Symmetric Key Infrastructure

Karel Masařík, Daniel Cvrček

xmasar01@stud.fit.vutbr.cz, cvrcek@fit.vutbr.cz

Faculty of Information Technology
Brno University of Technology

Abstract

The denouncement of some properties of key management systems based on X.509 is growing in several last years. This article briefly summarises problematic features of X.509 standard. It is followed by description of key management system based on symmetric key infrastructure. The proposed scheme does not reject public key cryptography completely but is able to use it for authentication purposes. This proposed system is able to fully replace X.509 key management systems in most application environments. The key management system is followed by a communication protocol allowing secure message exchange and it is also utilised for key management procedures. The scheme is currently being implemented so the design as introduced can not be treated as detailed and complete.

Keywords: X.509, key management, symmetric cryptography, diffie-hellman, logging, audit.

1 Introduction

It is crucial to return to the beginning and put questions why we are using key management systems based on X.509 (PKI from now on), while analysing problems related to PKI. The original idea – existence of a kind of yellow pages containing public key certificates turned out to be infeasible. There was an attempt for a printed register of certification authorities but it did not spread [x]. The main idea of PKI is to use trusted third party (replacement of yellow pages) for verification of principle‘s identity and its binding with a public key. Important presumption of X.509 approving was creation of world-wide system of unique names – distinguished names (DN). The original proposal did not expect any problems with DN ambiguity, because there was supposed existence of just one root certification authority. The same assumption eliminated problems with verification of certificates issued in different certification domains because there was just one such domain. There should be just one certification authority that is simply trustworthy.

Trustworthiness of a certificate is usually implied by certification policy that may be, but often is not part of each of the certificates. We make the certificate verification procedure simpler by not entering that attribute in the certificate, but it is necessary to set trustworthiness of each root certification authority manually. Certificates issued by commercial certification authorities contain only basic attributes (validity of verification key, purpose applicable on the certificate) that are easily verifiable. The operations with public keys and their certificates performed nowadays deny original ideas of PKI.

It is not surprising that there exist several attempts to change PKI or replace public key cryptography with symmetric one entirely. Let us take a look at the weakest link of PKI to define a set of mechanisms necessary for replacing it. The weakest point is represented by registration authorities and their clerks that verify certification requests. The only security mechanisms allowing control of the activity consist of paper records and copies of documents, i.e. paper logging.

The question arising in the given context is possibility to realise key management some other way. We can ask if it is possible to realise key management just with primitives of symmetric cryptography. A proposal from Christianson, Crispo, and Malcolm appeared in 2000 [14]. It contains mechanisms that are sufficient to realise some of functional properties of key management.

The most important limitation of symmetric mechanisms lies in the sharing of a key between two principals. This sharing disables non-repudiation property for messages exchanged between those two parties. We will solve the problem by routing all messages by several independent principals that in pairs share symmetric keys. Consistent logging of the communication passing through each subject allows not only to detect fraudulent behaviour but also find the originator of such behaviour.

2 PKI Properties

PKI is fully determined by a trusted third party (certification authority) that fully defines security properties of the key management system. This fact brings advantages implied by the existence of security domain and “full” trust inside the domain. On the other side, failure of TTP forbids to conduct even the basic functions of key management. If a certificate owner wanted to spread information about his certificate non-validity without TTP, he would not be even able to determine the set of subjects using his certificate.

PKI allows local verification of certificates/signatures without on-line access to certification authority. However, it is necessary to communicate with TTP (revocation authority) or to have access to the site with valid CRLs, when verifying certificates. Existing schemes using revocation authority violate principle stating that signer must be able to provide all information necessary for the signature verification. The on-line access to revocation authority may overload TTP’s servers.

Unique identification of certificate owner is the necessary condition for certificate issuance. We do not see a problem in the relationship between TTP and certificate owner, but between signer and signature verifier. Do you know exact information identifying your partner? Rivest and others proposed solution by name uniqueness in a certain context. This approach breaks importance of the domain defined by certification authority and creates new domains around users not respecting any borders.

The non-repudiation is the most important advantage of asymmetric cryptography and there is no such an elegant mechanism able to replace it.

Other disadvantages of PKI are described in many articles. The main problem is implementation of key-pair revocation when there is a problem of decentralisation introduced by independent usage of public key certificate. It is a typical example of temporary PKI implementations schizophrenia. They are trying to gain advantage from public key cryptography and on the other side they force users to perform all procedures with an on-line communication with TTP. It results in all existing disadvantages. There is a high load of TTP and it is not possible to initiate any procedure by TTP because the existing decentralisation does not allow TTP to know sets of users dependent on public keys.

3 Properties of symmetric key management

When we create a key management system based on symmetric cryptography, we are able to preserve most good properties of PKI and at the same time obtain a functionality not possible in PKI systems.

1. Revocation of symmetric keys is simple, because each owner of a key knows who else is using the shared key. This is sufficient to define a mechanism for direct notification. When using asymmetric keys it is necessary to act through TTP and hope that verifiers will be able to connect to the TTP to obtain information about key revocation. The whole procedure is much more complicated, indirect, slower, and with lower reliability – it is necessary to promote on-line revocation authority.
2. PKI (according to its motivation) allows local certificate verification. The praxis, however, demands on-line access to the TTP because accredited certification authorities must provide bullet-proof certificate verification in the moment of signature checking. Symmetric key management reduces the problem because a key is either valid or was explicitly declared as revoked.
3. Usage of asymmetric algorithms forbids anyone (including TTP) to masquerade for some other certificate owner (however, this is ensured only by administrative security and paper documents in the case of TTP!). The bad news is that there is no forward secrecy in the existing schemes and we feel it as a severe weakness. Backward secrecy is ensured only by TTPs through logging and audits against physical records. We are able to provide forward secrecy very easily in symmetric key management schemes.
4. Regarding non-repudiation, there is no easy solution. We propose a procedure based on a special communication scheme in the following paragraphs. The scheme assumes routing of each message by several intermediate nodes. Each message is secured with a key shared by sender and receiver and with keys shared by neighbour communication nodes. This combination allows unique determination of the message originator.
5. Asymmetric keys offer locality of trust unreachable by symmetric cryptography. Symmetric algorithms offer either relation of two principals or sharing of a key among more entities. The latter situation decreases probability to find originator of messages and overall security of the scheme.

PKI has definite advantage of easier authentication originated from the existence of public key. We are able to provide all other properties with symmetric key algorithms. The most difficult is non-repudiation that must be supported by communication logging by mutually untrusting (independent) parties. The parties may be represented by e.g. firewalls placed on borders of security domains.

When we agree on the assumption that key management is simpler with symmetric cryptographic algorithms we can implement most parts of key management with symmetric algorithms. Public key cryptography may be used where it is more convenient for authentication because of requirements on non-existence of physical contact or to minimise of KDC influence.

4 Design of Symmetric Key Management

We understand key management system as a set of domains with a binary relation. That binary relation represents existence of shared secret between pairs of principals. Each secret key may have a symmetric public key that can be publicly distributed. Public keys are used for verification of relation validity between two particular principals.

Domains (defined by the principals) are mutually identifiable by secret key and name of the principal - $\{H(K_{AX} \mid ID_A), ID_A\}$. The existing implementation of key management system uses authorisation server (AS) with functionality of KTC. There is created a shared secret between AS and a new domain when adding the new domain into the key management system. AS is able to remove a domain from the system to block its ability to send and forward messages. Strong role of the AS allows to spread information about exclusion of the domain to all other principals in the binary relation with the affected domain.

Each domain is able to operate as a communication point – mirror – functioning as a message transceiver. Messages consist from data with MAC value computed with the key shared between sender and receiver. The second part contains MAC computed by a key of neighbour principals. This MAC value changes as the message moves through net of principals. Each mirror securely logs the message, checks correctness of the MAC (created with a shared key it possesses) and generates a new second part of the message MACed with a key shared with the following mirror or final recipient.

4.1 Certificates

We create new shared secret between two domains, when adding a new domain into the scheme. In the most simple example, we can use the following certificates:

$$\{ID_A, ID_B, ID_{AS}, SS_{AB}\}_{Kas}$$

$$\{ID_A, ID_B, ID_{AS}, SS_{AB}\}_{Kbs}$$

where A, B are domains, AS is an authorisation server SS_{XY} is shared secret and K_{XY} are shared symmetric keys. That basic scheme allows AS to follow all communication between principals A and B. This negative effect can be partially eliminated with a more sophisticated scheme based on public key agreement scheme.

4.2 Key Agreement

Key agreement is the crucial process to be done during introduction of a new principal into the scheme. There are basically three ways how to do it:

1. physical contact with each principal I want to share a secret with – infeasible in many cases
2. use symmetric cryptography to make use of KDC or KTC to create a new share – that TTP (AS in our scheme) is able to decrypt all the communication between pairs of principals
3. use an asymmetric key agreement scheme while TTP is used to confirm identity of the so far mutually anonymous principals – TTP would have to mediate all communication relations to hide its fraudulent behaviour

We believe that the third option is the most appropriate. We can use the biggest advantage of asymmetric cryptography, when private keys do not have to be transmitted over unsecured channel.

4.2.1 DH Key Agreement Scheme

The basic protocol is very simple, let us assume that A and B are the principals to agree on a shared secret and AS is the authorisation server. There exists public modulus *N* for modular arithmetic operations and G as a generator.

A->B: G^{Xa} mod N

B->A: G^{Xb} mod N

First two messages may be exchanged directly between A and B. That is followed by secure exchange of information that can be derived from the created shared secret. We can use any secure protocol involving trusted third party. The following lines show Denning Sacco protocol (variation of the flawed Needham-Schroeder protocol) [16, pg. 47], where the hash (and second hash) of shared secret is used for the freshness property of the authentication.

A->S: $\{ID_A, ID_{B,} H^2(G^{XaXb}) \}$

S->A: $\{ID_B, K_{AB}, H^2(G^{XaXb}), \{ID_A, K_{AB}, H^2(G^{XbXa})\}K_{BS}\}K_{AS}$

A->B: $\{ID_A, K_{AB}, H^2(G^{XbXa})\}K_{BS}$

B->A: $\{H(G^{XaXb})\}K_{AB}$

Where K_{AS} and K_{BS} are actual shared keys between A – AS and B – AS, respectively. K_{AB} is a session key used just for this message exchange. G^{XbXa} is the shared secret, where mod N operation is assumed implicit.

The TTP has at least one additional task during this procedure. We need each of the principal to have a binary relation with several other principals. There are several schemes suitable for that data exchange. We can name e.g. …

TTP should ask appropriate principals to create new relations whenever necessary. This requirement may arise when a new principal is introduced into the scheme and also, when the number of principals/domains increases.

4.2.2 RSA encryption

A and B has generated an RSA key pair (PK, SK). There is also a key shared between principals A/B and AS. What follows is an outline of how the shared secret is

A->S: $\{ID_A, ID_B, PK_A, T\}K_{AS}$

S->B: $\{ID_A, ID_B, ID_{AS}, PK_{A,} T\}_{Kbs}$ – secret key certificate as described above

B->A: RSA$\{ID_B, ID_A, T, R_B, PK_B\}PK_A$

A->B: RSA$\{ID_A, ID_B, T, R_A\}PK_B$

A, B: $H(R_A)$ **xor** $H(R_B)$

This is just an outline of what should be exchanged by the protocol. T is a freshness information and RSA{X}K is an RSA operation (encryption of signing) with key K on data X. **xor** is bitwise operation on two binary numbers.

4.3 Message Transmission

Security of message transmission is based on shared secrets. Let us assume that A and B are sender and receiver of a message respectively, and M_1, … M_n are mirrors that provide message forwarding. $SS_{XY,i}$ is an i-th secret shared between principals X and Y. $K_{XY,i}$ is i-th symmetric key between principals X and Y. Let H be a cryptographic MAC function and S_M be keyed hash of a message M.

The first step toward sending a message is generation of a new shared key and a new shared secret.

$$K_{AB,i}=H(0 \mid SS_{AB,i}),\ SS_{AB,i+1}=H(1 \mid SS_{AB,i})$$

$$K_{AM1,i}=H(0 \mid SS_{AM1,i}),\ SS_{AM1,i+1}=H(1 \mid SS_{AM1,i})$$

This procedure of shared secret generation is used for message authentication between already known principals. The necessary assumption is the existence of secure keyed hash function and a secure random number generator.

We can now format a new message. The first part is destined for the receiver, the second part for the neighbour communication point M_1.

H_0=($SS_{AB,i}$ | h($SS_{AB,i+1}$) | S_M) and the message encrypted by key $K_{AB,i}$ - M

H_1=($SS_{AM1,i}$ | H($SS_{AM1,i+1}$) | S_M)

Each mirror M_i forwarding the message during its transmission verifies H_i, logs the message (together with H_0) and generates a new MAC

H_{i+1}=($SS_{MiMi+1,i}$ | H($SS_{MiMi+1,i+1}$) | S_M)

The whole message {M, H_0, H_{i+1}} is sent to the next mirror/receiver. If an error is detected during verification of the hash, the problem is reported to authorisation server that tries to detect originator of the incorrect data. The recipient is able to verify both MACs H_0 and H_{i+1}. When a problem is detected (the results do not equal), it is again reported to AS.

The index of the shared secret is possible to increment after successful acknowledge of the message delivery. The described mechanism allows us to bind subsequent messages. The data produced during such a message exchange is possible to use for identification of a cheating principal/domain.

4.4 Logging

Trustworthiness of the scheme depends on secure logging of the traffic on all principals/domains. It must not be possible to change order of log entries or change their content. We use the following structure of the log entries to ensure the mentioned requirement.

- Message ID – allows its identification in the system and it is therefore prime key of the record.
- Type of the message
- Recipient and receiver ID
- ID of the neighbour that sent the message and that the message will be sent to.
- Two MACs of the data with keys K_{AB}, $K_{Mi-1,i}$
- MAC of the message path trace
- Time – if applicable, or ordinary number of the record
- Cumulative hash of previous log records

The most important is the cumulative hash that restricts manipulations with logs pretty well.

5 Trustworthy Hardware

Exploitation of symmetric cryptographic algorithms is vulnerable regarding storage and operation with secret keys and shares. We hope the scheme to be as simple as possible to allow implementation of crucial operations into cryptographic smart cards. Implementation of the scheme, where each principal uses is trustworthy hardware module will be fulfilled next year as the second phase of the student project being solved.

The main purpose of hardware tokens will be secure logging of processed and forwarded communication traffic and co-operation with authorisation server, when detected illegal behaviour of a principal in the scheme.

6 Conclusion

We propose a scheme for key management and message exchange that uses symmetric cryptography to the maximum extent while preserving properties usually available seen in public key infrastructures. The system that is actually being implemented according to the ideas described should offer simpler communication between principals and sufficiently powerful mechanisms for detection of problems and identification of cheating principals.

The appropriateness of the proposed scheme is dependable on application requirements. The current proposal tries to minimise power of TTP. It can be seen in the key agreement scheme and in the design of improper behaviour detection. The role of the authorisation server is supposed to be primarily control.

References

[1] David A. Cooper: *A Model of Certificate Revocation*. Proceedings of the Fifteenth Annual Computer Security Applications Conference, pg. 256-264, December 1999.

[2] David A. Cooper: *A More Efficient Use of Delta-CRLs*. Proceedings of the 2000 IEEE Symposium on Security and Privacy, pgs. 190-202, May 2000.

[3] M. Naor, K. Nissim: *Certificate Revocation and Certificate Update*. Proceedings 7th {USENIX} Security Symposium (San Antonio, Texas), January 1998.

[4] R. N. Wright, P.D. Lincoln, J.K. Millen, A.I. Lincoln: *Efficient Fault-Tolerant Certificate Revocation*. ACM Conference on Computer and Communications Security, pgs. 19-24, 2000.

[5] A. Arnes, S. J. Knapslog: *Selecting Revocation Solutions for PKI*. NORDSEC 2000, Reykjavik, Iceland, 2000.

[6] R. L. Rivest: *Can We Eliminate Certificate Revocation Lists?*. Financial Cryptography, pgs. 178-183, 1998.

[7] C.A. Gunter, T. Jim: *Generalized Certificate Revocation*. Symposium on Principles of Programming Languages, pgs. 316-329, 2000.

[8] A. Buldas, P. Laud, and H. Lipmaa: *Elliminating counterevidence with applications to accountable management*. Jounal of Computer Security (2002). To appear.

[9] ITU-T. *Draft revised ITU-T Recommendation X.509 (v4)*. 2000

[10] Roger Clark: *Conventional Public Key Infrastructure: An Artefact Ill-fitted to the Needs of the Information Society*. submitted to the Euro. Conference in Information Systems 2001, Bled, Slovenia.

[11] Ellison C., Schneier B.: *Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure*. Computer Security Journal, vol. XVI, November, 2000

[12] Bellare M., Miner S.: *A forward secure digital signature scheme*. Advances in Cryptology - Crypto99 Proceedings, LNCS 1666, Springer-Verlag, 1999.

[13] J.Lin, S. Kent, D. Balenson, B. Kalinski: *Privacy Enhancement for Internet Electronic Mail: Parts I-IV.* RFC1421-1424, 1993.

[14] Christianson B., Crispo B., and Malcolm J.A.: *Public-Key Crypto-systems Using Symmetric-Key Crypto-algorithms*. Security Protocols, 8th International Workshops Cambridge, UK, April 3-5, 2000.

[15] Cvrcek D., *Real World Problems of PKI Hierarch*: Security and Protection of Information 2001, Brno, 2001.

[16] Clark J., Jacob J.: A Survey of Authentication Protocol Literature: Version 1.0, November 1997, http://citeseer.nj.nec.com/clark97survey.html

Information about authors

Karel Masarik is attending 4th year of M.Sc. studies at Faculty of Information Technology, Brno University of Technology. He is working on the project of implementation of key management system.

Daniel Cvrcek born in 1974. He graduated Faculty of Electrotechnics and Informatics, Brno University of Technology by acquiring Ph.D. in the area of authorisation model for large information systems in 2001. Main research interests include security of smart-cards and key management – public key infrastructures.