Key Recovery Compatible with IP Security

Yoon-Jung Rhee[1], Chan Koh[2], and Tai-Yun Kim[1]

[1] Dept. of Computer Science & Engineering, Korea University
{genuine, tykim}@netlab.korea.ac.kr
[2] Dept. of Computer Science, Seoul National Univ. of Technology
chankoh@duck.snut.ac.kr

Abstract

IPSec is a security protocol suite that provides encryption and authentication services for IP messages at the network layer of the Internet. Key recovery has been the subject of a lot of discussion, of much controversy and of extensive research. Key recovery, however, might be needed at a corporate level, as a form of key management. The basic observation of the present paper is that cryptographic solutions that have been proposed so far completely ignore the communication context. Static systems are put forward for key recovery at network layer and solutions that require connections with a server are proposed at application layer. We propose example to provide key recovery capability by adding key recovery information to an IP datagram. It is possible to take advantage of the communication environment in order to design key recovery protocols that are better suited and more efficient.

1. INTRODUCTION

IPSec is a security protocol suite that provides encryption and authentication services for IP messages at the network layer of the Internet [5,6,7,8]. Two major protocols of IPSec are the Authentication Header (AH) [7], which provides authentication and integrity protection, and the Encapsulating Security Payload (ESP) [8], which provides encryption as well as (optional) authentication and integrity protection of IP payloads.

Key recovery has been the subject of a lot of discussion, of much controversy and of extensive research, encouraged by the rapid development of worldwide networks such as the Internet. A large-scale public key infrastructure is required in order to manage signature keys and to allow secure encryption. However, a completely liberal use of cryptography is not completely accepted by governments and companies so that escrowing mechanisms need to be developed in order to fulfill current regulations. Because of the technical complexity of this problem, many rather unsatisfactory proposals have been published. Some of them are based on tamper-resistant hardware, others make extensive use of trusted third parties. Furthermore, most of them notably increase the number of messages exchanged by the various parties, as well as the size of the communications. Based on these reasons, the widespread opinion of the research community, expressed in a technical report written by well-known experts, is that large-scale deployment of a key recovery system is still beyond the current competency of cryptography. Despite this fact, key recovery might be needed at a corporate level, as a form of key management. The basic observation of the present paper is that cryptographic solutions that have been proposed so far, completely ignore the communication context. Static systems are put forward for key recovery at IP layer in the Internet.

This paper proposes a method for carrying byte oriented Key Recovery Information in a manner compatible with the IPSec architecture. We design a key recovery protocol that is connection oriented and more robust than other proposals

2. BACKGROUND ON KEY RECOVERY

The history of key recovery started in April 1993, with the proposal by the U.S government of the Escrow Encryption Standard, EES, also know as the CLIPPER project. Afterwards, many key recovery schemes have been proposed.

To protect user privacy, the confidentiality of data is needed. For this, key recovery (KR) seems useless, but there are some scenarios where key recovery may be needed :

- When the description key has been lost or the user is not present to provide the key

- Where commercial organizations want to monitor their encrypted traffic without alerting the communicating parties; to check that employees are not violating an organization's policy, for example
- When a national government wants to decrypt intercepted data for the investigation of serious crimes or for national security reasons.

3. RELATED PROTOCOLS

3.1 RHP Encapsulation

The RHP (Royal Holloway Protocol) [1] architecture is based on a non-interactive mechanism with a single exchanged message and uses the Diffie-Hellman scheme. The RHP system allows messages sent to be decrypted using the user's private receive key. Each user is registered with a TTP denoted TTP_A for user A. The following is the mechanism used in RHP.

1. A obtains $K_{pu-r(B)}$,$(= g^b \bmod p)$. TTP_A can compute $K_{pr-r(B)}$,$(= b)$, from B's name and $K(TTP_A, TTP_B)$.
2. A derives a shared key, $(g^b \bmod p)^x \bmod p = g^{xb} \bmod p$ from $K_{pr-s(A)}$. This is the session key, or the encryption key for the session key
3. A transmits $K_{pu-s(A)}$ signed by TTP_A and $K_{pu-r(B)}$. This information serves both as a KRF and as means of distributing the shared key to B.
4. Upon receipt, B verifies $K_{pu-s(A)}$ from A's public send key and $K_{pr-r(B)}$.

The main advantage of the RHP is to be robust in terms of basic interoperability. But, the drawback of the RHP is to mix key negotiation and key recovery. It is difficult to integrate this scheme inside the security protocols of the ISAKMP since the protocol has only one phase. Another drawback is that the KRF is sent once. In fact, this is a major disadvantage in the system since the session can be long and the KEA can miss the beginning. We refer to this difficulty as the session long-term problem. It is necessary to send the KRF more than once. However, the advantage of this system is to encrypt the session key with the shared key so that the security depends on the communicating peers and not on the TTP. But since the private receive keys depends on the TTP, this advantage disappears. Finally, this solution is hybrid between encapsulation and escrow mechanisms because the private send key is escrowed and the private receive key can be regenerated by both TTPs.

3.2 KRA Encapsulation

The KRA (Key Recovery Alliance) system proposes to encrypt the session key with the public key of the TTPs (Trusted Third Party). Key Recovery Header (KRH) is designed to provide a means of transmitting the KRF across the network so that they may be intercepted by an entity attempting to perform key recovery. The KRH carries keying information about the ESP security association. Therefore, KRH is used in conjunction with an ESP security association [9]. In the ISAKMP, the use of the KRH can be negotiated in the same manner as other IPSec protocols (e.g., AH and ESP).

Various schemes using this technique have been proposed such as TIS CKE (Commercial Key Escrow) [3], or IBM SKR (Secure Key recovery) [2]. The system is quite simple and allows many variations according to the cryptographic encryption schemes. This proposal separates the key recovery information and the key exchange. The system modularity is also compatible with the IETF recommendation. But, the KRF contains the encryption of the same key under a lot of TTP public key. Thus, the KRF can rapidly grow and one must take proper care against broadcast message attacks. The KRA solution is not necessary to sends a KRF in each IP packet inside the IPSec [9]. The intervals at which the initiator and responder send KRF are established independently. But since the KRF size is big, the KRF cannot be included in the IP Header. So, it can be sent in the IPSec header that is a part of the IP packet data. This leads to decrease the bandwidth. The second drawback is to encrypt the session key under the TTP public key. Finally, this solution is not robust because if this key is compromised, the system collapses.

4. PROPOSED KEY RECOVERY FOR IPSEC

4.1 System Overview

The main problem with the RHP proposal is that the protocol is connectionless-oriented. Therefore, the protocol is not well suited to IPSec or ISAKMP that are connection-oriented and allow interactivity.

The KRA's proposal seems a better solution than the RHP. Still, the security of the session key depends on a fixed key for all communications and, furthermore, the resulting IPSec protocol is not optimized in terms of network efficiency.

Our solution is based on IETF protocols in order to improve the security of the system, the network communication, and the interoperability for cross-certification. We can integrate modified RHP method in the IETF protocols (ISAKMP, IPSec) if we realize a real Diffie-Hellman key exchange such as in Oakley [4] in

order to negotiate a shared key. After this first phase, the KRF is sent with the data.

In the ISAKMP, the negotiation for security association of key recovery arises. To increase flexibility, we modify the step 2 in RHP mechanism.

1. A obtains $K_{pu-r(B)}$,$(=g^b \bmod p)$.
2. A derives a shared key, $(g^b \bmod p)^{x^*} \bmod p = g^{bx^*} \bmod p$; This is the encryption key for the session key
3. A transmits $K_{pu-s(A)}$ signed by TTP_A and $K_{pu-r(B)}$.
4. Upon receipt, B verifies $K_{pu-s(A)}$ from A's public send key and $K_{pr-r(B)}$.

In the step 2, x^* could be a temporary secret, computed as:

$$x^* = f(x, TT).$$

Where f is a one-way function and TT is a time stamp. Consequently, this can be more robust because it reduce the influence affected by escrowing the private receive keys depends on the TTP.

The TTPs can recover the key as well as the user (execute a Diffe-Hellman operation) since they escrow the user's private send key. At the beginning of the session, A sends the cross-certificates of both TTPs. This enables B to verify A's certificate signed by TTP_A without connection to TTP_B as in the RHP. In this scheme A has the cross-certificate of the TTP in the initialization phase. This improves the first phase.

During the IPSec session, we send the KRF with the encrypted message. Even if we keep the same secret key, a KRF must be sent, since the session key is not escrowed. Hence, the KRF is sent many times according to an accepted degradation bandwidth. We send the KRF in the IPSec packet as a part of IP packet. Finally, a variant can send the session key encrypted with the public key of both users instead of the shared Diffe-Hellman key. So, the KRF only depends upon a specific user. This allows sending the KRF in a single direction according the user's policy. User A can choose to send (or not) the session key encrypted with his TTP's public key and user B can do the same. This is an interesting feature compared to the RHP, since in the RHP scheme both TTP can decrypt all messages without communication with each other.

4.2 Comparison of Protocols

In this section, we compare existing protocols and our proposed protocols. In Table 1, we show the performance evaluation result between proposals of RHP, KRA and our proposed.

Table 1: Comparison of protocols

(O: high support △: low support X: not support)

	RHP	KRA	The Proposed
compatibility with IETF	X	O	O
robustness	△	X	O
reducing overhead of network	O	△	△

5. CONCLUSION

Our proposal is a mix of the RHP and the KRA solutions that combines the advantages of both systems. This scheme is based on an escrow mechanism. First, we keep the interoperability of the RHP, improve robustness comparing with RHP, and include it in the Internet Protocols. Secondly, the KRA solution is used but we encrypt the session key with a shared key by Diffie-Hellman key exchange between communicating users or user's public key and not with the TTPs public keys to gain robustness.

REFERENCES

1. N. Jefferies, C. Mitchell, and M. Walker, "A Proposed Architecture for Trusted Third Party Services", in Cryptography: Policy and Algorithms, Proceedings: International Conference BrisAne, Lecture Notes In Computer Science, LNCS 1029, Springer-Verlag, 1995.
2. R. Gennaro, P. Karger, S. Matyas, M. Peyravian, A. Roginsky, D. Safford, M. Zollett, and N. Zunic. "Two-Phase Cryptography Key Recovery System." In computers & Security, Pages 481-506. Elsevier Sciences Ltd, 1997.
3. D. M. Balenson, C. M. Ellison, S.B. Lipner and S. T. Walker, "A new Approach to Software Key Encryption", Trusted Information Systems.
4. The Oakley Key Determination Protocol (RFC 2412)
5. Internet Security Association and Key Management Protocol (ISAKMP) (RFC 2408)
6. The Internet Key Exchange (IKE) (RFC 2409)
7. IP Authentication Header (AH) (RFC 2402)
8. IP Encapsulating Security Payload (ESP) (RFC 2406)
9. T. Markham and C. Williams, Key Recovery Header for IPSEC, Computers & Security, 19, 2000.