A Modular Approach to Key Distribution

Walter Fumy and Michael Munzert
Siemens AG, Germany

Abstract. *The purpose of key management is to provide procedures for handling cryptographic keying material to be used in symmetric or asymmetric cryptographic mechanisms. As a result of varied design decisions appropriate to different conditions, a large variety of key distribution protocols exist. There is a need to explicate key distribution protocols in a way that allows to understand which results they achieve and on which assumptions they depend. We define a modular system that can be used to transform cryptographic protocols into a generic form and that has proven to be useful in the analysis and the construction of such protocols.*

1. Introduction

The purpose of key management is to provide procedures for handling cryptographic keying material to be used in symmetric or asymmetric cryptographic mechanisms. Key management includes user registration, key generation, key distribution, key storage, and key deletion. A fundamental problem of key management is to establish keying material whose origin, integrity, and - in the case of secret keys - confidentiality can be guaranteed. Most of the important properties of key management protocols do not depend on the underlying cryptographic algorithms, but rather on the structure of the messages exchanged. Bugs in such protocols therefore usually do not come from weak cryptographic algorithms, but from mistakes in higher levels of the design.

A large variety of mechanisms for key distribution and especially for key agreement can be found in the literature. [Diff 76] and [Ruep 88] e.g. describe procedures which allow the establishment of a common secret key for two users and which only require the communication of public messages. [Okam 86] proposes similar schemes that utilize each user's identification information to authenticate the exchanged messages. [Günt 89] and [Baus 89] use data exchanged during an authentication protocol to construct a session key. [Koya 87] shows how to generate a common secret conference key for two or more users that are connected in a ring, a complete graph, or a star network.

Key management has also been addressed by different standardization bodies which led to several national and international standards, in particular in the area of banking [ANSI 85], [CCIT 87]. Standards for other application areas as well as base standards

A.J. Menezes and S.A. Vanstone (Eds.): Advances in Cryptology - CRYPTO '90, LNCS 537, pp. 274-283, 1991.

dealing with generic issues of key management can be expected to follow [ANSI 89], [ISO 90c], [ISO 90d]. Until now, standardization bodies focus on key distribution in contrast to key agreement.

Key management schemes generally depend on the type of keys to be distributed, on the given facilities (e.g. the properties of the specific environment) and on the specific application. The most important considerations are the threats to be protected against, and the physical and architectural structure of the system.

This paper addresses the problem of key distribution by an approach that is modular, open, and generic, such that additional requirements, building blocks, and composition rules can be added if desired. It allows the construction of a large variety of individual key distribution schemes with desired properties. Such a modular approach is advocated within IEC/ISO/JTC1/SC27 for its standards on key management currently under development [ISO 90c], [ISO 90d]. The following section will describe generic security requirements for key distribution protocols. Section 3 gives some examples for building blocks and composition rules. Section 4 finally shows the analysis and construction of typical key distribution mechanisms.

2. Generic Requirements for Key Distribution

The fundamental problem of key distribution is to establish keying material to be used in symmetric or asymmetric cryptographic mechanisms whose origin, integrity, and - in the case of secret keys - confidentiality can be guaranteed. A key may be distributed either manually or automatically. Manually distributed keys are exchanged between parties by use of a courier or some other physical means. When using only symmetric cryptographic techniques at least the first key has to be manually exchanged between two parties in order to allow secure communications.

An automatic distribution of keys typically employs different types of messages. A transaction usually is initiated by requesting a key from some central facility (e.g. a Key Distribution Centre), or from the entity a key is to be exchanged with. Cryptographic Service Messages (CSMs) are exchanged between communicating parties for the transmission of keying material, or for authentication purposes. CSMs may contain keys, or other keying material, such as the distinguished names of entities, key identities, count or random values. CSMs have to be protected depending on their contents and on the security requirements which for their part depend on several parameters, and especially on the given facilities. Generic requirements include:

(a) Data Confidentiality: Secret keys and possibly other data are to be kept confidential while being transmitted or stored.

(b) Modification Detection is to counter the active threat of unauthorized modification of data items. In most environments cryptographic service messages have to be protected against modification.

(c) Replay Detection / Timeliness: Replay detection is to counter unauthorized duplication of data items. Timeliness requires that the response to a challenge message is prompt and does not allow for play-back of some authentic response message by an impersonator.

(d) Authentication is to corroborate that an entity is the one claimed. This is part of data origin authentication (see below). The general problem of authentication is to establish a message whose origin, uniqueness and timeliness can be verified.

(e) Data Origin Authentication is to corroborate that the source of a message is the one claimed. As defined in [ISO 88] it does not provide protection against duplication or modification of the message. In practice, however, data origin authentication often is a combination of sender authentication and modification detection.

(f) Proof of Delivery shows the sender of a message that the message has been received by its legitimate receiver correctly.

3. Building Blocks and Composition Rules

As a result of varied design decisions appropriate to different circumstances, a large variety of key distribution protocols exist (see e.g. [Mill 87], [Need 78], [Otwa 87]). Therefore, there is a need to explicate key distribution protocols in a way that allows to understand which results the different protocols achieve and on which assumptions they depend. We define a modular system that can be used to transform cryptographic protocols into a generic form and that has proven to be useful in the analysis and the construction of such protocols. Formal methods devoted to the analysis of (authentication) protocols have recently been developed by M.Burrows, M.Abadi and R.Needham [Burr 90].

The basic elements of cryptographic service messages can be classified in several ways. There are building blocks that append data to a message which is independent of the message itself, building blocks that append data which is dependent on the message, and building blocks that transform a message in a certain way. Note that in the first case the original message may be empty. Each of the building blocks described below addresses one or more of the requirements given in section 2.

(a) Encipherment: The confidentiality of a data item D can be ensured by enciphering D with an appropriate key K. Depending on whether a secret key algorithm or a public key algorithm is used for the enciphering process, D will be enciphered with a secret

key K shared between the sender and the legitimate recipient of the message (building block a1), or with the legitimate recipient B's public key K_{Bp} (a2). Encipherment with the sender A's private key K_{As} provides a digital signature which may be used to authenticate the origin of data item D, or to identify A (a3). Encipherment with a secret key (a1, a3) provides modification detection if B has some means to check the validity of D (e.g. if B knows D beforehand, or if D contains suitable redundancy).

generic:		**eK(D)**
(a1)	A → B:	$eK_{AB}(D)$
(a2)	X → B:	$eK_{Bp}(D)$
(a3)	A → X:	$eK_{As}(D)$

(b) Modification Detection Codes: To detect a modification of a data item D one can add some redundancy that has to be calculated using a collision-free function, i.e. it must be computationally infeasible to find two different values of D that render the same result. Moreover, this process has to involve a secret parameter K in order to prevent forgery. Appropriate combination of K and D also allows for data origin authentication. Examples of suitable building blocks are message authentication codes as defined in [ISO 89] (see b1), or hash-functions, often combined with encipherment (b2 through b5).

generic:		**D || mdcK(D)**
(b1)	A → B:	$D \,\|\|\, macK_{AB}(D)$
(b2)	A → B:	$D \,\|\|\, eK_{AB}(h(D))$
(b3)	A → X:	$D \,\|\|\, eK_{As}(h(D))$
(b4)	A → B:	$D \,\|\|\, h(K_{AB} \,\|\|\, D)$
(b5)	A → B:	$eK_{AB}(D \,\|\|\, h(D))$

These building blocks enable the legitimate recipient to detect unauthorized modification of the transmitted data immediately after receipt. The correctness of distributed keying material can also be checked if the sender confirms his knowledge of the key in a second step (see section (d) below).

(c) Replay Detection Codes: To detect the replay of a message and to check its timeliness, some explicit or implicit challenge and response mechanism has to be used, since the recipient has to be able to decide on the acceptance. This paragraph only deals with implicit mechanisms; explicit challenge and response mechanisms are being dealt with in section (d) (see below). In most applications the inclusion of a replay detection code (e.g. a timestamp TD, a counter CT, or a random number R) will only make sense if it is protected by modification detection. If modification detection of the data item D is required, the concatenation of D and the rdc also has to be protected against separation.

(c1)	$D := D \,\|\|\, rdc$	$rdc \in \{CT, TD\}$

With symmetric cryptographic mechanisms key modification can be used to detect the replay of a message. Building block c2 combines (e.g. XORs) the secret key with an rdc (e.g. a counter CT, or a random number R). Key offsetting used to protect data enciphered for distribution is a special case of building block c2. In this process the key used for encipherment is XORed with a count value [ANSI 85].

(c2) $K_{AB} := f(K_{AB},rdc)$ $rdc \in \{R, CT\}$

(d) Proof of Knowledge of a Key: Authentication can be implemented by showing knowledge of a secret (e.g. a secret key). Nevertheless, a building block that proves the knowledge of a key K can also be useful, when K is public. There are several ways for A to prove to B the knowledge of a key that are all based on the principle of challenge and response in order to prevent a replay attack. Depending on the challenge which may be a data item in cleartext or in ciphertext, A has to process the key K and the rdc in an appropriate way (e.g. by encipherment (see d1), or by calculating a message authentication code (see d2)), or A has to perform a deciphering operation (d4).

The challenge may explicitly be provided by B (e.g. a random number R) or implicitly be given by a synchronized parameter (e.g. a timestamp TD, or a counter CT). For some building blocks the latter case requires only one pass to proof knowledge of K; its tradeoff is the necessary synchronization. If B provides a challenge enciphered with a key K*, A has to apply the corresponding deciphering key K. In these cases the enciphered data item has to be unpredictable (e.g. a random number R, or a key K**).

generic:		**authK(A to B)**	
(d1)	A ← B:	rdc	obsolete if rdc ∈ {CT, TD}
	A → B:	eK(rdc)	
(d2)	A ← B:	rdc	obsolete if rdc ∈ {CT, TD}
	A → B:	rdc || macK(rdc)	
(d3)	A ← B:	rdc	obsolete if rdc ∈ {CT, TD}
	A → B:	rdc || h(K || rdc)	
(d4)	A ← B:	eK*(rdc)	rdc random value
	A → B:	rdc	
(d5)	A ← B:	eK*(K**) || rdc	rdc arbitrary
	A → B:	eK**(rdc)	
(d6)	A ← B:	eK*(K**) || rdc	rdc arbitrary
	A → B:	macK**(rdc)	
(d7)	A ← B:	eK*(K**) || rdc	rdc arbitrary
	A → B:	h(K** || rdc)	

Building blocks d5 through d7 of course also confirm that A knows K** in addition to the deciphering key K.

(e) **Composition Rules**: Some composition rules extracted from the description of the above building blocks and their effects on different security requirements are summarized in table 1 below.

Building Block	Requirement			
	Confidentiality	**Modification Detection**	**Replay Detection**	**Data Origin Authentication**
A → B: $eK_{AB}(D)$	yes	only if recipient can check validity of D	D := D || rdc	only if recipient can check validity of D
X → B: $eK_{Bp}(D)$	yes	no	D := D || rdc	no
A → X: $eK_{As}(D)$	no	only if recipient can check validity of D	D := D || rdc	only if recipient can check validity of D
A → B: D || $macK_{AB}(D)$	no	yes	D := D || rdc *or* K_{AB} := f(K_{AB},rdc)	yes
A → B: D || $eK_{AB}(h(D))$	no	yes	D := D || rdc *or* K_{AB} := f(K_{AB},rdc)	yes
A → X: D || $eK_{As}(h(D))$	no	yes	D := D || rdc	yes
A → B: D || $h(K_{AB}$ || D)	no	yes	D := D || rdc *or* K_{AB} := f(K_{AB},rdc)	yes
A → B: eK_{AB}(D || h(D))	yes	yes	D := D || rdc *or* K_{AB} := f(K_{AB},rdc)	yes

The building blocks (cx) have to be applied first, so that the combination of (c1) with (a1) e.g. results in

$$A \rightarrow B: \qquad eK_{AB}(D \,||\, rdc)$$

Besides the above composition rules one also can give several general rules:

(e1) A secret key shall not be used for both encipherment and authentication of the same data item.

(e2) Two consecutive transmissions from A to B may be replaced by one transmission of concatenated messages.

(e3) D1 || D2 may be replaced by D2 || D1.

(e4) eK(D1) || eK(D2) may be replaced by eK(D1 || D2).

4. Examples for Point-to-Point Key Distribution Protocols

The basic mechanism of every key distribution scheme is point-to-point key distribution. If based on symmetric cryptographic techniques point-to-point key distribution requires that the two parties involved already share a key that can be used to protect the keying material to be distributed. In this section we discuss protocols for point-to-point distribution of a secret key that are derived from [ISO 90a], [ANSI 85], and [ISO 90b], respectively. The first example is to illustrate the construction of a key distribution protocol out of building blocks. Generic descriptions are used to exhibit similarities and differences between the discussed protocols.

General assumptions are:

- The initiator A is able to generate or otherwise acquire a secret key K*.
- Security requirements are confidentiality of K*, modification and replay detection, mutual authentication of A and B, and a proof of delivery for K*.

For point-to-point key distribution protocols based on symmetric cryptographic techniques we additionally assume:

- A key K_{AB} is already shared by A and B.

The first two security requirements can be met by an appropriate combination of building blocks a1 through c2 (see table 1), whereas for the other two requirements one can choose from building blocks d1 through d7. As a first example we show a protocol built up from building blocks a1 and c1 (step 1: confidentiality of K*, modification and replay detection), d1 and d5 (steps 2 to 4: mutual proof of knowledge of K_{AB} that includes B's proof of knowledge of K*), and d4 (steps 5 and 6: A's proof of knowledge of K*).

A	Protocol 1: Point-to-Point Key Distribution	B
(1) →	$eK_{AB}(K^* \| rdc_1)$	
(2) →	$eK_{AB}(rdc_2)$	
(3) →	$eK_{AB}(K^*) \| rdc_3$	
	$eK^*(rdc_3)$	← (4)
	$eK^*(rdc_4)$	← (5)
(6) →	rdc_4	

The above protocol can be greatly simplified by identifying the parameters rdc_2 and rdc_3 with rdc_1 (= N which can be chosen to be a counter CT or a timestamp TD), and by

applying composition rules e2 and e4 to steps 4 and 5. The resulting point-to-point key distribution protocol is one proposed in [ISO 90c].

A	Protocol 1': Point-to-Point Key Distribution (ISO/IEC CD 9798-2)	B
(1) →	$eK_{AB}(K^* \| N)$	
	$eK^*(N \| R)$	← (2)
(3) →	R	

In generic form this protocol can be desribed as follows:

A	Protocol 1': Generic Form	B
(1) →	eK(K* || rdc)	
(2) →	authK*(A to B)	
	authK*(B to A)	← (3)

A point-to-point key distribution protocol proposed in [ANSI 85] takes a somewhat different approach. To achieve replay and modification detection protocol 2 (see below) makes use of building blocks c1 and b1 (see also table 1). A proves to B its knowledge of K* (and thus the knowledge of K_{AB}) using building block d2, whereas B proves its knowledge of K_{AB} with building block d6 which also confirms the correct receipt of K*.

A	Protocol 2: Point-to-Point Key Distribution (ANSI X9.17)	B
(1) →	$D \| eK_{AB}(K^*) \| N \| macK^*(...)$	
	macK*(D)	← (2)

The generic form of protocol 2 exhibits the essential differences between the two protocols.

A	Protocol 2: Generic Form	B
(1) →	eK(K*) || rdc || mdcK*(...)	
	authK*(B to A)	← (2)

Finally we give an example for a point-to-point key distribution protocol based on public key techniques. We make the following supplementary assumptions:

- There is no shared key known to A and B before the key exchange process starts.

- There is a trusted third party C, where A can receive a certificate that contains the distinguished names of A and C, A's public key K_{Ap}, and the certificate's expiration date TE. The integrity of the certificate is protected by C's signature. As an example A's certificate is shown below:

$$ID_C \parallel ID_A \parallel K_{Ap} \parallel TE \parallel eK_{Cs}(h(ID_C \parallel ID_A \parallel K_{Ap} \parallel TE))$$

The exchange of certificates can be performed off-line and is not shown in the following protocol. In this protocol A sends a message (often refered to as *token*) to B that consists of a secret key K* enciphered with B's public key (building block a2) and an appended rdc. The integrity of the token is protected by A's signature (building block b3 combined with c1). This guarantees modification and replay detection, as well as data origin authentication. B responds with the enciphered rdc thereby acknowledging that it has received the key K* (building block d3).

A	Protocol 3: Point-to-Point Key Distribution (ISO/IEC CD 9798-3)	B
(1) →	$eK_{Bp}(K^*) \parallel rdc \parallel eK_{As}(h(eK_{Bp}(K^*) \parallel rdc))$	
	$eK^*(rdc)$	← (2)

The generic form of protocol 3 shows its similarity with protocol 2.

A	Protocol 3: Generic Form	B
(1) →	**eK(K*) || rdc || mdcK'(...)**	
	authK*(B to A)	← (2)

Acknowledgements

We would like to thank Thomas Berson and Peter Landrock for helpful suggestions and stimulating discussions on the subject.

References:

[ANSI 85] ANSI X9.17-1985: *Financial Institution Key Management (Wholesale)*, 1985.

[ANSI 89] ANSI X12.42-198x: *EDI Security Structures and Cryptographic Service Message Transaction Set*, 1989.

[Baus 89] Bauspieß, F.; Knobloch, H.-J.: "How to Keep Authenticity Alive in a Computer Network", Proceedings of Eurocrypt'89, Springer LNCS **434** (1990), 38-46.

[Burr 90] Burrows, M.; Abadi, M.; Needham, R.: *A Logic of Authentication*, DEC System Research Center Report **39**, 1990.

[CCIT 87] CCITT Draft Recommendation X.509: *The Directory - Authentication Framework*, 1987.

[Diff 76] Diffie, W.; Hellman, M.E.: "New Directions in Cryptography", IEEE Transactions on Information Theory, **22** (1976), 644-654.

[Günt 89] Günther, Ch.G.: "An Identity-Based Key-Exchange Protocol", Proceedings of Eurocrypt'89, Springer LNCS **434** (1990), 29-37.

[ISO 88] ISO 7498-2: *Open Systems Interconnections - Part 2: Security Architecture*, 1988.

[ISO 89] IEC/ISO 9797: *Data Integrity Mechanism Using a Cryptographic Check Function Employing a Block Cipher Algorithm*, 1989.

[ISO 90a] IEC/ISO Committee Draft 9798-2: *Entity Authentication Mechanisms - Part 2: Entity Authentication Using Symmetric Techniques, 1990.*

[ISO 90b] IEC/ISO Committee Draft 9798-3: *Entity Authentication Mechanisms - Part 3: Entity Authentication Using a Public-Key Algorithm*, 1990.

[ISO 90c] IEC/ISO/JTC1/SC27/WG2 Working Draft: *Key Management Part 2: Key Management Using Symmetric Cryptographic Techniques*, 1990.

[ISO 90d] IEC/ISO/JTC1/SC27/WG2 Working Draft: *Key Management Part 3: Key Management Using Public Key Techniques*, 1990.

[Koya 87] Koyama K.; Ohta, K.: "Identity-Based Conference Key Distribution Systems", Proceedings of Crypto'87, Springer LNCS **293** (1988), 175-184.

[Mill 87] Miller, S.P.; Neuman, C.; Schiller, J.I.; Saltzer, J.H.: *Kerberos Authentication and Authorization System*, Project Athena Technical Plan, MIT, 1987.

[Need 78] Needham, R.M.; Schroeder, M.D.: "Using Encryption for Authentication in Large Networks of Computers", Communications of the ACM, **21** (1978), 993-999.

[Okam 86] Okamoto, E.: "Proposal for Identity-Based Key Distribution Systems", Electronic Letters, **22** (1986), 1283-1284.

[Otwa 87] Otway, D.; Rees, O.: "Efficient and Timely Mutual Authentication", Operating Systems Review, **21** (1987), 8-10.

[Ruep 88] Rueppel, R.A.: "Key Agreements Based on Function Composition", Proceedings of Eurocrypt'88, Springer LNCS **330** (1988), 3-10.