProve ($K'_G$ Authenticates $G'$)

**[A2']** $G'$ CanProve ($K_{Del}$ Authenticates $G'$)

**[A3]** $G'$ CanProve ($G$ CanExercise $\Omega$)

**[A4]** $G$ CanProve ($G'$ isTrustedOn ($K_{Del}$ Authenticates $G'$)

**[A5]** $G$ CanProve ($K'_G$ Authenticates $G'$)

**[A6]** $G'$ CanProve ($K_G$ Authenticates $G$)

Assumptions **A1**, **A2** and **A2'** state that the association between principals and their public keys can be proved.

Of course, we assume that $G'$ can prove that $G$ is able to exercise the set of rights $\Omega$ (assumption **A3**). $G$ is delegating $G'$ to exercise $\Omega$, but $G'$ must be convinced that $G$ owns these rights.

We assume also that principal $G'$ is trusted when announcing its own delegation key, because she is responsible of the messages signed with this key (assumption **A4**).

Finally, in the protocol we did not specify the part concerned with authentication of principals, because we are focusing on the delegation part. We can assume that the generic goals of a public key distribution protocol are reached before the delegation protocol starts (i.e., using a certificate distribution center, as in the SPX protocol). As a consequence, we can make assumptions **A5** and **A6**.

**Analysis** Applying **Sign** postulate on message 3 and **A6**, we obtain:

**[S1]** $G'$ CanProve ($G$ Says (delegation of $\Omega$ to $G'$))

Using **Conj** postulate on **A3**, **S1** and **A2'**, the following statement is true:

**[S2]** $G'$ CanProve ($G$ CanExercise $\Omega$,
$G$ Says (delegation of $\Omega$ to $G'$),
$K_{Del}$ Authenticates $G'$)

Finally, we obtain **Goal1**, using **Inf** and **CanExercise1** postulates on statement **S2**:

**[Goal1]** $G'$ CanProve (G' CanExercise $\Omega$ with $K_{Del}$)

We can apply **Sign** postulate to message 2 and **A5** assumption to show that:

**[S3]** $G$ CanProve ($G'$ Says ($K_{Del}$ Authenticates $G'$))
Finally, **Goal2** is inferred by **Trust** postulate using **S3** and **A4** as premises $\Box$.

## 4 Conclusions

Despite its importance in supporting any commercial and financial transaction, accountability has been usually negletted in the formalisation of protocols. Also, other important security properties (e.g., non-repudiation) rely on accountability and on the possibility to examine unforgeable evidence collected by the party during the execution of a transaction [3]. In this paper, we tried to raise the attention to this issue and in particular we introduced a framework to reason about accountability in the particular case of delegation protocols. Delegation protocols aim to perform the hand-over of rights from delegator to delegate. Our studies however, proved that many of them do not consider the important issue of the accountability associated to those rights. As we said, this lack of specification can vanish or jeopardise the subsequent use of the delegated rights in applications where accountability is required in case of possible disputes (i.e., electronic commerce).

## References


1. M. Burrows, M. Abadi, and R.M. Needham. A logic of Authentication. *ACM Transaction on Computer Systems*, 8(1), Febraury 1990.
2. B. Crispo. Delegation protocols for electronic commerce. In *Proceedings of the 6th IEEE Symposium on computers and communications (ISCC01), Hammamet, Tunisia*. IEEE press, 2001.
3. B. Crispo and M. Lomas. A Certification Scheme for Electronic Commerce. In *Security Protocol Workshop*, volume LNCS vol. 1189. Springer-Verlag, 1997.
4. M. Gasser and E. McDermott. An Architecture for Practical Delegation in a Distributed System. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1990.
5. R. Kailar. Reasoning about Accountability in Protocol for Electronic Commerce. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1995.
6. M.R. Low and B. Christianson. Self Authenticating Proxies. *Computer Journal*, 37(5):422–428, October 1994.
7. K.R. Sollins. Cascaded Authentication. In *Proceedings of the IEEE Conference on Security and Privacy*, April 1988.
8. J.J. Tardo and K. Alagappan. SPX: Global Authentication Using Public Key Certificates. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1991.