SPKI Performance and Certificate Chain Reduction

Yki Kortesniemi
Helsinki Institute for Information Technology
Yki.Kortesniemi@hiit.fi

Abstract:

Authorisation certificate based access control owes much of its expressive power to delegation; delegation enables distributed access control management, where the authorisation decisions are manifested as certificate chains. Unfortunately, these chains have to be evaluated every time a right is used, and if the right is used repeatedly, this can result in significant performance overhead. However, if the chains are replaced with reduction certificates, this overhead can be cut down.

In this paper we discuss performance in SPKI and how it can be improved with certificate chain reduction. We elaborate on certificate chains, reduction certificates, and their performance implications, the choice of issuers of reduction, and take a look at the problems of reducing chains with online validity checks.

1 Introduction

Implementing a global service for a multitude of users can present daunting management challenges for the access control technology used. One solution is to use a technology that allows the management rights to be distributed along with the access rights as authorization certificates do. The Internet Engineering Task Force (IETF) has been developing Simple Public Key Infrastructure (SPKI) as a more flexible alternative to X.509 [EFL+99, EFL+]. The key idea in SPKI is that anyone (or anything) with access to a resource can authorize others to use the resource by issuing them an authorisation certificate. Further, the authorisation certificates can be used to delegate the rights to other users without any help from the owner of the resource: users can delegate their own rights. These certificates therefore form chains, which always start from the verifier controlling access to the resource, go through 0-N intermediate entities (e.g. administrators) and end with the actual user of the resource. This means that it becomes possible e.g. for parents to create new credit cards that make it possible for children to use their parent's credit rights in such a way that the parents keep their own card and the children have a limit to the amount they can charge from the card.

A result of this process is that the user (e.g. the child) could end up with a long chain of certificates that has to be presented whenever the right is used - and storing, handling and evaluating long chains can result in significant performance overhead. In this paper we look at how this overhead could be reduced by using chain reduction certificates (or certificate result certificates), CRCs, that replace a chain of certificates with a single certificate having the same properties as the chain. The rest of the paper is organized as follows: Section 2 elaborates on the motivations for chain reduction; section 3 discusses reducing chains that have online validations; section 4 talks about different reducers and section 5 presents my conclusions.

2 Motivations for Reducing Certificate Chains

The SPKI theory introduces the concept of a CRC - a certificate that corresponds to the semantics of the underlying certificates and online test results [EFL$^+$99]. The main motivation for creating CRCs is performance benefits:

- discovering the correct chain from a pool of certificates is not a trivial operation [Aur98],
- as neither the user's terminal nor the verifier always has storage space for long chains, some of the certificates might even have to be fetched from the network, adding further overhead [HK00],
- and even with the correct certificates, deducing the access decision from the rights expressed in the certificates present challenges [BD02].

By using a CRC, we can avoid repeating these costly operations and the verifier can instead evaluate a single certificate to reach the access decision. But here we also note that for a CRC to make sense from performance perspective, it normally has to be used repeatedly. More precisely, the cost of creating and using the CRC over its lifetime has to be less than the cost of using the chain (without creating the CRC) for the CRC to be beneficial. Naturally, it is not always possible to know in advance, whether a particular CRC will be used again in the future, but there has to be at least the possibility for that CRC to make sense. Another justification for a CRC can be the need to free resources from the (potentially burdened) verifier by having someone else create a reduction of a section of the chain. A third motivation for creating CRCs is to promote anonymity by hiding parties in the chain as proposed in [NKP99].

3 Reduction Certificates and Online Validations

SPKI structure draft [EFL$^+$] defines several online validity conditions used to limit the usage of the certificate. [KHS00] further adds a couple more, one of which, `limit`, creates particular complications for reductions, as we shall soon see (`limit` is used to create certificate with a controlled amount of usage such as a credit card with a monthly quota or a bus ticket for 10 journeys - without `limit` these kind of applications are not possible).

In creating CRCs, there are two options: all the online validations can be performed before reduction, in which case the resulting certificate has no online conditions, but presumably a shorter validity period. The other option is to include some or all the online conditions in the CRC and let the verifier perform them as needed. However, there are problems in both approaches. It is not possible to perform all online validations in advance of usage. `CRL` and `Reval` can be performed in advance - their result is a validity period, which can be used to determine the validity period of the CRC. `One-time` and `limit`, on the other hand, have to be evaluated at the time of usage and therefore they have to be included in the CRC. Finally, due to the design of `limit`, it is not possible to perform a reduction over a certificate containing a `limit` condition, because that particular certificate has to be in the chain for the `limit` check to work.

A structural definition is required to include online tests from other certificates. The current SPKI structure does not define how CRCs are to be constructed, so the inclusion of online test from the other certificates is still undefined. Nevertheless, the size of the CRC with online checks will be rather large, as we have to include complete certificates. Because the instructions to the online servers can be included in the s-part of the original certificate, we have to include the whole certificate to convince the online server that the instructions really come from the issuer. Just including the relevant validity part will not suffice, as there is no signature authenticating the information. With these limitations in mind, the performance improvements achievable with CRCs containing validity conditions are still an open question. Further performance improvements could be achieved, if all the remaining online validations in a CRC could be replaced with a single online validation representing all of them. Naturally, this raises trust issues, but could provide significant improvements, particularly in situation with limited network access. However, the other type of a CRC should still be very useful in many situations. Particularly, if no online validations are left, the resulting CRC can be quite fast to evaluate.

4 Different Reducers

The SPKI structure draft only talks about the verifier creating CRCs, possibly also for the benefit of others, who choose to trust this verifier. However, any other certificate issuer in the chain can also issue partial reductions starting from themselves and ending at any point after them in the chain. The largest reduction naturally comes from the original issuer until the final user. The trust issues in all the cases, where the reduction issuer is already a member of the chain, are fairly clear. As they simply use their existing right to issue certificates, no new parties are introduced, which could change the trust model. However, the reduction issuer has to be additionally trusted to make correct reductions. If the reduction carries fewer rights than the original chain, the original chain can still be used to get to the remaining rights, but this might be inconvenient or even impossible thus mandating a new reduction. If, on the other hand, the reduction carries more rights (larger amount or other/larger rights), the reducer is doing this at its own expense - the original chain issuers can not be expected to take responsibility for this. Therefore, it always makes sense for the reduction issuer to keep a copy of the reduced chain so that any disputes can later be solved.

The cost of performing reductions are different for verifier and other issuers. The verifier would anyway have to check the chain and reduce it, so the additional effort of creating the CRC is not very large. The other issuers, however, would not normally evaluate the chain, so for them the additional effort is bigger. Therefore, not all issuers are likely to offer reduction services for arbitrary users. The verifier, on the other hand, should probably always issue a CRC (with the exception of chain without any remaining rights, naturally). The structure draft talks about other entities trusting the verifier for creating CRCs. This apparently implies that the verifier can act as a TTP creating reductions for others. If we accept TTPs, they would not necessarily even have to be verifiers; any suitable TTP could be used. But this changes the trust model of the system by introducing an outsider capable of creating certificates for anyone without limits (or at least the limits have to be much higher than for regular certificate issuers). Of course, an incorrectly acting TTP can be asked to justify the actions afterwards by presenting the original chains, but the TTP still creates a tempting target for attacks due to larger than normal rights.

5 Conclusions

We have discussed the role of certificate reduction certificates and the motivations for using them. Certificate chains are a product of the management process and should be viewed as such. We conclude that CRCs could provide performance improvements at minimal cost, if issued by the verifier. Finally, online validations still present challenges for reduction and should be further looked into.

Bibliography

[Aur98] T. Aura. Fast access control decisions from delegation certificate databases. In *Proceedings of 3rd Australasian Conference on Information Security and Privacy ACISP '98*, volume 1438 of *LNCS*, pages 284–295, Brisbane, Australia, July 1998. Springer Verlag.

[BD02] O. Bandmann and M. Dam. A Note on SPKI's Authorisation Syntax. In *Proceedings of 1st Annual PKI Research Workshop*, Maryland, USA, April 2002.

[EFL$^+$] C. Ellison, B. Franz, B. Lampson, R. Rivest, B. Thomas, and T. Ylnen. Simple Public Key Certificate. Internet draft (expired), IETF SPKI Working Group, July 1999.

[EFL$^+$99] C. Ellison, B. Franz, B. Lampson, R. Rivest, B. Thomas, and T. Ylnen. SPKI Certificate Theory. *Request for Comments: 2693*, September 1999.

[HK00] T. Hasu and Y. Kortesniemi. Implementing an SPKI Certificate Repository within the DNS. In *Poster Paper Collection of the Theory and Practice in Public Key Cryptography (PKC 2000)*, Melbourne, Australia, January 2000.

[KHS00] Y. Kortesniemi, T. Hasu, and J. Sars. A Revocation, Validation and Authentication Protocol for SPKI Based Delegation Systems. In *Proceedings of Network and Distributed System Security Symposium (NDSS'00)*, San Diego, USA, February 2000.

[NKP99] P. Nikander, Y. Kortesniemi, and J. Partanen. Preserving Privacy with Certificates in Distributed Delegation. In *Proceedings of 1999 International workshop on Practice and Theory in Public Key Cryptography*, Kamakura, Japan, March 1999.