Department of Communications and Networking

# Access Control in Distributed Systems using SPKI Authorisation Certificates

Yki Kortesniemi

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
**DOCTORAL DISSERTATIONS** 63/2015

# Access Control in Distributed Systems using SPKI Authorisation Certificates

**Yki Kortesniemi**

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall S1 of the school on 29 May 2015 at 12.

**Aalto University**
**School of Electrical Engineering**
**Department of Communications and Networking**

**Supervising professor**
Prof. Jukka Manner

**Thesis advisors**
Prof. Hannu Kari
Dr. Mikko Särelä

**Preliminary examiners**
Prof. Gerald Q. Maguire Jr., KHT Royal Institute of Technology, Sweden
Doc. Göran Schultz, University of Turku, Finland

**Opponent**
Prof. Valtteri Niemi, Helsinki University

Aalto University publication series
**DOCTORAL DISSERTATIONS** 63/2015



ISBN 978-952-60-6193-1 (printed)
ISBN 978-952-60-6194-8 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-6194-8

Unigrafia Oy
Helsinki 2015

Finland

441 697
Printed matter

Aalto University

# Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

**Author**
Yki Kortesniemi

**Name of the doctoral dissertation**
Access Control in Distributed Systems using SPKI Authorisation Certificates

**Publisher** School of Electrical Engineering

**Unit** Department of Communications and Networking

**Series** Aalto University publication series DOCTORAL DISSERTATIONS 63/2015

**Field of research** Networking Technology

**Manuscript submitted** 23 February 2015 **Date of the defence** 29 May 2015

**Permission to publish granted (date)** 9 April 2015 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

**Abstract**

In distributed systems, the ability to effectively manage access to a large number of resources can be challenging. The situation becomes even more difficult, when there are limited computational resources or network availability to implement the access control solution. Examples are Internet of Things (IoT) applications, such as the many internet-connected devices at home. To make them easy to use, there has to exist a relatively simple way to manage the large number of devices and to, e.g., grant temporary access to some of them for a visiting friend. In this dissertation, I examine how the problem can be overcome with the Simple Public Key Infrastructure (SPKI), which expresses access rights as cryptographically signed authorisation certificates.

I approach the issue from several angles. First, I develop a phase model to analyse the access control process / certificate life-cycle and use it to study SPKI and other certificate technologies for access control while pointing out areas requiring future work. Although SPKI has been studied for some 20 years, standardisation has not been completed. I identify three important missing parts of SPKI in utilising the certificates, as well as in managing and validating online conditions. I also expand the SPKI model to support usage quotas. I then design solutions for all these areas and analyse the resultant system for its applicability, scalability, security and usability. Of particular interest are system performance and privacy. My final focus area is certificate chain reduction, a proposed way to improve performance and privacy of SPKI. I study the approach in detail, identify the relevant design choices for the systems architect, and design a protocol for requesting reductions.

For performance evaluation we implemented a prototype, which demonstrates that even modern embedded devices can reach transaction times of one second including all communication delays and using only a software implementation for cryptography. We also found that the transaction was over 40 % faster with chain reduction thus proving the promise of improved performance. Using such reductions does requires a reduction server, but calculations from our use case show that even with pessimistic assumptions, a single reduction server can support millions of users thus making scalability a manageable issue. Privacy-wise, SPKI is a good solution with support for anonymous identities - and chain reduction can further improve user privacy by hiding additional information. Finally, all my use cases demonstrate the same certificate chain structure, an hourglass-model, which I hypothesise is prevalent in many other systems, as well. It forms natural basis for reduction and provides for a consistent performance regardless of certificate chain length.

**Keywords** SPKI, authorisation certificates, access control, distributed systems

**ISBN (printed)** 978-952-60-6193-1 **ISBN (pdf)** 978-952-60-6194-8

**ISSN-L** 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

**Location of publisher** Helsinki **Location of printing** Helsinki **Year** 2015

**Pages** 163 **urn** http://urn.fi/URN:ISBN:978-952-60-6194-8

**Tekijä**
Yki Kortesniemi

**Väitöskirjan nimi**
Hajautettujen järjestelmien pääsynhallinta SPKI-valtuussertifikaattien avulla

**Julkaisija** Sähkötekniikan korkeakoulu

**Yksikkö** Tietoliikenne- ja tietoverkkotekniikan laitos

**Sarja** Aalto University publication series DOCTORAL DISSERTATIONS 63/2015

**Tutkimusala** Tietoverkkotekniikka

**Käsikirjoituksen pvm** 23.02.2015 **Väitöspäivä** 29.05.2015

**Julkaisuluvan myöntämispäivä** 09.04.2015 **Kieli** Englanti

☐ **Monografia** ☒ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

**Tiivistelmä**

Hajautetuissa järjestelmissä lukuisien resurssien tehokas pääsynhallinta voi olla haastavaa. Tilanteen tekee vielä vaikeammaksi, jos pääsynhallinnan toteuttamiseen on vain rajallisesti laskentatehoa tai verkkoyhteyksiä. Esimerkki tästä ovat Esineiden Internetin (Internet of Things) sovellukset kuten monet internetiin liitetyt laitteet kotona. Jotta laitteiden käyttö olisi helppoa, tarvitaan yksinkertainen tapa hallita niitä ja antaa esimerkiksi vierailevalle ystävälle tilapäinen pääsy. Tässä väitöskirjassa tutkin, miten Simple Public Key Infrastructure (SPKI) voi ratkaista ongelman. SPKI esittää pääsyoikeudet kryptografisesti allekirjoitettuina sertifikaatteina.

Lähestyn asiaa useasta näkökulmasta. Ensinnäkin, kehitän vaihemallin, jolla analysoin pääsynhallintaprosessia ja sertifikaattien elinkaarta, sekä hyödynnän sitä tutkiakseni SPKI- ja muita sertifikaatteja tunnistaen samalla jatkokehitystä vaativia alueita. Vaikka SPKI-sertifikaatteja on tutkittu jo noin 20 vuotta, standardointia ei ole viety loppuun. Tunnistan kolme tärkeää puuttuvaa osaa SPKI-sertifikaateissa liittyen sertifikaattien käyttöön sekä verkotettujen voimassoloehtojen hallintaan ja validointiin. Laajennan myös SPKI-mallia tukemaan käyttökiintiöitä. Sen jälkeen suunnittelen ratkaisut kaikkiin näihin osa-alueisiin sekä analysoin järjestelmän toimintaa soveltuvuuden, skaalautuvuuden, turvallisuuden ja käytettävyyden näkökulmasta. Keskityn erityisesti järjestelmän suorituskykyyn ja yksityisyyteen. Viimeinen fokusalueeni on SPKI-sertifikaattiketjujen reduktio, suorituskyvyn ja yksityisyyden parantamiseksi ehdotettu menetelmä. Tutkin menetelmää yksityiskohtaisesti, tunnistan järjestelmäarkkitehdin kannalta keskeiset suunnitteluvalinnat sekä suunnittelen protokollan reduktioiden pyytämiseen.

Suorituskyvyn arvioimiseksi toteutimme prototyypin, joka osoittaa, että jopa moderni sulautettu järjestelmä voi saavuttaa alle sekunnin vasteajan sisältäen kaikki tiedonsiirtoviiveet ja käyttäen ohjelmallisesti toteutettua kryptografiaa. Havaitsimme myös, että ketjureduktiolla transaktio nopeutui yli 40 % saavuttaen näin lupauksen paremmasta suorituskyvystä. Tällaisten reduktioiden hyödyntäminen vaatii järjestelmään erillisen reduktiopalvelimen, mutta laskelmat käyttötapauksestamme osoittavat, että pessimistisilläkin oletuksilla yksi reduktiopalvelin voi palvella miljoonia käyttäjiä tehden järjestelmästä skaalautuvan. Yksityisyyden kannalta SPKI on hyvä ratkaisu, joka tukee anonyymejä identiteettejä ja ketjureduktion avulla voidaan piilottaa vielä lisää tietoa. Lopuksi, kaikki käyttötapaukseni noudattivat samaa sertifikaattiketjurakennetta, tiimalasimallia, jonka oletan esiintyvän monissa muissakin järjestelmissä. Se muodostaa luontevan pohjan reduktioiden käytölle ja mahdollistaa tasaisen suorituskyvyn riippumatta sertifikaattiketjujen pituudesta.

**Avainsanat** SPKI, valtuussertifikaatit, pääsynhallinta, hajautetut järjestelmät

**ISBN (painettu)** 978-952-60-6193-1 **ISBN (pdf)** 978-952-60-6194-8

**ISSN-L** 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

**Julkaisupaikka** Helsinki **Painopaikka** Helsinki **Vuosi** 2015

**Sivumäärä** 163 **urn** http://urn.fi/URN:ISBN:978-952-60-6194-8

# Preface

First, I would like to express my great gratitude to my supervisor Prof. Jukka Manner for his support and advice on constructing this work. I am also deeply grateful to my instructors Hannu Kari and Mikko Särelä for the guidance and encouragement they have given me over these years.

I also want to thank my co-workers at HIIT for the many insightful discussions throughout the years. I am especially indebted to my co-authors Tero Hasu, Jonna Särs, Kristiina Karvonen, Antti Latva-Koivisto, Timo Kiravuo, Mikko Särelä and Hannu Kari for their contributions to four of the publications in this dissertation. I am also thankful to Olli Pitkänen and Martti Mäntylä for their helpful advice, and Lauri af Heurlin for his work on the prototype implementation.

This dissertation was completed thanks to the financial support of Helsinki Institute for Information Technology, Graduate School on Networks for Information Society (GSNIS), Qentinel Oy, and Aalto University School of Electrical Engineering.

Finally, I would like to thank my family and friends for their support throughout the writing process.

Espoo, May 5, 2015,

Yki Kortesniemi

# Contents

# List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

**I** Yki Kortesniemi, Mikko Särelä. Survey of Certificate Usage in Distributed Access Control. *Computers & Security*, Volume 44, July, pp. 16-32, 2014.

**II** Yki Kortesniemi. Validity Management in SPKI. In *Proceedings of the 1st Annual PKI Research Workshop*, 2002.

**III** Yki Kortesniemi, Tero Hasu, Jonna Särs. A Revocation, Validation and Authentication Protocol for SPKI Based Delegation Systems. In *Proceedings of the Network and Distributed System Security Symposium*, 2000.

**IV** Yki Kortesniemi. SPKI Performance and Certificate Chain Reduction. In *Proceedings of Informatik 2002, Workshop on "Credential-basierte Zugriffskontrolle in offenen, interoperablen IT-Systemen"*, 2002.

**V** Yki Kortesniemi, Timo Kiravuo, Mikko Särelä, Hannu Kari. Chain Reduction of Authorisation Certificates. *International Journal of Security and Networks*, Accepted for publication, 2015.

**VI** Kristiina Karvonen, Yki Kortesniemi, Antti Latva-Koivisto. Evaluating Revocation Management in SPKI from a User's Point of View. In *Proceedings of Human Factors in Telecommunication*, 2001.

# Author's Contribution

### Publication I: "Survey of Certificate Usage in Distributed Access Control"

This paper surveys how certificates are used in the different phases of access control. The author is responsible for the initial idea behind the paper and most of the analyses.

### Publication II: "Validity Management in SPKI"

This paper examines the life-cycle of access control rights and presents a protocol for managing the validity of SPKI certificates. The author is solely responsible for the paper.

### Publication III: "A Revocation, Validation and Authentication Protocol for SPKI Based Delegation Systems"

This paper extends the SPKI model to managing limited resources and proposes a mechanism for implementing this. It also discusses the requirements for authorisation certificate revocation, introduces a new revocation methods (a chain of short-lived certificates) and defines a protocol for verifying SPKI certificate validity at the time of usage. The author was responsible for the initial idea behind this paper and was the chief designer for the proposed solutions.

### Publication IV: "SPKI Performance and Certificate Chain Reduction"

This paper discusses the problems resulting from the need to evaluate long certificate chains. The author is solely responsible for the paper.

### Publication V: "Chain Reduction of Authorisation Certificates"

This paper analyses the practical issues in creating chain reductions, evaluates the effect chain reduction has on performance and privacy of certificate based access control, and discusses the limitations in privacy enhancement. The author is responsible for the initial idea behind the paper and most of the analyses.

### Publication VI: "Evaluating Revocation Management in SPKI from a User's Point of View"

This paper looks at the SPKI revocation methods from the user's point of view and gives system designers advice on how to implement usable systems. The author is responsible for the initial idea behind the paper and its technical content.

# Abbreviations

| | |
|---|---|
| ACL | Access Control List |
| AES | Advanced Encryption Standard |
| CA | Certification Authority |
| CRC | Chain Reduction Certificate |
| CRL | Certificate Revocation List |
| DHT | Distributed Hash Table |
| IETF | Internet Engineering Task Force |
| IoT | Internet of Things |
| JXTA | Juxtapose |
| LDAP | Lightweight Directory Access Protocol |
| MANET | Mobile Ad-hoc NETwork |
| OCSP | Online Certificate Status Protocol |
| P2P | Peer-to-Peer |
| PDS | Push-Down System |
| RBAC | Role-based Access Control |
| SPKI | Simple Public Key Infrastructure |
| TLS | Transport Layer Security |
| VANET | Vehicle Ad-hoc NETworks |
| VIP | Very Important Person |
| WPDS | Weighted Push-Down System |

# 1. Introduction

Throughout history, people have sought to protect their valuable resources against unauthorised access. Here, I'll use the VIP (Very Important Person) lounge of a night club as an example: the club owner wants to limit the access to only the VIPs. To this end, there has to be an *Access Control* system that knows which guests are allowed to access the lounge.

There are two main ways of storing the information about the authorised guests: either the resource (the door to the lounge) has at hand a list of authorised users (known as an *Access Control List* or *ACL*) or the authorised user carry a proof of their rights (known as a *Capability*). In our example, these would be a VIP list and a VIP card, respectively.

Both approaches have their advantages. Typically, the ACL is located next to the resource with all the relevant information in one place. So, when the *administrator* (Club owner) wants to create a new right or change an existing one, she merely has to change the list. Furthermore, because the administrator controls the list, it is relatively easy to protect the integrity of the list, i.e. to make sure that no-one else can change the contents of the list and thus create new or extended rights. The downside of this solution is that the administrator cannot make any changes if she cannot access the list. Also, if there are multiple alternative resources (e.g. a chain of Night Clubs with common VIPs), all of these would need access to a central ACL or their own copies of the ACL, which can result in a large amount of unused information being passed around: the information about a particular VIP is only relevant to the Club if that VIP actually attends the club.

The capability-based approach has some inherent advantages in large distributed systems compared to an ACL: we can make access control decisions (grant new VIP cards) without contacting any of the clubs and

we avoid flooding all resources with access control information they might never need. The disadvantage is that we need additional mechanisms to revoke the rights since the administrator no longer controls the card once it has been issued to the user.

Computer systems have adopted both approaches. Traditionally, ACLs have been used more, but the emergence of large, distributed systems has popularised some capability-based solutions. An example of a capability is authorisation certificate, such as the Simple Public Key Infrastructure (SPKI) certificates used in this work [34, 35, 36].

## 1.1 Scope

In this dissertation, I examine the problem of access control in distributed systems and look at how the problem can be overcome with the Simple Public Key Infrastructure (SPKI).

In SPKI, access control rights are expressed as cryptographically signed authorisation certificates. SPKI certificates have been studied since mid 1990's, but standardisation has not finished and the system has been missing several parts to make it a complete solution.

In this work, I identify a number of important missing parts and design solutions for them. I then analyse the resultant system for its applicability, scalability, security and usability. Of particular interest are system performance and privacy. The motivator for the first focus, performance, is the emergence of applications with limited computing and energy resources (e.g. some battery operated Internet of Things (IoT) applications) – access control solutions have to cope with these limitations. The motivators for the privacy focus have been both the increasing management of sensitive data even with small devices (e.g. wearable devices used to collect health related information of the wearer) and the proliferation of monitoring of Internet, both of which emphasise the need for privacy enhancing solutions also in the area of access control.

## 1.2 Contributions

In this work, I claim the following contributions:

**Certificates in general**

- proposing a life cycle model for access control and using it to analyse the SPKI authorisation certificates (Publication II, Publication I)

- extensively surveying certificate based access control research (Publication I)

**Certificate validity und usage quotas**

- developing a classification for certificate validations and using it to dentify the need and designing a protocol for quota-based access control with SPKI (Publication III)
- identifying the need and designing a protocol for validating SPKI certificates (Publication III)
- identifying the need and designing a protocol for managing certificate validity (Publication II)

**Certificate chain reduction**

- identifying the need and designing a protocol for creating and requesting chain reductions (Publication IV, Publication V)
- identifying the hourglass-shape of many certificate graphs and its effect on chain reduction (Publication V)

**Analyses**

- analysing the performance of SPKI, particularly in constrained environments (Publication V)
- analysing privacy in SPKI based access control (Publication V)
- analysing what performance and privacy improvements can be achieved with chain reduction (Publication V)
- analysing, how incorporating the business model and related risks in system design can lead to access control solutions that are technically easier to implement (Publication V)
- analysing the usability issues of using SPKI certificates (Publication VI)

The key finding is that with the proposed extensions, SPKI is a flexible access control solution for many application areas. It is particularly suited for distributed systems with multiple resources that benefit from the ability to operate partially offline.

Resource-wise even small embedded platforms are sufficient for SPKI and it scales well to large systems. Certificate chain reduction can be used to further improve performance in both cases: with small platforms it makes IoT applications easier to implement, and with large systems it enables more consistent performance, particularly for systems, where the hourglass-model for certificate chains applies. IoT applications can also greatly benefit from using delegation to group the devices, thus rendering the management of a large number of devices much simpler.

Privacy-wise SPKI is a good solution and chain reduction can be used to improve privacy further by hiding even more information.

## 1.3 Structure

The rest of the dissertation is organised as follows: Chapter 2 provides a short background for authorisation certificate based access control and SPKI. Then, Chapter 3 summarises the proposed extensions to SPKI. Chapter 4 evaluates the resultant system from performance, privacy, security, applicability, scalability, and usability angles. Chapter 5 then comments on certificate based access control and on utilising additional factors for access control decisions. Finally, Chapter 6 presents my conclusions.

# 2. Authorisation Certificates in Distributed Access Control

In this chapter I illustrate the requirements for a distributed access control solution by using a Public Transit system as an example, introduce SPKI certificates, and show how the example can be solved using SPKI. Finally, I present the previous work in this area. The previous work and example are based on Publication I.

## 2.1 Example: A ticket system for public transport

In all access control solutions, we can identify three main roles: Resource, Administrator and User:

**Resource** is a valuable asset that needs to be protected. However, since an asset is often an inanimate object unable to act as a party in access control, we also often use the term Resource to refer to the access control mechanism representing the resource. This mechanism enforces the rules and grants or denies access.

**Administrator** is the party deciding who will have access to the resource and what they can do with it. This can be the owner of the Resource or a another party authorised to manage the access.

**User** is the party utilising the right to access the resource.

In simple systems, the same party can have multiple roles: for instance, when the resource is the front door to one's home, the administrator and user are usually the same person. But in more complicated systems, such as a public transit service, each role is a separate party – or even divided between multiple parties as shown in Figure 2.1: vehicles are the resources, the passenger is the user and there are several administrative roles (the resource owner and all the intermediate parties). In addition to

these three main roles, there can be additional roles to overcome technical and even theoretical limitations in the system. They can, for instance, provide computational and storage capacity so that the system can use even smaller devices or survive intermittent network outages.

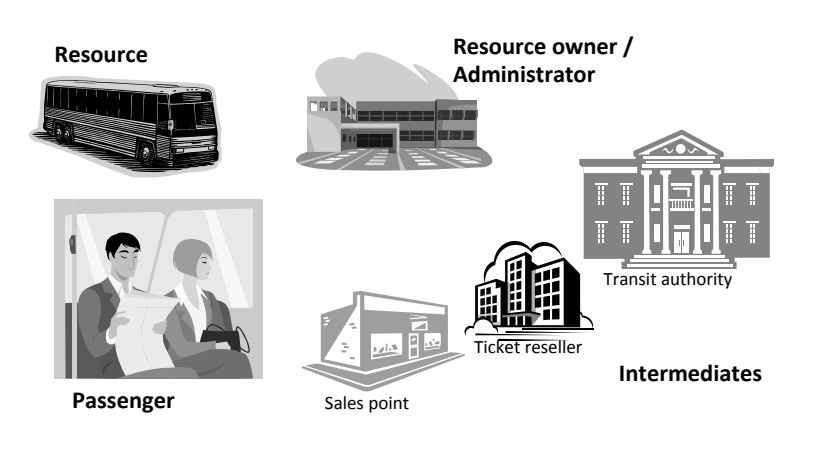


**Figure 2.1.** Parties of a Public Transit service.

The parties of the public transit system have varying requirements for the access control solution:

**Resource / Ticket control**

Ticket control machines represent the resources (i.e. vehicles) and they are used to enforce access rights. In practice, they validate the tickets when the passenger boards the vehicle in question. Each vehicle (i.e. bus, tram, local train) has at least one ticket control machine, but in the underground transit system the machines are at the stations. The machines can be complemented with a secondary control mechanism in the form of ticket inspectors that verify that the passenger is using an appropriate ticket (verifying the right more than just once is an important way to prevent misuse in some application areas).

**Transport providers**

These independent companies provide the actual transporting service and also sell some of the tickets in the vehicles. They require that the tickets are easy to verify so that the ticket system does not unnecessarily slow down the passengers. Also, the transport providers should be able to easily discover people trying to travel with other people's (discount) tickets.

**Transit authority**

The authority is responsible for organising public transport by purchasing transport services from individual bus companies. Their main requirements for the ticket system include flexibility (there can be different kinds of tickets in use at the same time) and preventing people from misusing the system (passenger traveling without a ticket or with a (discounted) ticket they are not entitled to use).

**Ticket resellers**

Resellers may include, for example, news stands and other sales locations, but tickets can also be bought from vending machines or even over the network.

**Passengers**

Passengers' main need is to travel, but they also have other requirements for the system: for instance, they want to know how much of the right they have left or when the ticket is about to expire. Also, if they lose a ticket, they would prefer to get a refund or a replacement ticket. Finally, passengers would prefer the system to have sufficient privacy.

Finally, the access control system has to be able to support different types of tickets. Some can provide unlimited trips within an area for the specified period. Others, on the other hand, can have a limit to how much the passenger can travel: this quota can define, for instance, the total number of times the user can travel (a bus ticket that is valid for 10 trips) or a certain amount of value with which to travel.

The public transit system has been further analysed in my publications, particularly in Publication V.

## 2.2 SPKI Authorisation Certificates

The Internet Engineering Task Force (IETF) has been developing the Simple Public Key Infrastructure (SPKI) since mid-1990's as a more flexible alternative to the X.509 certificates used e.g. for protecting web browsing. SPKI supports certificate-based authorisation but it can also be used to certify an identity. The evolution of SPKI has been detailed in Publication I.

SPKI authorisation certificates [34, 35, 36], like any authorisation certificates, are signed statements of authorisation. The certificate can be abstracted into a signed quintuple `(I,S,D,A,V)`, where

**I** is the Issuer's (signer's) public key.

**S** is the Subject of the certificate, typically a public key.

**D** is a Delegation bit indicating whether the right can be delegated further.

**A** is the Authorisation field, describing what access rights the Issuer delegates to the Subject.

**V** is a Validation field, describing the conditions (such as a time range) under which the certificate can be considered valid.

The meaning of an SPKI authorisation certificate can be stated as follows: Based on the assumption that `I` has control over the rights or other information described in `A`, `I` grants `S` the rights/property `A` whenever `V` is true. Furthermore, if `D` is true, `S` may further delegate `A` or any subset of it. A signature created with the issuer's private key protects the integrity and authenticity of the certificate. The certificates have further fields to assist e.g. locating relevant information and presenting the certificates to human users, but for this dissertation they can be ignored.

Using SPKI certificates, we can implement ticket system for public transit by delegating the right to travel from the Resource (Bus) through the Administrators to the User. This means that the passenger's ticket is a chain of 5 certificates: bus $\rightarrow$ transport provider $\rightarrow$ transit authority $\rightarrow$ ticket reseller $\rightarrow$ sales point $\rightarrow$ passenger. When the passenger uses the right, the chain then forms a loop via a challenge-response process [83], as illustrated in Figure 2.2. An example of an SPKI certificate is shown in Figure 2.3. It is the last certificate of the ticket chain and grants (together with the rest of the chain) the right to travel in zones A and B during September 2014.

As the tickets are digital documents, each party in the system requires a suitable device to handle the tickets. For the sales representatives, these could be small desktop devices, whereas for the passengers they could be their mobile terminals, such as mobile phones or smart cards. The requirements for the terminal are threefold. First, there has to be

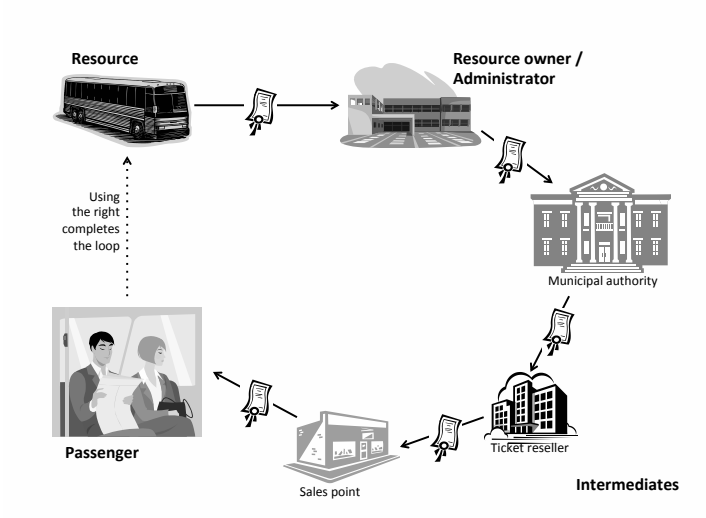


**Figure 2.2.** Chain of trust. The certificates form a chain of trust from the resource to the passenger. This chain of trust then forms a loop when the passenger uses the right.

sufficient space to store the tickets. Secondly, there has to be suitable network connection. As the tickets are only of the order of a kilobyte, neither is a difficult requirement to fulfil, even in a mobile terminal. Finally, the devices require sufficient processing power to manage the required signature operations. Without hardware support for, this can be a harder requirement for the smallest devices.

So far, we have looked at the trust relations relating to the ticket system, but there is also another chain of trust that flows in the opposite direction: the passengers buying tickets trust that eventually they will be able to travel with these tickets. In this case, the passenger mostly trusts the brand of the service, guaranteed in this case by the transit authority – who is likely the primus motor for the service. Hence, the trust chain does not always reveal the order in which the trust relations were built.

## 2.3 Previous Work

In this section, I summarise Publication I that presents a thorough summary of the research activity on certificate based access control and on SPKI certificates during the past 15 years.

The main research topics are shown in Table 2.1. They have been

```
(11:certificate
  (6:issuer
    (10:public-key (23:ecdsa-secp256k1-sha-256)
      (#032db1ae60c8953545dcf404160c82723ce94b
        121d3c377e64d0abea87766437d9#))
  )
  (7:subject
    (10:public-key (23:ecdsa-secp256k1-sha-256)
      (#03531e6c2be3ea0ff135b38ba9fcd0616d8aff
        ce93be131bda568b1d18e286b95d#))
  )
  (3:tag
    (16:public_transport (6:ticket (8:provider(3:TA))
        (5:zones (1:A) (1:B) )))
  )
  (5:valid
    (10:not-before (19:2014-09-01_00:00:00))
    (9:not-after (19:2014-09-30_23:59:59))
  )
)
(9:signature
  (4:hash (7:SHA-256)
    (#3f2b181808187d92ca4d41d98f0c2035d389ae
      867e0e5209de0f3f1cf3b9d1e0#)
  )
  (10:public-key (23:ecdsa-secp256k1-sha-256)
    (#032db1ae60c8953545dcf404160c82723ce94b
      121d3c377e64d0abea87766437d9#)
  )
  (3:sig
    (#f25af41c924fc6d2927bdcff40febf90b1b022cedfbf
      029ce4e417310e773a0ade128863ca74db3ff9908fe8
      9a9dca5bb0791191926f97e9214169ab21206ba7#)
  )
)
```

**Figure 2.3.** A SPKI certificate representing part of a ticket in a public transit system.

grouped using the life-cycle model from Publication II (introduced in more detail in Section 4.1).

**Table 2.1.** Major research areas in certificate based access control during the past 15 years

| Phase 0 | Phase 1 |
|---|---|
| Federated access control<br>Trust management<br>Role-based access control | Policy misconfiguration detection<br>Manageable policies |
| **Phase 2** | **Phase 3/4** |
| Chain discovery<br>Certificate storage<br>Chain evaluation<br>Alternative rights<br>Chain reduction<br>Limited rights | Revocation<br>CRL optimisation<br>Validity statements<br>Short-lived certificates<br>Revocation incentives<br>MANET/VANET environments |

### 2.3.1 Phase 0: Making the decision

The goal of this phase is to have the identity of the person/device/etc. receiving the rights and to know the rights to be granted. A key research topic in this phase is *federated access control*, which aims at crossing organisational boundaries by establishing the ways in which different organisations can trust each other and how users from one organisation can use the resources of another organisation. In this case, we have to solve problems such as how organisations with different trust management systems can establish trust relations [12] and how we can develop suitable identities by, for example, using distributed hash tables [84]. Federated access control is particularly relevant to applications such as Grid [9].

But what if there is no organisational connection between the subject and resource – is there a way to be sufficiently confident when making the decision about whether or not to grant the right? This area is covered by *trust management*, which considers trust relationships between entities and how this trust can be expressed in statements such as certificates. This is the idea behind PolicyMaker [17] and SPKI [35], among others. Trust management also takes into account just what can be deduced from

the often incomplete and contradictory evidence of trust statements (this approach is also popular in areas such as reputation management). Here, we look at issues such as how trust management systems compare to establishing trust in the real world [94], how transitive can trust actually be[57] and how we can automatically negotiate trust relationships [69]. Chapin et al.[24] have provided a thorough review of trust management systems and their properties, while Viriyasitavat and Martin[103] similarly provided a thorough survey of trust management activities in recent years in view of networked services. As an application area, MANETs (mobile ad-hoc networks) have received a great deal of attention as has trust management in MANET environments [44, 26].

A related issue is the problem of managing a large number of rights and users: if each subject has an individual set of rights, understanding and managing the system becomes quite complicated. These problems can be mitigated by grouping the subjects suitably and granting rights to those groups. *Role-based access control (RBAC)*, where users are grouped based on their 'roles' in the organisation, has received the most attention. In social media, relationship-based access control (e.g. friend of a friend) has recently been studied [39]. The large number of proposed access control models has prompted Barker to propose creating a meta model in place of creating a new model for each application area[8]. Finally, Patil et al. looked at interoperability issues between different access control models[86].

### 2.3.2 Phase 1: Expressing the decision

The goal of this phase is to express the decision made in Phase 0 in a machine-understandable form. This is not always trivial and the resulting security policy misconfigurations can be difficult to detect manually, which has prompted research on how to automatically detect these misconfigurations [11, 5, 55, 96, 65].

Another approach to this problem is to make the policies more manageable to humans by limiting the number of configuration options [78]. Solutions developed for organisations with many administrators can benefit from design guidelines so as to prevent errors [10]. With certificate-based solutions, this phase includes things such as identifying the servers to manage the revocation and validity status of the right, which can also present usability problems for the user.

### 2.3.3 Phase 2: Enforcing the decisions

Whenever the right is being used, the enforcing party has to make sure that the correct rights have been granted to the party trying to use the right and that the right is still valid. In addition, there are application-dependent performance requirements: for instance, it should be possible to verify a passenger's ticket within one second. If all the required information is available locally, performance requirements tend to be easier to meet. However, with distributed solutions, the fact that we sometimes have to obtain information also from other sources can present challenges.

The key problems in this phase include:

- discovering the correct chain from the local/provided certificates
- (if all the required certificates are not available locally) finding and obtaining the missing certificates
- evaluating the chain to see whether it grants the requested action

Because all of these problems require the construction of a certificate chain, they are usually treated together. Clarke et al. first proposed an algorithm for finding and evaluating a chain[27]. Li et al. modelled the certificates as searchable credential graphs, which makes an alternative search algorithm possible[71]. Schwoon et al. used weighted pushdown systems as bases for their algorithm[93]. Hristova et al. used Datalog rules to generate algorithms for evaluating SPKI rules[51].

A more complete approach to discovering the certificates is to combine the storage and discovery solutions. ConChord is a storage and name resolution solution that utilises distributed hash tables as a basis for storing and locating certificates [4]. Hui et al. improved on this solution in trusted peer-to-peer (P2P) networks by creating a more efficient solution that relies on nodes trusting some other nodes and forming co-operating group points[52]. Finally, Ying and Jiang proposed another distributed hash table (DHT) -based solution for storing and discovering certificates, but they also used Hilbert space-filling curves to optimise the implementation[112].

A big question is, which party is responsible for finding and acquiring the evidence. If it is the client, the solutions scale better [91, 90]. An alternative approach, one that is particularly interesting for thin clients, is to have separate services for finding the chain [107].

Borders et al. implemented an alternative policy evaluation solution

using extensive caching for the results and achieved significantly better performance than, for instance, KeyNote has[19]. Guan et al. also developed an alternative trust management solution, DTMAN, and an efficient algorithm for compliance proofing that they claim is more efficient than KeyNote or SPKI[45].

Another problem appears if the user has multiple alternative rights: which of the alternative rights should be used and who gets to decide this? One solution is to extend SPKI to allow the issuer to prioritise the certificates – this then affects the order in which the certificates are used [100]. However, if the rights consist of, e.g. credit cards, the users of the credit cards would presumably like to decide for themselves, which card they want to use (as opposed to the issuing organisations deciding on their behalf). Further research in this area is therefore warranted.

If a right is somehow limited, there needs to be an additional mechanism for keeping track of the usage, because the certificates alone cannot keep track of how much they have been used [62, 49].

Finally, even though chains are a necessary by-product of the delegation mechanism, repeatedly using the same chain can be inefficient compared to reducing the chain, because we have to discover and evaluate the whole chain every time the right is used [59, 101].

### 2.3.4 Phase 3: Changing or revoking the decision and Phase 4: The right expires

Revocation is a particularly relevant problem with certificates, because once a certificate has been issued to a user, there is no way of making the user give up all their copies - a separate mechanism is required to change or revoke a certificate. Thus, we would like a solution that

- efficiently communicates only the minimum amount of information required
- spreads the information with only a minimum amount of delay

Unfortunately, these two properties tend to conflict: a low delay solution requires more frequent communication, so we have different solutions depending on the application priorities.

The traditional solution has been Certificate Revocation Lists (CRL), which identify the revoked certificates and which are published at certain intervals. The periodic nature of this process means that there is always some delay before the revocation information is spread. Also,

the amount of information that has changed between each version of the list is usually quite small, so often much of the bandwidth is wasted by repeating previously published information. Thus, much effort has been put into reducing the amount of information in each CRL update by only communicating the changes from the previous CRL. These approaches include only publishing the changes to some previous CRLs (known as a delta CRL) or using more sophisticated data structures, such as B-trees, to further reduce the amount of information communicated.

Cooper proposed segmenting large CRLs into smaller ones, thus (hopefully) reducing the amount of information downloaded by individual verifiers[28]. He also proposed spreading the requests for CRLs more evenly by issuing CRLs more often than just when they expire (e.g. every 6 hours for CRLs that expire after 24 hours). Later, he proposed extending the latter approach to delta CRLs by having each delta CRL be valid for the same time period [29]. Zhang and Wang implemented these approaches in a solution that stored the CRLs in an lightweight directory access protocol (LDAP) server organised as a binary tree[114].

Naor and Nissim proposed reducing CRL traffic by organising the revoked certificates as a hash tree (as opposed to a traditional list)[79]. Li et al. then improved on this by first grouping the certificates using one-way accumulators and only storing the groups in the hash tree, which further reduced the amount of traffic[66].

CRLs are suitable for solutions where there are a limited number of certificate issuers that can each maintain their own revocation lists. Typical examples include Certification Authorities (CA) issuing identity certificates for a large number of users. If, however, there is a very large number of parties issuing certificates (as with non-hierarchical PKIs, such as SPKI), other solutions can be better suited.

Hash chains have been used to simplify the required computations as compared to the signatures required for publishing, e.g. CRLs. One solution is to periodically publish hash chain values to signify continued certificate validity [67]; another is to use a hash chain to bind a group of certificates together so that they can be revoked with a single operation [99].

An alternative approach to issuing CRLs for revoked certificates is to issue validity statements for certificates: this means that a certificate would only be valid with a current validity statement. In some application areas, this approach works better than CRL, but it requires more

resources to generate and transmit all the validity statements. To improve efficiency, technologies such as Merkle hash trees [13, 37] and square residues [72] have been proposed as optimisations.

For some applications, the revocation information has to be up to date, so the information has to be checked whenever the right is used. The standard solution is to use the OCSP protocol [41], but an alternative based on Web Distributed Authoring and Versioning (WebDAV) has been proposed [22].

Yet another approach to revocation is to use short-lived certificates that do not require revocation in the first place [92, 40].

When the answer to a validation query cannot be precomputed, there is always the risk of key exposure. [14] proposed using forward secure signatures to manage the risk.

Some solutions reverse the idea of checking certificate validity by pushing certificate status information to interested parties [111] or use a P2P network to spread the revocation information [113, 85, 63].

Previous studies have compared the different revocation methods [115, 79, 109], and researchers have developed platforms for performing these comparisons [77, 76].

In volatile wireless networks, motivating the parties to co-operate with respect to revocation requires suitable incentives [15].

The limited resources of mobile nodes in MANETs require special care when implementing the revocation method, as the environment requires solutions that are scalable to a large number of nodes, have low performance and bandwidth requirements, and are robust in a dynamic network environment [54, 73, 33, 30]. As a special case of MANETs, VANETs (Vehicle ad-hoc networks), characterised by a large number of fast-moving nodes, present even more stringent requirements for revocation [47, 46, 89, 108].

Most of the research on revocation has focused on identity certificates; authorisation certificates such as SPKI have slightly different requirements, as there are typically a much larger number of certificate (and, thus, revocation) issuers and chains of certificates that have to be efficiently evaluated [87, 60].

### 2.3.5 SPKI

SPKI has been analysed using many different formalisms. One of the earliest was by Abadi et al., who presented a logical language for

access control lists in distributed systems as well as theories for deciding whether access should be granted[2]. Later, Abadi analysed the linked local name spaces concept that enables SDSI to function without globally unique names[1]. Halpern and van der Meyden improved on this by making the logic more closely match the SDSI name space concept [**?** ], and, later, showed that the approach can be extended to most features of SPKI [48]. Howell and Kotz extended the work done by Abadi [1] to support restricted delegation [50]. Li et al. also built a logic-based language for specifying delegation logic [70]. Aura formalised the concept of delegating access rights using a graph theoretic approach [7]. Delegation has since received attention in the research community both from applicability and threat standpoints [74, 38, 23].

Appel and Fleten showed that distributed authentication frameworks can be analysed using higher order logic and showed that, even though the high-order logic is undecidable, it can be used for practical purposes by requiring the client to construct the proof [6]. Schwoon et al. proposed a generalised authorisation problem, presented algorithms for solving it and utilised it for privacy, recency, validity and trust [93]. First-order logic has also been used to analyse SPKI [68, 43]; in particular, Li and Mitchell [68] showed that SPKI certificate theory in RFC 2693 [35] is semantically incomplete. Ermani and Sistla proposed a first-order temporal logic for defining and analysing distributed access control policy [32]. Chetcuti and Massacci extended the logic-based analysis to include time constraints [25].

An algebraic approach by Bonatti et al. extended formal analysis to cases where there are multiple parties creating and enforcing access control policies, possibly with many different policy languages [18]. Biskup and Wortmann extended the approach further by making it possible to compose access control policies into compounds [16].

Jha and Reps demonstrated that a set of SDSI/SPKI certificates creates a push-down system (PDS) that can be used to efficiently solve many certificate-based problems [56]. Ganesh and Gopinath used weak Monadic Second Order (WS1S) logic and a Weighted Push-down System (WPDS) to analyse SPKI and to formally answer authorisation questions, particularly with respect to validity conditions [42].

SPKI has also been analysed from a trust management perspective to see how the various trust metrics can be applied[110].

Lately, research activity has focused on using SPKI for access control

in different applications: Saito et al. used SPKI to implement privacy supporting access control for the web [88], Saito et al. [88] and Nazareth and Smith [81] used SPKI to improve privacy in Shibboleth. Certificate creation and revocation have been automated using suitable policies [101, 21] . K-SPKI, a Kerberos/SPKI hybrid, used Kerberos for authenticating the user in order to reduce the number of key pairs used in the system [104, 102],

Due to its distributed nature, SPKI is naturally suited for securing P2P [58, 75] and JXTA [105] applications. SPKI has also been applied to ubicomp applications [98] and for securing Jini [64]. Another distributed application type that SPKI works well with is agent systems [106, 82, 80]. Finally, SPKI has also been applied to secure semantic web [3], home appliances [95] and grid applications [53, 31].

## 2.4 Chapter Summary

In this chapter we have seen that authorisation certificate based access control can be particularly well suited for distributed application and the main areas of research activity in this field during the past 15 years.

In the next chapter I will present the functionalities I identified should be included in SPKI to make it a well rounded solution and the solutions I designed for these functionalities.

# 3. Extending SPKI

In this chapter, I introduce the extensions to SPKI proposed in the publications of this dissertation. The extensions include two new validity methods that enable new usage models (Publication II, Publication III), as well as protocols for using the certificates and validating online conditions (Publication III), a protocol for managing online conditions (Publication II), and a protocol for requesting chain reductions (Publication IV, Publication V). In the publications, I have analysed the effect these changes have on performance, privacy and other aspects of the access control system – these will be discussed further in Chapter 4.

## 3.1 New online validations

To better understand the need for the new online validations *Renew* and *Limit* introduced in Publication III, I first classify use cases according to the types of validations they have as shown in Table 3.1 (adapted from Publication VI). The classification goes from simplest (A) to most sophisticated (E), so a more sophisticated level is always capable of implementing the same functionality (and more) as a lower level, but at a higher implementation cost.

**Type A: Implicit Trust** means unlimited trust: the right will not expire and there are no usage limits; no external information is required to evaluate validity. Typical examples include a resource trusting its owner, e.g. a key to one's home.

**Type B: Expiring Right** suits a situation where the right can be used without limit until it expires and the right is not valuable enough to require revocation. The only external information required is current time. An example would be a single use bus ticket valid

for one hour.

**Type C: Revocable Right** comes into play when the right is so valuable that revocation capability is required though revocation does not have to take effect immediately. An example is a bus ticket valid for a whole year — if it is lost, the subject would want it revoked and a replacement issued. In this case, the capability to revoke the certificate within some reasonable timeframe, e.g. after at most two weeks, would suffice. Here, the resource needs at least periodic access to outside information about the validity of the certificate.

**Type D: Context Dependent** applies to situations where the context of usage determines whether the right can be used or where revocation has to take place immediately. One example might be a space in the bus reserved for a wheelchair or baby strollers available for discount with a special ticket. Here, the ticket machine needs access to external information about whether the space is still available before charging the ticket and letting the passenger pay.

**Type E: History Dependent** applies to situations where the right depends on usage history. An example is a credit card with a monthly limit or a bus ticket for 10 trips.

In SPKI, in addition to the validity period, which enables types A and B, there are three online validity checks: CRLs (Certificate Revocation Lists) and revalidation (also known as reval), which enable type C, and one-time checks, which enable type D. However, SPKI offers no support for type E and applications requiring such functionality.

In Publication III the authors propose two new online validations to enable new applications: `Renew` and `Limit`.

**Table 3.1.** Classification of validities (adapted from [61])

| Type | Name | In SPKI |
|---|---|---|
| A | Implicit Trust | Field left empty |
| B | Expiring Right | A validity period |
| C | Revocable Right | CRL, Reval, *Renew* |
| D | Context Dependent | One-time |
| E | History Dependent | *Limit* |

`Renew` is an alternative type C method. Instead of issuing certificates with long validity periods and then requiring online validations to manage their validity, we issue a sequence of shorter-lived certificates, which together cover the lifetime of the certificate with a long validity period. As each certificate is valid only for the required revocation period (e.g. two weeks for a transit ticket), validating the right is simpler for the resource and the resource can operate offline. Also, fetching subsequent certificates is not limited to the time of using the right (e.g. morning rush hour) but can be made proactively (e.g. already during the previous night), the load on the server can be distributed more evenly over time.

`Limit`, on the other hand, is a type E method that enables a large number of new applications. Limit operates on the concept of first reserving the required amount of right (to ensure that all limit conditions in a certificate chain are valid) and then committing the reservation. The mechanism is implemented using a network server to keep track of the quota. Each issuer is free to choose the server, though in many cases it is probable that the system designer will provide suitable servers as part of the service.

### 3.2 Protocol for Requesting Service

While the SPKI specification [36] proposes formats for the certificates, it defines no protocol for the user to request service from a resource. In Publication III, the authors define a protocol for this purpose.

For the system to implement quota and non-repudiation correctly, the protocol was designed so that after establishing a connection to the resource, the user sends a *service request* that both specifies the exact type and amount of service requested (e.g. how much the user wants to charge from the credit card) and provides the necessary certificates as a proof of right to that service. The resource then replies with a message whether it agrees to provide the service. If, for instance, the user does not have the necessary right, the service will be denied and a negative reply with a corresponding reason code sent to the user.

The service request and the reply are signed, so they authenticate the parties, guarantee the integrity of the transaction even if no cryptographically protected communication channel is available, and provide non-repudiation (i.e. neither party can retroactively deny that the client requested service and the resource agreed or refused to provide it).

With this protocol, there is now a common way for requesting service and ensuring an audit trail.

### 3.3 Protocol for Validating Online Conditions

The SPKI specification [36] only partially defines the protocol for validating online conditions. In Publication III the authors have completed the specification.

With the amended protocol, a resource can validate all online conditions. C and D type conditions are straightforward to validate, but the E-type `Limit` condition requires special care as a successful validation also consumes a corresponding amount of the user's quota. Here, the signed service request acts as proof that the user wishes to consume the quota and prevents anyone without a valid service request from validating the conditions. The service is addressed to the specific resource and can be used only once, thus preventing a dishonest resource from repeatedly using the service request to, for instance, charge the user's credit card.

Another issue is that there can be more than one `Limit` condition in a chain being verified. To make sure that all of them either succeed or fail, the protocol operates in two phases: the first round makes sure all `Limit` conditions can succeed by reserving the required quota, and the second round then commits the reservations only if all of them succeeded.

### 3.4 Protocol for Managing Online conditions

For an online server to monitor a quota or otherwise comment on a certificate's validity, the server has to be given information about the initial status and then later, should the status change, be updated about the new status. The initial status can either be included in the certificate itself or communicated separately to the server. Any status changes, however, will always have to be separately communicated and for this, Publication II defines another missing piece in the SPKI system: a way to manage the different validity conditions. The protocol enables the administrator to:

- delete existing rules used to define the status of particular validity conditions,
- define new rules for the status, and, finally,

- inquire the status of a condition (e.g. how much of a limit has been used).

To be able to issue these commands, the administrator has to prove the right to manage the condition (related to a specific certificate) by proving they are the original issuer of the certificate or by proving they have been authorised by the original issuer.

### 3.5 Chain Reduction

As we learned with the public transit example in Section 2.1, using SPKI authorisation certificates for access control results in chains of certificates. In the example, a ticket consists of a chain of five certificates as shown in Figure 3.1. The ticket reader already has the first certificate (since it created that certificate), but the passenger will present the remaining 4 certificates every time she climbs on the bus. And even if the chain stays the same for several trips, the ticket reader has to re-evaluate the whole chain every time, which can be both slow and resource intensive.

A solution proposed in the SPKI specification [36] is the Chain Reduction Certificate (CRC), where a chain of certificates is replaced with a single certificate granting the same rights, thus reducing the number of certificates to evaluate and potentially hiding some information to improve user's privacy. However, the SPKI specification provides no guidance for implementing this idea. In Publication IV and Publication V we delve into the details of using CRCs and analyse the effect they have on performance and privacy.

For our analysis, we use, again, the public transit use case. In that example, any consecutive certificates could be replaced with a CRC, but as the Transit Authority is an organising party for the service and every trust chain goes through it, a natural solution is to replace the chain with two CRCs: one from the vehicle to Transit Authority (issued by the vehicle) and another from the Transit Authority to the passenger (issued by the Transit Authority), as shown in Figure 3.1. Again, the bus would have the first certificate, so the passenger would present a single certificate for the ticket control machine to evaluate, thus significantly improving performance.

For a CRC to be valid, it has to be issued by a party that has the right

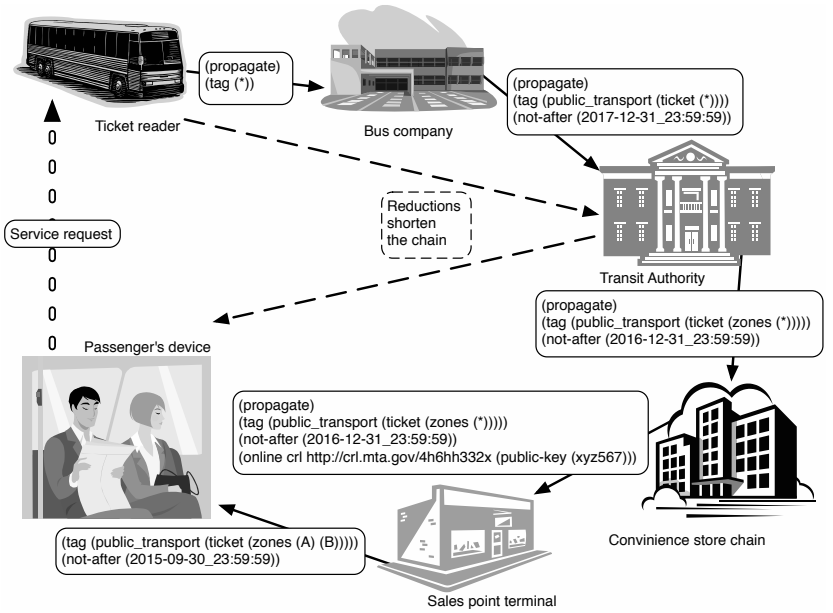


**Figure 3.1.** A chain of certificates representing a ticket in a Public Transit system.

to issue such certificates in the first place. Therefore, all parties in the chain are only able to issue CRCs for certificates that are downstream from themselves. The first party in a chain (the resource) is, therefore, the only party that can reduce the whole chain. Similarly, the Transit Authority is the only party that can issue CRCs starting from the Transit Authority and extending to any downstream party up to the Passenger. Finally, any party could also outsource its reduction duties to a third party by authorising the third party to issue such certificates.

We also note in the publication that since the bus is the party creating the first CRC and also the only party using that CRC, there is no need for the bus to sign that certificate as long as it stays in the trusted memory of the ticket control machine. We call these *internal reductions*, as they can only be used internally by the creator as an unsigned certificate is of no use to other parties. Avoiding the signature, however, makes creating internal reductions even cheaper (for the bus, they are essentially free as they are created as a by-product of normal chain evaluation).

Any party is always free to issue reductions on their own volition, but the SPKI specification provides no standard way for a downstream party to request reductions, so our Publication V proposes one. The key elements of the protocol are that the user provides the chain to be reduced and defines, if the reduction should contain all or just some of the rights, and whether it should be valid for all or only a part of the time period the

original chain would allow. The user could, for instance, request a single-use credit card with a much lower credit limit and an ephemeral identity to avoid revealing too much information about themselves. Also, if this one-use credit card is given to someone else, the lower limit prevents them spending more than intended.

Creating reductions behaves similarly to issuing alternative certificates to the same right - they do not replace the original chain and the user is free to use either the original chain or the CRC. And if there is a limit in the original chain, the same limit will be contained in the CRC, so using either will consume the limit.

## 3.6 Chapter Summary

I this chapter I have presented the extensions to SPKI I have proposed in my publications. In the next chapter I will analyse the resultant system from different angles.

# 4. Analysis

In this chapter, I summarise the access control life cycle model I proposed in Publication II and evaluate the extended SPKI system from performance, privacy, security, applicability, scalability, and usability angles. The analysis draws mostly from Publication VI and Publication V, but also from all the other publications.

## 4.1 Phases of Access Control

In Publication II I point out that the process of access control consists of several distinct phases. In Publication I I then use this model to analyse the different certificate technologies and their roles in access control.

In the model (depicted in Figure 4.1), each phase has particular goals, but the means of reaching them vary depending on the technology used to implement them:

In **Phase 0**, someone administering a resource makes the decision to grant the user the right to utilise the resource. This decision could be based on the administrator knowing the user (a friend), the user holding some position in an organisation or the user being a paying customer of the administrator's service. In our public transit example, the administrator might be the ticket reseller and the user a passenger

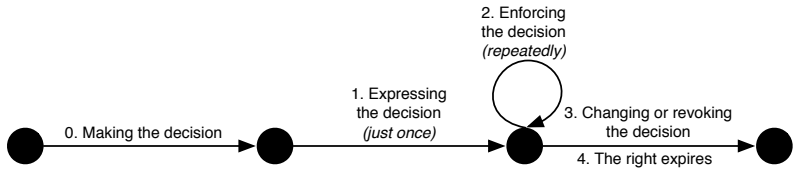


**Figure 4.1.** Phases of Access Control. A simplified illustration of the types of activities related to each phase of the access control process.

who wants a ticket and has initiated the phase by entering the sales office. So, Phase 0 can be initiated by either the administrator or the user.

In **Phase 1**, the administrator must somehow express the decision in a form that can later be used to automatically verify the user's right to use the resource. For instance, the passenger can be issued a ticket, which she then shows whenever she wants to travel. To issue a ticket, the administrator requires the user's identity (with SPKI this is usually an (anonymous) public key, but with other technologies it could be e.g. a user name) to which the right is granted. Though theoretically easy to obtain, in most practical systems the creation, management and transfer of these identifiers can present real challenges, particularly from a usability point of view. The chosen method can vary depending on, for example, whether the administrator and user meet face to face or whether the right is issued over network or even without the user's participation by e.g. looking up the identity from a directory.

In **Phase 2**, whenever the user tries to use the resource, the resource ensures that the right is still valid. In our example, this would be the ticket control machine inspecting the passengers' tickets when they step into the bus. This validation process entails checking the subject's right to the operation in question, i.e. checking that the ticket is valid for the intended trip as opposed to, for instance, for a trip in another zone. The process also verifies that the subject is the correct user of the right, i.e. that the passenger's application has the correct identity for which the ticket has been granted. With current technology, however, we usually cannot easily verify that the device is in the right hands and has not been lost/stolen or lent to someone else (e.g. someone entitled to buy a discount ticket could lend their mobile device containing the ticket to another user without such entitlement).

**Phase 3** and **Phase 4** complete the model. Should the passenger lose the ticket, the administrator might be able to revoke the ticket (**Phase 3**) and issue a new replacement. Finally, even if the certificate does not wind up being revoked, it will eventually expire when the subject exhausts the right or its validity period simply runs out (**Phase 4**).

Compared to Phase 1, which only takes place once, Phase 2 can be repeated numerous times. Therefore, it usually makes sense to design the access control solution so that Phase 2 is as simple to perform as possible, even at the expense of Phase 1. The Renew method aims to simplify Phase 2 activity compared to existing solutions (CRL and reval).

## 4.2 Performance

We can formulate the key performance question by asking what kind of platform is required so that the access control solution provides sufficiently fast operation for the application. By using a large and expensive platform we can achieve very good performance, but this would significantly limit the applicability of the technology. Ideally, we would prefer the solution to run very fast on a cheap platform utilising only the minimum of energy (this is particularly relevant to battery operated IoT applications).

In Publication V we report on the results of our research prototype for an access control solution for a parking lot. Here, the parking lot is guarded by a system running on a modern high-end embedded platform and the drivers use their mobile phones to run the client application. As a mobile phone is at least an order of magnitude faster than the fastest embedded platform we used, we concentrate on analysing only the embedded platform.

The parking use case ended up having certificate chains of equal length and structure as the public transit example and had the same hourglass-shaped graph, so we used the same reduction structure resulting in five certificates in the original chain and two certificates in the reduced chain. In both cases, the gate had the first certificate of the chain. This means that we can discuss the parking example and transit example interchangeably in our performance analysis.

Our embedded platform did not have hardware support for cryptography, so all operations had to be implemented in software. This meant that the system ended up spending over 99.7% of the computation time doing cryptographic calculations. As hardware support can make cryptography both much faster and energy efficient we conclude that for devices with either strict time demands or limited energy budget (e.g. IoT), hardware support of signature operations would be highly beneficial. Some constrained devices already have such support for cryptographic operations such as hash functions and AES encryption, but the support for signature operations and ECDSA, in particular, is not so prevalent.

As this application serves human users waiting at the gate for the system to make an access control decision, we prefer the whole access control transaction take less than 1 s. In many machine-to-machine applications (e.g. IoT applications) the time demands can be much more

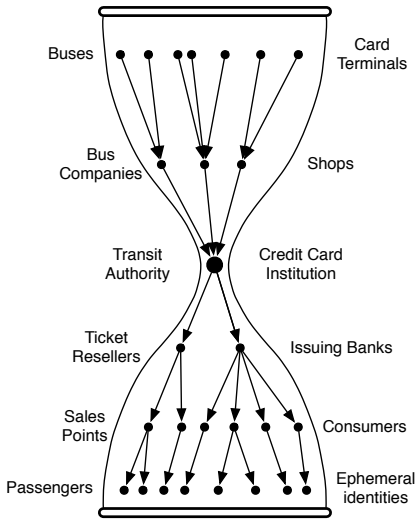


**Figure 4.2.** Our use cases form an hourglass-shaped graph with a supernode. Such a graph makes it easy to reduce a chain of any length to just two certificates.

relaxed. Our platform yielded transaction times of 1.3s and 0.8s for the original and reduced chains, respectively. Hence, an embedded platform is capable of achieving good performance with realistic chain lengths.

Also, using reduction improves performance and reduces energy consumption. The first reduction is essentially free for the gate/bus, as it has to evaluate the chain, it can use an internal reduction thus avoiding expensive signature creation. The second reduction, however, requires a separate reduction service run by the parking service/transit authority. These reductions are created by a dedicated server with ample resources and no strict time limits (the client application will request this reduction automatically well before it is required to be available, so the driver does not have to wait for this operation to complete). Effectively, we transfer some of the verification activity from the embedded system to the server.

Another way for the gate/bus to save processing is by selectively caching certificates already verified, thus avoiding repeated verifications. As it suffices to store the hash of each verified certificate to avoid the the signature re-verification, this consumes minimal memory and is useful even if only a small number of the certificates end up being reverified.

Our use case had no online validations as they were not required. However, performance can be affected by online validation checks. Validating online checks requires network communication and cryptographic computation for validating the replies and creating some of the validation requests, which can be a problem for energy or computation limited

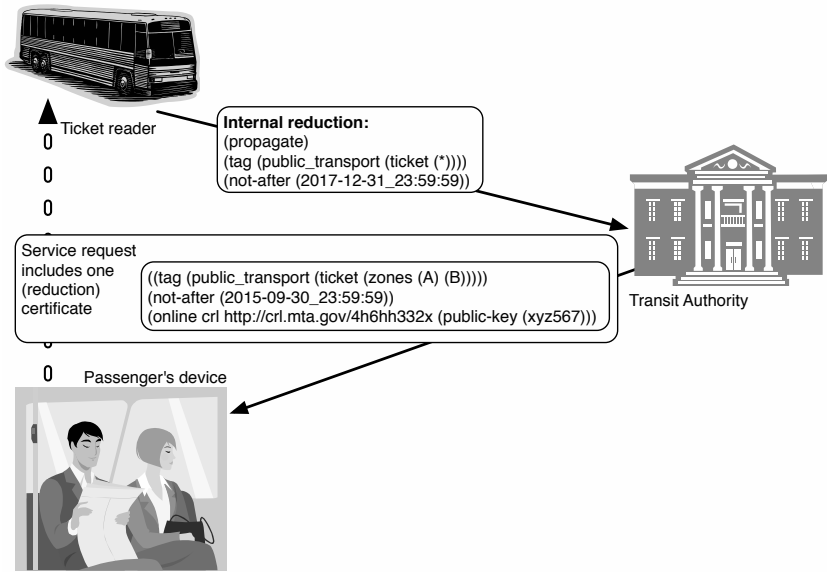


**Figure 4.3.** Hourglass-shape with internal reduction creates consistent performance regardless of the length of the original chain.

platforms or for platforms with slow network connection. If the parties creating the certificates use online checks liberally, they may degrade the performance below acceptable levels – and reductions are of no help with online checks. Faster platforms with good network connection are not similarly affected, but with small platforms, the system designer should take care whether online validations are required and which method is sufficient to implement the required functionality (the more thorough D and E type methods also require more frequent online validations thus reducing performance).

Based on our prototype we conclude that performance-wise SPKI is a suitable solution for many IoT-applications on modern embedded platforms, but energy limited environments, such as sensors, may require hardware cryptography to keep the energy consumption low. Performance wise, reduction is a good tool for improving performance and can be used to transfer activity from resource limited devices to less constrained devices.

Finally, we hypothesise in Publication V that a significant number of use cases are likely to exhibit the same hourglass-shaped trust network as the credit card and public transit examples: there is a single organising party in the middle of the network through which every trust chain travels and that party has the motivation to offer reduction service for the users (e.g. a credit company or a transit authority) as shown in Figure 4.2. Such

a network structure would make using CRCs more straightforward and offer a consistent level of performance for the users: regardless of the chain length, the end user would only have to present a single certificate to be evaluated, as shown in Figure 4.3. The downside of this model is that the reduction service is well informed of every user's rights, which can be a privacy problem in some applications.

## 4.3 Privacy

A privacy enhancing access control solution does not require the user to reveal any information about her identity, rights, or how she uses them other than what is necessary for the proper operation of the system.

SPKI certificates support anonymous identities, so from the start they have been promoted by their designers as a privacy enhancing solution. Chain reduction has also been promoted to provide additional privacy protection by hiding some information the original chain would have revealed. However, what information is actually hidden and what remains has not been analysed in detail.

In Publication V we develop a model for classifying the parties within and outside the system, and use the model to analyse what information is passed between parties both with and without chain reduction.

We find that by design, the chain contains a fair amount of information about the rights and from where they came. Some of the fields can be hidden using encryption or by not including it in the certificate in the first place. Both methods, however, require care from the system designer.

We agree that the anonymous identities of public keys do help improve the privacy of users, but repeatedly using the same identity will still reveal more about the user. To further improve privacy, a user should utilise ephemeral identities, i.e. regularly switching the identity they use for a particular application to prevent comprehensive monitoring. A way of simplifying this is by requesting reductions to a new, ephemeral identity at suitable intervals.

Chain reduction helps in hiding some information, e.g. details about where the right came. Also, if the reduction is issued to a more limited right, it also hides the original right. A typical usage scenario would be a credit card owner wishing to keep the exact credit limit private by using a single use credit card with just the required amount of credit to pay for a purchase, thus improving the buyer's privacy from the shop.

The tradeoff of using reduction to improve privacy is that the reducer has to be given the whole original chain – i.e. more information than they would otherwise have. In our example, this would mean that the passenger has improved privacy from the busses, but the transit authority would be even better informed on how individual passengers travel.

The online methods also leak information about when (and with the limit method, how much of) a right is being used. Again, the more demanding methods (one-time, limit) with more frequent online checks provide a more accurate picture of usage – so improving privacy is another reason (in addition to performance) to be careful about using the more demanding online methods. As stated in the performance section, reductions in most cases cannot help with online validations.

We conclude that anonymous identities and CRC can be used to improve privacy from the resource and outside parties, while preserving non-repudiation. However, unlike previously suggested in the literature, improving privacy with reduction is limited in its effectiveness and cannot ensure privacy for the user. Finally, the choice of the reducer affects what is hidden and how expensive a reduction is to create: using the neck of the hourglass as a reducer makes this party quite knowledgable of the activities in the system; a trusted third party as a reducer would improve privacy.

### 4.4 Security

The security of SPKI certificate based access control has been extensively analysed using different formalisms (see Publication I for further details), so here I concentrate on analysing the changes to the security model the proposed extensions introduce.

The new validity conditions Renew and Limit do not significantly change the situation: they follow the model of existing online validations. Online validations do not normally create new vulnerabilities to gain additional rights, as the rights are defined in the certificate itself and the online validation only limits *when* or *how much of this right* can be used. Technically it is of course possible to issue certificates with limitless rights and then use online validations to monitor what the user does, but this largely defeats the purpose of certificate based access control and is not advised.

The major weakness of online validations is their reliance on a network

connection being available – and, thus, susceptible to network outages. Here, the new `limit` condition, which requires more online activity as other conditions due to the need to first reserve and then commit the quota, is, correspondingly, more susceptible. The susceptibility has two main variants. First, if the certificate issuer is unable to manage the online conditions, the system operates on old data, which could mean the user does not yet have access to her new rights or is able to continue to use her revoked rights. Second, if the resource is unable to contact the server, it either has to deny service or make an access control decision based on incomplete data – I discuss this issue in more detail in Chapter 5.

As chain reduction is based on the same evaluation as the certificates otherwise use, it does not change anything in the security model. By reducing the chain length, it reduces the potential for Denial-of-Service-attacks flooding verifiers with long chains of certificates. At the same time, the reducers are a new party now vulnerable to attack. Particularly if reductions are performed by Trusted Third Parties, these new reducers with (necessarily extensive) rights are a tempting target for attacks to gain additional new rights. In both cases, this risk can somewhat be reduced with a system policy that defines the types and size of chains it tries to process or how often a user can request reductions. Also, even if the reducer is unavailable, the original chain can still be used, just with slower performance and less privacy, so the system does not become unavailable, but performance and privacy might suffer. This final risk can be mitigated in some cases by proactively requesting reductions well before they are required, thus increasing the likelihood that the reduction will succeed in time (only a long term unavailability would prevent reduction).

A new benefit stemming from reduction is the possibility of users granting themselves limited rights, thus reducing potential loss should this new right be lost. This functionality does not necessarily require reductions (simple delegation to a new identity suffices), but reduction helps keep the chain short, thus making the reduced right at least equally fast as the original chain. Similarly, a right could be delegated to many devices to improve redundancy - and reduction then offers equal performance.

As in the case of performance and privacy, online checks affect security. If a check requires that the verifier retrieves information, such as a certificate revocation list, from another source, this can be used to create

a resource exhaustion attack against the verifier. This attack can be countered partially by, e.g. verifying the authority of the requester before online validations or by requiring the requester to provide a CRL instead of the verifier fetching it (all CRLs are signed by their issuer). Existing methods are already susceptible to this weakness, so the new method does not introduce new weakness.

We can conclude that the new online methods and chain reduction rely on existing methods, so they do not open any new major security problems, while making some existing easier to handle.

## 4.5 Applicability and Scalability

The certificate model is particularly suited for situations with distributed rights granting (e.g. many issuing locations for a credit card) and a large number of resources (e.g. the public transit). In this case, it is more effective to have the right with the user rather than maintain a centralised database of rights that is available to all involved parties whenever they require. Also, use cases which do not require continuous validation of the right can benefit from the offline nature of certificate based access control. Certificates can also be used in many other cases, but particularly in simple scenarios with e.g. a single resource and few users, other technologies might prove more effective solutions.

As certificates require cryptography, storage and communications, this limits the types of devices that can use certificates. The prototype in Publication V demonstrated that modern embedded platforms with software cryptography are already capable of utilising certificate based access control. With the progress of technology, such capabilities are within reach for smaller devices. One emerging application area is Internet of Things (IoT), which is characterised by users having a large number of devices to manage – here, the ability to use delegation for simplifying rights management e.g. by grouping devices and and granting temporary rights to visiting friends has great benefits. Also, some IoT devices, such as sensors embedded in the structure of buildings, have very limited resources (particularly energy), but hardware cryptography can make these applications feasible. Hence, even quite small embedded devices are already capable of utilising certificates. And using chain reduction to cut down the number of operations required can be a improvement by moving some of the resource consuming operations to

other, better equipped devices.

Scalability is mainly limited by the resources. Our analysis in Publication V shows that storage and communication requirements are relatively modest and cryptography is likely the most difficult requirement for implementation. However, with current hardware, software implementations are in many cases sufficient and hardware implementations are an option, if further performance improvements are required.

### 4.5.1 Scalability example

To quantify the required level of performance, I use the reduction server run by the Transit Authority in the public transit example and make some pessimistic worst case assumptions. We assume that for every trip the user requests a new reduction to an ephemeral identity. For one million daily trips, we can assume roughly half are during the morning hours, and to make the situation harder, half of those would request a reduction during the busiest 15 minutes. So, during the busiest 15 minutes there would be 250.000 requests.

Now, each reduction request requires 4 signature operations (as discussed in Section 4.2, the cryptographic operations take over 99.7% of the computation, so we can use them to estimate the overall computation required): one new certificate and the service request to verify, and the reduction certificate and the service reply to sign (the server would cache the certificates for all the sales points, so only the certificate from the sales point to the passenger has to be verified). So, for a city with 1 million daily trips, the server has to be able to do one million signature operations (250 000 x 4) during the busiest 15 minutes, which equals roughly 1 100 signature operations per second. Our embedded platform was capable of 15 signature operations per second, so we would need 2 orders of magnitude improvement in performance – almost within the capabilities of a modern mobile phone let alone a server (as a comparison: a modern Transport Layer Security (TLS) server can manage almost 10.000 transactions per second using the same signature operation per transaction [97]). So, processing-wise, a single server is capable of performing the reductions for a city of some 10 million daily trips even with our worst case assumptions.

The storage requirements are even easier: one million 256-bit hashes takes only 32 MB, so caching every single passenger certificate in the system in feasible even for relatively small devices.

Finally, the communication for the (single) reduction server is quite feasible: as stated in Publication V, a certificate and a reduction request are both typically around 1 kB in size. Thus, a reduction request with three certificates takes some 4 kB, and the 250 000 reduction requests during 15 minutes results in traffic of less than 10 Mbps. We can conclude that scalability-wise, even very large systems are feasible.

## 4.6 Usability

The usability study as presented in Publication VI looks at how certificates should be presented to users and how much of the certificate technology should be revealed and what can and should be hidden. The key finding is that in most cases, the end-user does not need to be aware of certificates or validation methods as such. They would be confused by introducing new concepts that have nothing to do with the user's actual goals. Instead, the end-user should be presented with concepts and information that match her goals: a bus ticket with an expiration time etc.

A designer of a system has to understand certificates and validation methods. The designer is responsible for choosing a suitable method with which to support the user's goals and for analysing which options are relevant to the end-user. In each use case the required functionality could be implemented with only one or two different validity methods – therefore, offering the end user a full list of methods is a bad idea. We also noted that the CRL, Reval and renew methods can be made to look identical to the end-user (except in a limited set of cases), so the choice between them can be based on technical reasons.

The designer should follow established usability and interaction design guidelines in creating the system. Visibility of the user's data and the system's state are important. The end-user must be able to see what rights she has acquired and how much of the usage quota remains. The user must also be able to easily see to whom she has delegated rights and which rights have been revoked. Additionally, all information should be shown in its context, not in a separate 'certificate management' context.

# 5. Discussion

In this chapter, I comment on the extended SPKI as an access control solution and the new possibilities it opens up, as well as the opportunities we have, should we go beyond orthodoxically following the existing access control rules. This section is based mostly on Publication V.

## 5.1 Access Control with many possibilities

With the extensions proposed in this thesis, I find that SPKI is a comprehensive access control solution suitable for many use cases. It is particularly suited for distributed systems with its ability to tolerate being offline.

Our prototype implementation shows that certificates have quite modest system requirements, which means we can use such systems quite liberally: instead of having a gate controlling the whole parking area we could have a small device monitoring each parking space; it does not prevent a misbehaving driver from parking, but it alerts the attendant. Many indoor parking areas already deploy a sensor above every parking space to monitor their availability, so this would be an extension of it.

At the other end of the system, the good scalability means that even servers have moderate resource requirements, as our scalability calculation shows.

A particularly interesting application area is IoT. With a large number of devices that can have limited network connection (often due to a limited energy budget), the ability to operate offline much of the time can be a big benefit. Another issue that can potentially benefit from SPKI-type solutions is management of access right; with SPKI, it is easy to use delegation to group devices and then (even just temporarily) grant them the required rights.

Privacy-wise, anonymous, ephemeral identities and the ability to use reduction to further hide some information makes SPKI interesting for many privacy conscious applications - which in the era of pervasive network monitoring can be quite numerous.

Finally, the hourglass-hypothesis that many systems can benefit from offers an efficient reduction scheme, where the user only presents a single certificate to use a right regardless of the length of the original certificate chain. This helps provide consistent performance for the system and alleviate many of the traditional concerns about chain length.

Authorisation certificates use a lazy evaluation approach – that is, we only evaluate the state of the granted right or even evaluate, whether a particular right has been granted, when some tries to use the right (as opposed to maintaining an up-to-date database about which rights have been granted). In many applications this is a desirable property and it is in fact the property that allows us to grant new certificates offline. However, it also means that certificates might not be a good solution for systems where it is important to know at all times what rights have been granted. So, in the worst case, in our Public Transit System the Transit Authority would not know how many tickets have been sold until the tickets are used and the bus companies report the activity. However, in a real solution, the resellers would likely report their sales at regular intervals (this can be enforced using the `limit`-condition, if required) and the automatically requested reductions would also reveal the number of tickets before their use. Thus the use case and system design strongly determine whether this property is a weakness or a strength.

Regardless of the many positive attributes, so far, authorisation certificates have been used in only a few commercial applications. They still receive some research attention, but widespread utilisation has yet to materialise. One likely reason is the required cryptographic resources, which so far have been assumed to preclude all but the largest devices. However, the progress of technology and hardware support for cryptography can now bring these capabilities even to smaller devices – so these barriers are quickly coming down. While software implementation can flexibly support many cryptographic algorithms, successful deployment of hardware cryptography is contingent on a sufficiently wide agreement on the algorithms used for that application area. Another possible explanation is the unfinished standardisation of SPKI and similar solution and the resultant lack of protocols and

implementations.

Finally, it is possible that so far, the benefits have not been perceived to be significant enough compared to existing solutions. This work hopefully highlights some of the benefits and facilitates the adoption of SPKI.

## 5.2 Beyond Orthodox Access Control

So far, I have discussed access control with an orthodox approach: we make the reductions and access control decisions strictly as defined in the specifications. But this approach has limitations: sometimes there may be technical problems that prevent us from having all the required information to make a decision – and we still have to make a decision. The best way to deal with this situation depends on the business model of the application. Other times, we might have additional information when making the reductions that make it reasonable to make reductions that e.g. grant more rights or have fewer limitations than the original chain. In Publication V we discuss such an approach to reduction and access control in general, and point out some potential benefits of this approach. Finally, the business model of the application can and should influence application design as it can lead to technically more optimised solutions e.g. by using less potent but technologically lighter revocation methods to implement the required functionality.

### 5.2.1 Technical limitations

One important design limitation for all distributed systems is the CAP Theorem, which states that any distributed system can achieve at most two of the following three characteristics: Consistency, Availability and (network) Partition Tolerance [20]. In access control context this means that we can make correct access control decisions whenever required *only* if we have a network that is always available. In reality, of course, we have to live with occasional network outages and either cease to make decisions or make them based on incomplete data. Compared to a pure online solution, SPKI certificates do provide the resource some information to make a decision even if all the online validations cannot be made.

The choice is, therefore, application specific depending on whether the risk-benefit-ratio of a false positive or a false negative is higher to the system. If the solution is a transit ticket system, we would rather allow

the passenger travel with an invalid ticket (false positive), because the loss is minor; however, denying travel with a valid ticket (false negative) might result in bad publicity and much larger overall losses. However, if the solution is a credit card and the customer is buying something expensive, the loss from a false positive might be too big a risk to take. In practice, the behaviour of the system in such a situation would be defined by the system policy.

### 5.2.2 Using Additional Information to Create Reductions

So far, we have discussed reductions that match the original chain. Next we consider a case where the reducer takes additional factors into account while creating the reduction and can, based on the system policy, change the details in the CRC. Typically, the system includes a policy that states what other information can and should be used when creating reductions. In this case, we could e.g. create reductions to an ephemeral identity even if the original chain does not allow for this or reductions with more rights than the original chain had, if the requested right/validity period complies with the system policy.

If the system is designed so that all certificates use the same revocation server (database), an entry may be created in the database linked to the other certificates causing the CRC to be revoked automatically if any of the certificates upon which it was based on are revoked – enabling us to replace a number of online validations with a single validation having the same effect. This would make validating the CRC much easier and would provide additional privacy because all ephemeral CRCs (e.g. credit cards) would have a single, but different, online validation. Implementing such validation combining functionality merits further research.

The system policy may also allow the certificate to be validated for a period short enough that there is no requirement for revocation. For example, the certificate can be a bus ticket that needs to be valid only for a short period. In a similar way, the reducer may replace or omit various online tests according to the system policy, if the right is constrained enough. In both cases, omitting online validations enhances both performance and privacy.

The implication of this *context aware* reduction for system design is that certificates and chain reduction can be used to design a system that protects the users' privacy from most parties, while allowing portable access rights and authorisation.

### 5.2.3 The Business Model factor

The goal for system design should be to optimise the solution to meet the requirements of the business case. This means that we maintain the risks related to granting rights to different parties acceptable to the business model. Whenever we grant a right to a party, there is always a risk that the party loses or misuses the right. To minimise the resultant problems, the system designer is tempted to use limited rights and fast-acting revocation methods to limit the exposure. The downside is that fast-acting methods require more online checking than slower methods, which can result in costly implementation and *reduced* performance.

For a business model, the real question is that somebody has to pay for the usage – so immediate prevention of misuse is not a necessity, if somebody pays for the misuse. In the public transit example, for instance, if a passenger loses a ticket, the revocation does not need to be immediate, but could take e.g. a week, if the the service provider can simply charge the passenger for the revocation period.

# 6. Summary

In this thesis I have looked at the problem of access control in distributed systems and how it can be solved using SPKI authorisation certificates. I have identified and designed many missing pieces of the SPKI infrastructure to make it a well rounded access control solution. I then analysed the resultant system from several angles and found that with the extensions proposed in this thesis, SPKI certificates offer a flexible system for many application areas, particularly when privacy is a priority.

The key findings can be summarised from three viewpoints: technology, users and business model. Technology-wise, SPKI authorisation certificates are particularly suited for large distributed systems with multiple resources and users - here, the fact that the right stays with the user makes it an efficient solution. Another suitable application area are systems that benefit from tolerance for offline operation. Performance-wise, our prototype implementation demonstrated that small embedded devices are powerful to utilise SPKI certificates for access control. At the other end, our scalability calculations show, quite large systems can be implemented. By wisely utilising certificate chain reduction, the system designer can further improve both performance and scalability by moving some of the required computation to more capable devices, thus improving system responsiveness. The hourglass-reduction, in particular, seems to be an effective way of organising the system and improving performance.

For users, be they end users or system administrator, SPKI certificate based access control is a flexible tool. However, to make the access control user friendly, we found that it is important to provide information in its proper context using the terminology of the application in question, not within a generic certificate management interface. Another benefit of SPKI is support for user privacy, which can be further enhanced with use of certificate chain reduction. In both cases, the system designer has

a significant role in how well the information is presented and how well the user's privacy is preserved.

The business model of the application in question has a large impact on the design choices in access control. We can use technologically lighter revocation methods by including the increased risk in the business model, thus providing better user experience with increased performance. Also, unavoidable exceptional situations (e.g. network outages) can be better dealt with by preparing for these situations according to the business model of the application in question.

Future work in this area includes implementing prototypes to better understand how certificates can compete with established solutions. Another area for future work is to study how chain reduction could be applied and the effect it would have on systems using identity certificates, such as X.509 certificates.

# References


[1] Martín Abadi. On SDSI's linked local name spaces. In *Proceedings of the 10th Computer Security Foundations Workshop*, pages 98–108. IEEE Comput. Soc. Press, 1997.

[2] Martín Abadi, M Burrows, B Lampson, and G Plotkin. A calculus for access control in distributed systems. *... Languages and Systems ( ...*, 1993.

[3] Sudhir Agarwal, Barbara Sprick, and Sandra Wortmann. Credential based access control for semantic web services. *AAAI Spring Symposium-Semantic Web Services*, pages 44–52, 2004.

[4] S Ajmani, D Clarke, C H Moh, and S Richman. ConChord: Cooperative SDSI certificate storage and name resolution. In *First International Workshop on Peer-to-Peer Systems (IPTPS)*, pages 141–154. Springer, 2002.

[5] F Alberti, A Armando, and S Ranise. Efficient symbolic automated analysis of administrative attribute-based RBAC-policies. *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 165–175, 2011.

[6] A W Appel and E W Felten. Proof-carrying authentication. *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 52–62, 1999.

[7] T Aura. On the structure of delegation networks. In *11th IEEE Computer Security Foundations Workshop*, pages 14–26, 1998.

[8] S Barker. The next 700 access control models or a unifying meta-model? *Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 187–196, 2009.

[9] Jim Basney, Terry Fleury, and Von Welch. Federated login to TeraGrid. In *IDTRUST '10: Proceedings of the 9th Symposium on Identity and Trust on the Internet*. ACM, 2010.

[10] L Bauer, L F Cranor, R W Reeder, M K Reiter, and K Vaniea. Real life challenges in access-control management. *Proceedings of the 27th international conference on Human factors in computing systems*, pages 899–908, 2009.

[11] L Bauer, S Garriss, and M K Reiter. Detecting and resolving policy misconfigurations in access-control systems. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):2, 2011.

[12] Lujo Bauer, Limin Jia, Michael K Reiter, and David Swasey. xDomain: cross-border proofs of access. In *SACMAT '09: Proceedings of the 14th ACM symposium on Access control models and technologies*. ACM, 2009.

[13] D Berbecaru. MBS-OCSP: an OCSP based certificate revocation system for wireless environments. In *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology*, pages 267–272, 2004.

[14] D Berbecaru. On the tradeoff between performance and security in OCSP-based certificate revocation systems for wireless environments. In *IEEE Symposium on Computers and Communications, ISCC'06.*, pages 340–346. IEEE, 2006.

[15] I Bilogrevic, M H Manshaei, M Raya, and J P Hubaux. Optimal revocations in ephemeral networks: A game-theoretic framework. *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pages 21–30, 2010.

[16] J Biskup and S Wortmann. Towards a credential-based implementation of compound access control policies. *Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 31–40, 2004.

[17] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the IEEE Symposium on Security and Privacy 1996*, pages 164–173, 1996.

[18] Piero Bonatti, Sabrina De Capitani di Vimercati, and Pierangela Samarati. An algebra for composing access control policies. *ACM Transactions on Information and System Security*, 5(1):1–35, 2002.

[19] Kevin Borders, Xin Zhao, and Atul Prakash. CPOL: high-performance policy evaluation. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*. ACM, 2005.

[20] Eric A. Brewer. Towards robust distributed systems. *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing*, 19:7–10, 2000.

[21] Óscar Cánovas and Antonio F. Gómez. A distributed credential management system for spki-based delegation systems. *Proceedings of 1st Annual PKI Research Workshop*, pages 65–76, 2002.

[22] D Chadwick and S Anthony. Using WebDAV for improved certificate revocation and publication. *Public Key Infrastructure*, pages 265–279, 2007.

[23] P Chapin, C Skalka, and X S Wang. Risk assessment in distributed authorization. *Proceedings of the 2005 ACM workshop on Formal methods in security engineering*, pages 33–42, 2005.

[24] Peter C Chapin, Christian Skalka, and X Sean Wang. Authorization in trust management. *ACM Computing Surveys*, 40(3):1–48, 2008.

[25] N Chetcuti and F Massacci. Reasoning about Naming and Time for Credential-based Systems. page 23, 2008.

[26] Jin-Hee Cho, Ananthram Swami, and Ing-Ray Chen. A Survey on Trust Management for Mobile Ad Hoc Networks. *IEEE Communications Surveys & Tutorials*, 13(4):562–583, 2011.

[27] D Clarke, J E Elien, Carl M Ellison, M Fredette, A Morcos, and R L Rivest. Certificate chain discovery in SPKI/SDSI. *Journal of Computer Security*, 9(4):285–322, 2001.

[28] D A Cooper. A model of certificate revocation. In *Proceedings of the 15th AnnualComputer Security Applications Conference (ACSAC '99)*, pages 256–264. IEEE Comput. Soc, 1999.

[29] D A Cooper. A more efficient use of delta-CRLs. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pages 190–202, 2000.

[30] C Crépeau and C R Davis. A certificate revocation scheme for wireless ad hoc networks. *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 54–61, 2003.

[31] J de RP Braga Jr, A C T Vidal, F Kon, and M Finger. Trust in large-scale computational grids: an SPKI/SDSI extension for representing opinion. *Proceedings of the 4th international workshop on Middleware for grid computing*, page 7, 2006.

[32] A K Eamani and A P Sistla. Language based policy analysis in a SPKI trust management system. *Journal of Computer Security*, 14(4):327–357, 2006.

[33] S Eichler and B Muller-Rathgeber. Performance analysis of scalable certificate revocation schemes for ad hoc networks. *The IEEE Conference on Local Computer Networks*, pages 9 pp.–391, 2005.

[34] Carl M Ellison. SPKI Requirements, Request for Comments 2692, 1999.

[35] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylönen. SPKI certificate theory, Request for Comments 2693, 1999.

[36] Carl M Ellison, Bill Frantz, Butler W Lampson, Ronald L Rivest, Brian M. Thomas, and Tatu Ylönen. Simple Public Key Certificate, 1999.

[37] F Elwailly, C Gentry, and Z Ramzan. QuasiModo: Efficient certificate validation and revocation. *Public Key Cryptography–PKC 2004*, pages 375–388, 2004.

[38] Simon Foley and Hongbin Zhou. Authorisation subterfuge by delegation in decentralised networks. In *Proceedings of the 13th international conference on Security protocols*. Springer-Verlag, 2005.

[39] P W L Fong. Relationship-based access control: protection model and policy language. *Proceedings of the first ACM conference on Data and application security and privacy*, pages 191–202, 2011.

[40] A Fongen. Identity Management without Revocation. In *Fourth International Conference on Emerging Security Information Systems and Technologies (SECURWARE)*, pages 75–81, 2010.

[41] Slava Galperin, Stefan Santesson, Michael Myers, Ambarish Malpani, and Carlisle Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. Request for Comments 6960, 2013.

[42] A Ganesh and K Gopinath. SPKI/SDSI certificate chain discovery with generic constraints. *Proceedings of the 1st Bangalore Annual Compute Conference*, page 3, 2008.

[43] Xiuhua Geng, Zhen Han, and Li Jin. A First-order Logic Semantics for SPKI/SDSI. In *ISDPE '07: Proceedings of the The First International Symposium on Data, Privacy, and E-Commerce*. IEEE Computer Society, 2007.

[44] K Govindan and P Mohapatra. Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey. *Communications Surveys & Tutorials, IEEE*, 14(2):279–298, 2012.

[45] Shangyuan Guan, Xiaoshe Dong, Weiguo Wu, Yiduo Mei, and Guofu Feng. A Distributed Trust Management Based on Authorizing Negotiation in Open and Dynamic Environment. In *22nd International Conference on Advanced Information Networking and Applications (aina 2008)*, pages 32–39. IEEE, 2008.

[46] J J Haas, Y C Hu, and K P Laberteaux. Design and analysis of a lightweight certificate revocation mechanism for VANET. *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking*, pages 89–98, 2009.

[47] Jason J Haas, Yih-Chun Hu, and Kenneth P Laberteaux. Efficient Certificate Revocation List Organization and Distribution. *IEEE Journal on Selected Areas in Communications*, 29(3):595–604, 2011.

[48] Joseph Y Halpern and Ron van der Meyden. A logical reconstruction of SPKI. *Journal of Computer Security - Special issue on CSFW14*, 11(4):581–613, 2003.

[49] S Heikkinen and S Siltala. Service Usage Accounting. *Vehicular Technology Magazine, IEEE*, 6(1):60–67, 2011.

[50] J Howell and D Kotz. A formal semantics for SPKI. *Computer Security-ESORICS 2000*, pages 140–158, 2000.

[51] K Hristova, K T Tekle, and Y A Liu. Efficient trust management policy analysis from rules. *Proceedings of the 9th ACM SIGPLAN international conference on Principles and practice of declarative programming*, pages 211–220, 2007.

[52] Y Hui, L Jicheng, W Yuhua, and Z Dexian. An Algorithm of Constructing Certificates Chain for Trusted P2P Network. *Challenges in Environmental Science and Computer Engineering (CESCE), 2010 International Conference on*, 2:97–100, 2010.

[53] Ladislav Huraj and Vladimír Siládi. Authorization through trust chains in ad hoc grids. In *EATIS '09: Proceedings of the 2009 Euro American Conference on Telematics and Information Systems: New Opportunities to increase Digital Citizenship*. ACM, 2009.

[54] M Ingle and M Kumar. Comparative Analysis of Methods for Distribution of Certificate Revocation Information in Mobile Environment. *International Conference on Communication Systems and Network Technologies (CSNT)*, pages 166–169, 2011.

[55] K Jayaraman, V Ganesh, M Tripunitara, M Rinard, and S Chapin. Automatic error finding in access-control policies. *Proceedings of the 18th ACM conference on Computer and communications security*, pages 163–174, 2011.

[56] S Jha and T Reps. Model checking spki/sdsi. *Journal of Computer Security*, 12(3):317–353, 2004.

[57] A Jø sang, D Gollmann, and R Au. A method for access authorisation through delegation networks. *Proceedings of the 2006 Australasian workshops on Grid computing and e-research-Volume 54*, pages 165–174, 2006.

[58] J Kim and D Lee. An access control using SPKI certificate in peer-to-peer environment. *Computational Science and Its Applications–ICCSA 2007*, pages 148–156, 2007.

[59] Yki Kortesniemi. SPKI performance and certificate chain reduction. In *GI Jahrestagung*, pages 449–454, 2002.

[60] Yki Kortesniemi. Validity Management in SPKI. In *Proceedings of the 1st Annual PKI Research Workshop*, pages 27–36, 2002.

[61] Yki Kortesniemi. Managing the Usage of Authorisation Certificates. Licentiate's thesis, 2003.

[62] Yki Kortesniemi, Tero Hasu, and Jonna Särs. A Revocation, Validation and Authentication Protocol for SPKI Based Delegation Systems. In *Proceedings of the Internet Society Symposium on Network and Distributed System Security (NDSS)*, 2000.

[63] K P Laberteaux, J J Haas, and Y C Hu. Security certificate revocation list distribution for VANET. *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, pages 88–89, 2008.

[64] A Lee, J Boyer, C Drexelius, P Naldurg, R Hill, and R Campbell. Supporting dynamically changing authorizations in pervasive communication systems. *Security in Pervasive Computing*, pages 134–150, 2005.

[65] A J Lee and M Winslett. Safety and consistency in policy-based authorization systems. *Proceedings of the 13th ACM conference on Computer and communications security*, pages 124–133, 2006.

[66] Bao-Hong Li, Yi-Bin Hou, and Yin-Liang Zhao. A scalable scheme for certificate revocation. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, volume 6, pages 3852–3856, 2005.

[67] J Li, Y Zhu, H Pan, and S Liu. A distributed certificate revocation scheme based on one-way hash chain for wireless ad hoc networks. *2nd International Conference on Mobile Technology, Applications and Systems*, pages 5 pp.–5, 2005.

[68] Jiangtao Li, Ninghui Li, XiaoFeng Wang, and Ting Yu. Denial of Service Attacks and Defenses in Decentralized Trust Management. *Securecomm and Workshops*, pages 1–12, 2006.

[69] Jiangtao Li, Ninghui Li, and William H Winsborough. Automated trust negotiation using cryptographic credentials. *ACM Transactions on Information and System Security*, 13(1):1–35, 2009.

[70] N Li, B N Grosof, and J Feigenbaum. Delegation logic: A logic-based approach to distributed authorization. *ACM Transactions on Information and System Security (TISSEC)*, 6(1):128–171, 2003.

[71] Ninghui Li, William H Winsborough, and John C Mitchell. Distributed credential chain discovery in trust management: extended abstract. In *CCS '01: Proceedings of the 8th ACM conference on Computer and Communications Security*. ACM, 2001.

[72] C Lin, L Huan-zhou, and H Yong. A digital certificate revocation status promulgation scheme based on square residue. *Proceedings of the International Symposium on Autonomous Decentralized Systems, ISADS 2005*, pages 208–211, 2005.

[73] W Liu, H Nishiyama, N Ansari, and N Kato. A Study on Certificate Revocation in Mobile Ad Hoc Networks. *IEEE International Conference on Communications (ICC)*, pages 1–5, 2011.

[74] Javier Lopez, Isaac Agudo, and Jose A Montenegro. On the deployment of a real scalable delegation service. *Information Security Technical Report*, 12(3):139–146, 2007.

[75] C Ma, N Hu, and Y Li. On the release of CRLs in public key infrastructure. *Proceedings of the 15th conference on USENIX Security Symposium*, 2006.

[76] J L Munoz and J Forné. Design of a certificate revocation platform. In *Proceedings of ITRE2003, International Conference on Information Technology: Research and Education*, pages 259–263, 2003.

[77] J L Munoz, J Forné, O Esparza, and M Soriano. A test-bed for certificate revocation policies. In *2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, volume 2, pages 561–564, 2003.

[78] Divya Muthukumaran, Sandra Rueda, Hayawardh Vijayakumar, and Trent Jaeger. Cut me some security. In *SafeConfig '10: Proceedings of the 3rd ACM workshop on Assurable and usable security configuration*. ACM, 2010.

[79] M Naor and K Nissim. Certificate revocation and certificate update. *Selected Areas in Communications, IEEE Journal on*, 18(4):561–570, 2000.

[80] G Navarro, J Borrell, J A Ortega-Ruiz, and S Robles. Access control with safe role assignment for mobile agents. *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1235–1236, 2005.

[81] S Nazareth and S Smith. Using spki/sdsi for distributed maintenance of attribute release policies in shibboleth. *Computer Technical Report TR2004-485*, 2004.

[82] P Nikander and L Metso. Policy and trust in open multi-operator networks. *Proceedings of IFIP SmartNet'2000*, 2000.

[83] Pekka Nikander. *An Architecture for Authorization and Delegation in Distributed Object-Oriented Agent Systems*. PhD thesis, 1999.

[84] M Pala. A proposal for collaborative Internet-scale trust infrastructures deployment: the Public Key System (PKS). *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, pages 108–116, 2010.

[85] K Papapanagiotou, G F Marias, and P Georgiadis. A Certificate Validation Protocol for VANETs. *IEEE Globecom Workshops*, pages 1–9, 2007.

[86] V Patil, A Mei, and L V Mancini. Addressing interoperability issues in access control models. *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pages 389–391, 2007.

[87] B C Popescu, B Crispo, and A S Tanenbaum. A certificate revocation scheme for a large-scale highly replicated distributed system. *Proceedings of the Eighth IEEE International Symposium on Computers and Communications (ISCC 2003)*, pages 225–231 vol. 1, 2003.

[88] T Saito, K Umesawa, and H G Okuno. Privacy enhanced access control by SPKI. In *Seventh International Conference on Parallel and Distributed Systems: Workshops*, pages 301–306, 2000.

[89] Ghassan Samara, Sureswaran Ramadas, and Wafaa A H Al-Salihy. Design of Simple and Efficient Revocation List Distribution in Urban areas for VANET's. *International Journal of Computer Science and Security (IJCSS)*, 8(1), 2010.

[90] A Santin, J da Silva Fraga, and C Maziero. Extending the SDSI/SPKI model through federation webs. *Communications and Multimedia Security. Advanced Techniques for Network and Data Protection*, pages 132–145, 2003.

[91] A O Santin, J da Silva Fraga, F Siqueira, and E R de Mello. Federation web: A scheme to compound authorization chains on large-scale distributed systems. *Proceedings of the 22nd International Symposium on Reliable Distributed Systems*, pages 66–75, 2003.

[92] K Scheibelhofer. PKI without Revocation Checking. *4th Annual PKI R&D Workshop*, 2005.

[93] S Schwoon, S Jha, T Reps, and S Stubblebine. On generalized authorization problems. *Proceedings of the 16th IEEE Computer Security Foundations Workshop*, pages 202–216, 2003.

[94] S W Smith, C Masone, and S Sinclair. Expressing trust in distributed systems: the mismatch between tools and reality. *Forty-Second Annual Allerton Conference on Privacy, Security and Trust*, pages 29–39, 2004.

[95] B Song, I K Yu, J Son, and D K Baik. An effective access control mechanism in home network environment based on SPKI certificates. *IEEE International Conference on Information Theory and Information Security (ICITIS)*, pages 592–595, 2010.

[96] S D Stoller, P Yang, C R Ramakrishnan, and M I Gofman. Efficient policy analysis for administrative role based access control. *Proceedings of the 14th ACM conference on Computer and communications security*, pages 445–455, 2007.

[97] Nick Sullivan. ECDSA: The digital signature algorithm of a better internet. https://blog.cloudflare.com/ecdsa-the-digital-signature-algorithm-of-a-better-internet/, 2014.

[98] D Sun, Y Luo, Q Cao, Q Li, S Y Chen, and A Xu. UCSMdess: ubiquitous computing service model based on D-S evidence theory and extended SPKI/SDSI. *7th WSEAS International Conference on Applied Computer and Applied Computational Science*, (7), 2008.

[99] Y Sun, R Lu, X Lin, J Su, and X Shen. NEHCM: A Novel and Efficient Hash-chain based Certificate Management scheme for vehicular communications. *5th International ICST Conference on Communications and Networking in China (CHINACOM)*, pages 1–5, 2010.

[100] Dejvuth Suwimonteerabuth. Computing Minimum-Height Certificate Trees in SPKI/SDSI. *10th International Conference on Innovative Internet Community Systems*, 165:340–349, 2010.

[101] V Ungureanu. Efficient support for enterprise delegation policies. *Proceedings of the 2005 ACM symposium on Applied computing*, pages 340–345, 2005.

[102] V Vasudevan, N Sivaraman, S Senthil Kumar, R Muthuraj, J Indumathi, and G V Uma. A comparative study of SPKI/SDSI and K-SPKI/SDSI SYSTEMS. *Information Technology Journal*, 6(8):1208–1216, 2007.

[103] Wattana Viriyasitavat and Andrew Martin. A Survey of Trust in Workflows and Relevant Contexts. *IEEE Communications Surveys & Tutorials*, 2012.

[104] Hao Wang, Somesh Jha, Thomas Reps, Stefan Schwoon, and Stuart Stubblebine. Reducing the dependence of SPKI/SDSI on PKI. In *ESORICS'06: Proceedings of the 11th European conference on Research in Computer Security*. Springer-Verlag, 2006.

[105] Min Wang. On the Security of Peer-to-Peer Computing Based on SPKI/SDSI. In *International Conference on Internet Technology and Applications (iTAP)*, pages 1–4, 2011.

[106] M Wangham, J da Silva Fraga, R Obelheiro, G Jung, and E Fernandes. Security mechanisms for mobile agent platforms based on spki/sdsi chains of trust. *Software Engineering for Multi-Agent Systems II*, pages 362–363, 2004.

[107] M Wangham, E R de Mello, J da Silva Fraga, and D da Silva Boger. A Model to support SPKI Federations management through XKMS. *IEEE*

*International Conference on Web Services, ICWS 2007*, pages 338–345, 2007.

[108] A Wasef and Xuemin Shen. EDR: Efficient Decentralized Revocation Protocol for Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, 58(9):5214–5224, 2009.

[109] P Wohlmacher. Digital certificates: a survey of revocation methods. *Proceedings of the 2000 ACM workshops on Multimedia*, pages 111–114, 2000.

[110] D Wojtczak. Trust metrics for the SPKI/SDSI authorisation framework. *Automated Technology for Verification and Analysis*, pages 168–182, 2011.

[111] Rebecca N Wright, Patrick D Lincoln, and Jonathan K Millen. Efficient fault-tolerant certificate revocation. In *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*. ACM, 2000.

[112] G Ying and Z Jiang. Hildht: Hilbert-based SDSI/SPKI certificate storage and relative search algorithm. In *2nd International Conference on Education Technology and Computer (ICETC)*, 2010.

[113] Gao Ying and Zhan Jiang. Research on CRL distribution in P2P systems. In *2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009*, pages 574–577, 2009.

[114] S Zhang and H Wang. An Improved Delta and Over-issued Certificate Revocation Mechanism. *International Colloquium on Computing, Communication, Control, and Management*, 2:346–350, 2008.

[115] P Zheng. Tradeoffs in certificate revocation schemes. *ACM SIGCOMM Computer Communication Review*, 33(2):103–112, 2003.

Aalto-DD 63/2015

ISBN 978-952-60-6193-1 (printed)
ISBN 978-952-60-6194-8 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

**Aalto University**
**School of Electrical Engineering**
**Department of Communications and Networking**
**www.aalto.fi**

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

**DOCTORAL DISSERTATIONS**