A Proposal for Authenticated Key Recovery System [1]

Tsuyoshi Nishioka[a], Kanta Matsuura[a], Yuliang Zheng[b,c], and Hideki Imai[b]

[a]Information & Communication Business Div.
ADVANCE Co., Ltd.
5-7 Nihombashi Kobuna-cho
Chuo-ku, Tokyo, 103 Japan
Phone: +81-3-3668-5601 Fax: +81-3-3664-4387
E-mail: nishi@advance.co.jp

[b]Institute of Industrial Science
University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo, 106 Japan
Phone & Fax: +81-3-3402-7365
E-mail: Kanta@iis.u-tokyo.ac.jp

[c]The Peninsula School of Computing and Information Technology
Monash University
Australia

[1] Presented at the 1997 Joint Workshop on Information Security and Cryptology (JW-ISC'97), pp.19-30, October 26-28, 1997, Seoul, Korea.

Abstract

Information-security technologies are often implemented with redundant data fields attached to message containers. In a Key Recovery System that employs public key cryptography, a data recovery field (DRF) attached to a message typically contains a session key encrypted with a Key Recovery Agent's public key. At a later time when needs arise, the session key can be retrieved from DRF and used to recover the original message without the involvement of the message originator or recipient. Two problems with such a system are (1) that DRF is at least as long as the public modulo of the Key Recovery Agent, which represents an increasingly large communication overhead, (2) and that DRF is not created in an authenticated way, which opens a door for an originator to create a bogus DRF and deny his/her act at a later time. The main purposes of this paper are to address the two problems and exhibit a possible solution to them. In particular, we propose an authenticated Key Recovery System using a recently discovered signcryption primitive that combines the functions of digital signature with those of public-key encryption. We also carry out a detailed comparison between our proposal and existing Key Recovery Systems.

I Introduction

Key Recovery System (KRS)[2]–[11] is currently one of the most actively studied security issues, originated from national or legal points of view. For users, too, KRS might probably be of great help when they lose their private keys; any entity might make use of KRS.

Typically, as surveyed in Section II, KRS is implemented with an overhead called Data Recovery Field (DRF), which contains a session key encrypted with a public key of a Key Recovery Agent (KRA). Two problems with this type of KRS are (1) that DRF is at least as long as the public modulo of the Key Recovery Agent, (2) and that DRF is not created in an authenticated way. The first problem represents an increasingly large communication overhead, and seems unsolvable with public-key encryption primitives. The second problem opens a door for an originator to create a bogus DRF and deny his/her act at a later time. One might insist that this problem can be removed if the originator digitally signs a session key prior to encrypting it to generate a DRF. However, such a solution not only increases computational load on the message originator, but also consumes significantly larger communication bandwidth.

Therefore it is both of practical and theoretical importance to search for an efficient method for composing an authenticated DRF. This paper reports progress we have made in this line of research. In particular, Section II first overviews conventional KRS and addresses an importance of the authenticity of DRF. Section III.A then reviews a cryptographic primitive called "Signcryption", which has been recently developed in [1]. Signcryption is subsequently adapted to KRS in Section III.B, followed by a discussion on the efficiency in Section IV.

II Key Recovery System

KRS is a scheme in which a specified third party, called "data recovery agency (DRA)," is able to recover plain data from a ciphertext without a help of an originator or a recipient. Various KRSs have been proposed recently[2]–[11].

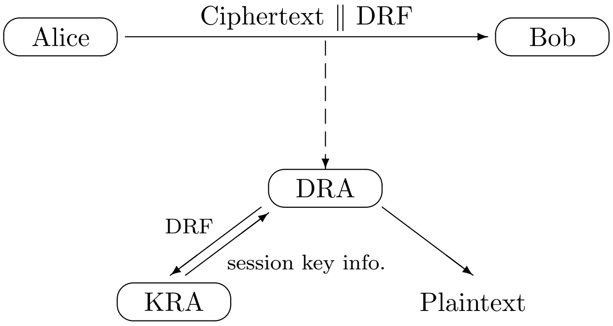

Fig.1: Key Recovery System with DRF

Each of those conventional KRSs considers four entities[2]: user A (an originator, say, Alice), user B (a recipient, say, Bob), a key recovery agent (KRA), and DRA (Fig.1).

The essence of KRS can be considered as key distribution from Alice to KRA. For this purpose, Alice attaches to a ciphertext an additional data field called Data Recovery Field (DRF), as is shown in Fig.2. DRF contains information of a session key with which Alice encrypts the plain data to generate the ciphertext.

When recovery of a particular data is required, DRA, who has the corresponding ciphertext and the DRF, requests KRA to recover the session key from the DRF. KRA recovers the session key and sends it back to DRA, if his request is successfully verified and accepted. It should be noted that KRA does not know the content of the data and that DRA cannot recover the data by himself. The condition for successful recovery includes the correctness of DRF; if Alice prepares not a correct DRF but a bogus one, KRS does not work.

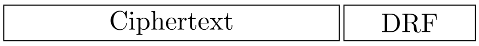

Fig.2: Data Recovery Field (DRF)

In some KRSs[4, 7, 9], in order to avoid a bogus DRF by verifying the DRF, Bob also computes DRF and checks whether it is identical to that sent from Alice. Such a solution is, however, inefficient; KRS imposes significant computational cost both on Alice and on Bob. The longer the size of keys, the less tolerable the cost paid for public-key cryptography. In addition, this type of KRS cannot completely avoid bogus DRFs because there still exist feasible attacks, for example, the use of a crypto-channel in a deeper layer.

We, then, propose an "authenticated DRF" by using an originator's signature; this signature is verified not by a recipient, but by KRA (Fig.3). In general, Alice's signature on DRF prevents a third party (including Bob, DRA, KRA) from forging the DRF. Thus the authenticated DRF discourages a malicious entity to generate a bogus DRF and is, moreover, useful for verification and identification of the responsibility for the DRF contents.

One might be afraid that the authenticated DRF also imposes increasingly larger cost both on computation and on communication. The following sections will show that the cost can be significantly reduced by employing a new cryptographic primitive called Signcryption.

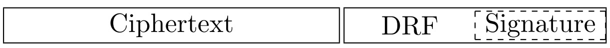

Fig.3: Authenticated DRF

III Key Recovery System based on Signcryption

III.A Signcryption

When both signature and public-key encryption are required, Signcryption provides us a more efficient scheme than conventional schemes where a message is firstly signed and then encrypted ("signature then encryption" schemes). In this section, we briefly review an implementation of Signcryption called SCS1 [1].

SCS1 is based on Diffie-Hellman key management and meta-ElGamal signature. We prepare a sufficient large prime p, a large subprime q of $p-1$ and g in $[1,\ldots,p-1]$ which is order q modulo p. All entities know these parameters as public values. They also know a public key-ed hash function KH and public encryption and decryption algorithms of a private key cipher (E,D). Each entity has his/her own public/secret-key pair; Alice has $x_A(\in Z_q^*)$ as her secret key and $y_A(=g^{x_A} \bmod p)$ as her public key. Bob has x_B as his secret key and $y_B(=g^{x_B} \bmod p)$ as his public key.

Alice signcrypts a message m for Bob as follows:

1. She picks x randomly from $[1,\ldots,q]$, and let $k = hash(y_B^x \bmod p)$. She ,then, splits k into k_1 and k_2 of appropriate length.

2. $r = KH_{k_2}(m)$.

3. $s = x/(r + x_A) \bmod q$.

4. $c = E_{k_1}(m)$.

5. She sends to him the signcrypted text (c,r,s).

On receiving (c,r,s) from Alice, Bob unsigncrypts it as follows:

1. He recovers k from r,s,g,p,y_A and x_B:
   $k = hash((y_A \cdot g^r)^{s \cdot x_B} \bmod p)$.

2. He splits k into k_1 and k_2.

3. $m = D_{k_1}(c)$.

4. He accepts m as a valid message originated from Alice if and only if $KH_{k_2}(m)$ is identical to r.

We will apply this implementation of signcryption to KRS in the next section. It should be noted that signcryption will be used not in generating a ciphertext but in constructing a DRF; the proposed scheme will not be devoted to any specific cryptographic algorithms used by Alice in constructing a ciphertext.

III.B Key Recovery System based on Signcryption

KRS might be regarded as confidential communication between an originator, Alice, and a KRA. The content of DRF is nothing but the session key encrypted with KRA's public key $y_K(=g^{x_K} \bmod p$ where x_K is KRA's secret key). When authenticated, DRF additionally includes Alice's signature. Signcryption allows Alice to carry out the public-key encryption and the signature generation in a logically single step as follows.

Given the session key k_{session}, Alice first picks x randomly from $[1,2,\cdots,q]$ and then generates the following DRF:

$$\text{DRF} = (r_K, s_K, c_K), \tag{III.1}$$

where

$$\begin{aligned} r_K &= KH_{k_2}(k_{\text{session}}), \quad &\text{(III.2)} \\ s_K &= x/(r_K + x_A) \bmod q, \quad &\text{(III.3)} \\ c_K &= E_{k_1}(k_{\text{session}}), \quad &\text{(III.4)} \\ k_1 \| k_2 &= hash(y_K^x \bmod p). \quad &\text{(III.5)} \end{aligned}$$

Computational cost
EXP=1, MUL=0, DIV=1, ADD=1, HASH=2, ENC=1

where
EXP = the number of modulo exponentiations,
MUL = the number of modulo multiplications,
DIV = the number of modulo division (inversion),
ADD = the number of modulo addition or subtraction,
HASH = the number of one-way or key-ed hash operations,
ENC = the number of encryptions using a private key cipher

Table 1: The additional computational cost for KRS

Once key recovery is requested, KRA unsigncrypts the DRF to recover the session key and make it sure that the DRF is generated by Alice. This unsigncryption procedure conforms to the way described in Section III.A.

In this scheme, the additional computational cost for KRS is given by Table 1. We neglect KRA's computational cost in Table 1 because key-recovery procedure occurs much less frequently than DRF generation.

The additional communication overhead, *i.e.*, the length of DRF, is as follows (in bits):

$$|q| + |KH.(\cdot)| + |k_{\text{session}}|. \tag{III.6}$$

The notation used in the above description of Signcryption-based KRS is summarized in Table 2:

Parameters public to all: p — a large prime q — a large prime factor of $p-1$ g — an integer with order q modulo p chosen randomly from $[1, \ldots, p-1]$ $hash(\cdot)$ — a hash function $KH.(\cdot)$ — a key-ed hash function (E, D) — encryption and decryption algorithms of a private key cipher
Alice's keys: x_A — Alice's secret key y_A — Alice's public key ($y_A = g^{x_A} \bmod p$)
KRA's keys: x_K — KRA's secret key y_K — KRA's public key ($y_K = g^{x_K} \bmod p$)
Other materials (usually temporary, or fresh): k_{session} — Session key (r_K, s_K, c_K) — Output as DRF x — Secret signcrypting parameter randomly picked from $[1, \ldots, q]$ by Alice k_1 — Private key for encryption E and decryption D k_2 — Key used in computing the key-ed hash function KH

Table 2: Notation used in the KRS based on Signcryption

Type	Functions	Computational Cost	
Encryption	Confidential	RSA encryption	EXP=1, MUL=0, DIV=0 ADD=0, HASH=0, ENC=0
		ElGamal encryption	EXP=2, MUL=0, DIV=0 ADD=0, HASH=0, ENC=1
Signature then Encryption	Signature + Confidential	RSA signature then encryption	EXP=2, MUL=0, DIV=0 ADD=0, HASH=1, ENC=0
		Schnorr signature + ElGamal encryption	EXP=3, MUL=1, DIV=0 ADD=1, HASH=1, ENC=1
Signcryption	Signature + Confidential	SCS1	EXP=1, MUL=0, DIV=1, ADD=1, HASH=2, ENC=1

Table 3: Comparison of the computational cost

IV Discussion

We described an "authenticated KRS" based on Signcryption in the previous section. This section examines the efficiency of the Signcryption-based KRS in comparison with other implementations of KRSs based on conventional primitives, including unauthenticated KRSs.

IV.A Comparison of the computational cost

Table 3 shows the computational costs of various KRS implementations.

Table 3 suggests us that the signcryption scheme is least costly among the schemes which provide both authenticity and confidentiality. For more specific comparison, we have to consider currently-required sizes of the exponents and the RSA composit $|n_A|$.

Assuming that Chinese Remainder Theorem reduces the cost of RSA-signature generation to be $0.375|n_A|$[1], Signcryption-based scheme saves the computational cost by

$$\frac{0.375|n_A| - 1.5|q|}{0.375|n_A|} = 37.5\%, \tag{IV.1}$$

where $|n_A| = |p| = 1024$ and $|q| = 160$, in comparison with RSA signature-then-encryption. On the other hand, the cost reduction in comparison with the scheme using "Schnorr signature + ElGamal encryption" is estimated as follows:

$$\frac{\text{2 modulo exponentiations}}{\text{3 modulo exponentiations}} = 66.7\%. \tag{IV.2}$$

IV.B Comparison of the communication overhead

The comparison of the communication overhead among various KRS models is given by Table 4.

The advantage of the signcryption scheme is obvious from Table 4, since the overhead of it includes no significant data block larger than $|p|$ or $|n_X|$; $|KH.(\cdot)| + |q| + |k_{\text{session}}| \ll |p|,\ |n_X|$.

Assuming the same security level in the previous subsection, the cost reduction of Signcryption-based scheme in comparison with the scheme with the second smallest overhead, (that is, the encryption-only scheme using RSA), is given by

$$\frac{|n_K| - (|KH.(\cdot)| + |q| + |k_{\text{session}}|)}{|n_K|} = 70.3\%, \tag{IV.3}$$

where $|n_K| = |p| = 1024$, $|q| = 160$, $|KH.(\cdot)| = 80$, and $|k_{\text{session}}| = 64$.

Type	Functions	Communication Overhead (in bits)	
Encryption	Confidential	RSA encryption	$\vert n_K \vert$
		ElGamal encryption	$\vert p \vert + \vert k_{\text{session}} \vert$
Signature then Encryption	Signature + Confidential	RSA signature then encryption	$\vert n_K \vert + \vert n_A \vert$
		Schnorr signature + ElGamal encryption	$\vert hash(\cdot) \vert + \vert q \vert + \vert p \vert + \vert k_{\text{session}} \vert$
Signcryption	Signature + Confidential	SCS1	$\vert KH.(\cdot) \vert + \vert q \vert + \vert k_{\text{session}} \vert$

where n_X is entity X's moduli used in RSA.
Table 4: Comparison of the communication overhead

Thus the signcryption scheme is more efficient than other implementation of authenticated KRS, both in the computational cost and in the communication cost. In addition, the communication overhead of the signcryption scheme is even smaller than RSA encryption-only scheme. Authenticated KRS is, thus, not only desirable from security point of view, but also sufficiently feasible from efficiency point of view if we implement it by using signcryption.

V Conclusion

We introduced a concept of "authenticated KRS" in replace of encrypted-only KRS. Once obtained, authentication can offer more positive use of KRS not only in open networks but also in closed or private networks. This is due to the unforgeability and non-repudiation of the authenticated DRF.

Considering the efficiency, straightforward use of both public-key encryption and digital signature costs too much in computation and in communication. As a feasible solution to this problem, we proposed a more efficient way — the use of Signcryption, a recently-developed cryptographic primitive which fulfills both the functions of public-key encryption and digital signature. Comprehensive comparison demonstrated how signcryption makes authenticated KRS feasible for practical use.

This new concept, authenticated KRS, might probably have lots of potential applications. The study will go further on the possibility and also on technical details and extensions.

References

[1] Y. Zheng, "Digital Signcryption or How to Achieve Cost (Signature & Encryption) $\ll$ Cost(Signature) + Cost(Encryption)," to appear in *Advances in Cryptography - Crypto'97*, Springer-Verlag, Berlin, New York, Tokyo, 1997.

[2] D. E. Denning and D. K. Branstad, "A Taxonomy for Key Recovery Encryption Systems," (URL:http://www.cosc.georgetown.edu/denning /crypto/taxonomy.html), May 11, 1997.
D. E. Denning and D. K. Branstad, "A Taxonomy for Key Escrow Encryption Systems," Communication of the ACM, Vol.39, No.3, pp.34 - 40, March 1996.

[3] D. E. Denning and M. Smid, "Key Escrowing Today," IEEE Comm. Maga., pp.58 - 67, Sept. 1994.

[4] S. T. Walker, S. B. Lipner, C. M. Ellison, and D. M. Balenson, "Commercial Key Recovery," Communication of the ACM, Vol.39, No.3, pp.41 - 47, March 1996.

[5] D. P. Maher, “Crypto Backup and Key Escrow,” Communication of the ACM, Vol.39, No.3, pp.48 - 53, March 1996.

[6] R. Ganesan, “The Yaksha Security System,” Communication of the ACM, Vol.39, No.3, pp.55 - 60, March 1996.

[7] Trusted Information Systems, Inc., “TIS announces encryption Key Recovery Technology - Technical Description,” RSA Data Security Conference, San Francisco, California, January 18, 1996.

[8] Trusted Information Systems, Inc.,
“Exportable Strong Encryption: RecoverKey$^{\text{TM}}$
CSPs,” (URL: http://www.tis.com/docs/products/recoverkey /rkey3.html), November 1996.

[9] IBM Corp., “The need for a global cryptographic policy framework,”
(URL:http://www.ibm.com/security/html/ pp_global.html), October 1996.

[10] IBM Corp., “Key management framework and key recovery technology,”
(URL:http://www.ibm.com/Security/html/ wp_keymgt.html), February 1997.

[11] T. Kubo, “Key Pre-distribution System as a Key Recovery System,” Proc. of PKS’97, April 1997.