Secure Key Recovery for Archived and Communicated Data in the Corporate Intranet

EMMANOUIL MAGKOS[1], VASSILIOS CHRISSIKOPOULOS[2],
NIKOS ALEXANDRIS[1] AND MARIOS POULOS[1]

[1]Department of Informatics,
University of Piraeus,
80 Karaoli & Dimitriou, Piraeus 18534
GREECE
emagos@unipi.gr; alexandr@unipi.gr; marios.p@usa.net
http://thalis.cs.unipi.gr/~emagos

[2]Department of Archiving and Library Studies,
Ionian University
Old Palace Corfu, 49100
GREECE
vchris@ionio.gr

Abstract: - During the last years there has been an explosion of interest in *key recovery systems* that enable recovery of plaintext from archived or intercepted ciphertext, for key management within the corporate environment or for law enforcement in forensic applications. In this paper we overview various approaches for key recovery and consider attacks against such systems. We also propose a key recovery model for archived or communicated data in the corporate intranet that deals with such attacks. Our model is equitable in the sense that it protects the employees' privacy while ensures time-efficient data recovery. For this reason we employ *traditional* recovery techniques for long-term keys as well as a *key encapsulation* technique for secure and efficient policy enforcement.

Key-Words: - Key Recovery, Key Management, Archived and Communicated Data, Intranets, Cryptography

1 Introduction

With the development of cryptography and its growing use of protecting communicated and archived date a critical issue has evolved concerning the loss of decryption keys. Loss of keys means that decryption is infeasible, resulting in inaccessibility of data. Corporations will find such situations unacceptable, especially if the inaccessible data hold potentially valuable information. *Key recovery systems* [3] provide retrieval of plaintext from intercepted, archived or confiscated ciphertext, under certain (well defined) conditions.

Various agencies may be involved in a key recovery system. These include the Users, a Message Recovery Agency, a Policy Enforcement Agency and Policy Makers. Key recovery can be imposed by Policy Enforcement Agencies or simply used to support user backup facilities. In this paper we focus on software-based key recovery systems that support policy enforcement in corporate environments, while at the same time protect the rights of the individuals. In this respect, these systems are *equitable* [2].

Ethical Issues in Key Recovery. There are numerous and important sociological, ethical and legal issues raised by key recovery, especially concerning systems intended for law enforcement [1]. There is also a strong debate whether such systems, being part of an organization's security policy, would violate the employees' privacy. On the other hand, key recovery within the corporate environment may be considered as a very important safeguard against fraud and error.

Security Issues in Key Recovery. Several weaknesses and attacks on key recovery systems have been described in the literature [1,6,9,13]. In this paper we consider attacks by users who wish to bypass the key recovery mechanism. Such attacks may involve *communicated data* (where a Sender circumvents the security policy and sends an unrecoverable ciphertext to a Receiver) as well as *archived data* (in this case it is obvious that

Sender=Receiver). There are essentially two types of such attacks:

The General Double Encryption attack. In this attack the sender pre- or post-encrypts data by using another (non-escrowed) cryptosystem [6]. This attack is considered difficult to deal with, unless one assumes that all users have no other encryption systems available. The above assumption may sound unrealistic in law enforcement scenarios where communicated data are intercepted over the Internet. However in the corporate environment such assumptions could be enforced as part of a robust security policy. As shown in the sequel, this does not trivialize the key recovery problem.

The Pfitzmann-Waidner attack. This attack is a special case of the General Double Encryption attack: the attacker uses the key recovery system itself for the inner encryption [13]. With this attack, users are able to defeat the system without the need to use any other cryptosystem.

Outside the corporate environment, it seems almost impossible to prevent two well-determined attackers from bypassing any key recovery process. For example, the attackers may use pure *steganography* [12], or even design their own steganographic / cryptographic algorithms. Moreover, by having an a priori shared secret, attackers may also use unconditionally secure cryptographic mechanisms such as *one-time pads* [17] to circumvent enforcement policies. This is the main reason why key recovery systems have never been massively deployed in the law enforcement field. On the contrary, we believe that a well-defined communication infrastructure established within corporate intranets and carefully extended to inter-organizational extranets would ensure the continued availability of critical corporate data. Such an infrastructure should also balance between the need for availability and the need for privacy for the employees.

2 An Overview of Key Recovery

The Traditional Approach (e.g. Clipper [5]). This approach is also referred to as *long-term* key escrow, and involves escrowing with (usually) a set of authorities [11] long-term secret decryption keys that correspond to certified public keys. A drawback of this approach is that there is little control over the period of key-recovery. Once a private key is recovered[1], even ciphertexts sent long *before* or *after* recovery may be illegally decrypted. Several attempts have been made to introduce time-bounding mechanisms in long-term key recovery (e.g. [2]).

Partially Weak Cryptosystems (e.g. restricted key-length crypto [10]). With such systems, while it is computationally possible for a designated authority to recover the key of an encrypted message, it is computationally prohibitive to launch large-scale wiretapping. In the literature, *partial key escrow* of session keys was first proposed by Shamir [16] as a method to escrow all but *k* bits of the key (e.g. k=48 bits). Drawbacks of such mechanisms are the complexity of the key recovery process and low user acceptability.

Key Encapsulation (e.g. IBM SKR [7]). This is also referred to as *session key recovery*, or *virtual addressing*. Short-term or ephemeral keys are encrypted in capsules that can be decrypted only by the receiver and a designated authority [18]. Such systems are inherently time-bounded.

Trusted Third Parties (e.g. Royal Holloway [8]). Session keys are distributed on-line by Trusted Third Parties. Key recovery mechanisms of this type are designed explicitly for multiple domain environments. Drawbacks are the high storage requirement, communication time and overheads.

Data Confiscation (e.g. RIP Act 2000 [14]). Encrypted data are confiscated and the receiver is obliged to decrypt it. Although it has been heavily criticized by the press and on the Internet[2], it inherently maintains a better level of privacy than traditional key escrow, since no long-term keys are escrowed.

3 A Secure Key Recovery Model for the Corporate Environment

All key recovery systems described so far are inherently subject to double encryption attacks (see also Section 1). In this Section we propose, at high level, a secure and practical key recovery model that deals with double encryption attacks under the assumption that all long-term secret decryption keys are escrowed. Since this assumption can only be enforceable in the corporate environment, our model is destined for protecting archived and/or

[1] With techniques such as *function sharing* [15], the plaintext of an encrypted message can be recovered without explicitly reconstructing a long-term private key (*message recovery*).

[2] For example, there has been much criticism on the *burden-of-proof* reversal in the RIP Act 2000 [14]: the proof of the inability to decrypt a message lies with the addressee.

communicated data within an organization's private network, as part of its key management policy.

3.1 Participants

The participants in our model are the users, a Service Provider (SP), a Policy Enforcement Agency (PEA), a Ticket Granting Service (TGS), a Message Recovery Service (MRS), and a Key Escrow Agency (KEA).

Service Provider (SP). The SP is the hub of our system. It provides a *packet filtering* mechanism that filters out ciphertexts that do not have the specified format or are defective. The SP keeps temporary logs[3] of all traffic. When presented with an appropriate ticket, it will give time-bounded access to its logs. While the SP is able to check encrypted communication for defects, it cannot decrypt it.

Policy Enforcement Agency (PEA). This agency triggers the key recovery mechanism. If there is reason to believe that some ciphertexts sent to a particular user are suspect, the PEA will request from the TGS a time-bounded ticket to access the plaintext. In our model we assume that the PEA has at least as much cryptanalytic power as any user of the system.

Ticket Granting Service (TGS). The TGS is an offline service that issues time-bounded tickets authorizing the decryption of an encrypted session between two individuals. To issue a ticket, the TGS must be presented with sufficient evidence that ciphertexts of a given period need to be recovered. In the corporate context the process of getting a ticket is subject to the company's security policy.

Message Recovery Service (MRS). When presented with a ticket and the ciphertexts, the MRS will recover sufficient information to access the plaintext, or at least an inner encryption of it.

Key Escrow Agency (KEA). This agency possesses the long-term secret decryption key of every user that is a member of the corporate infrastructure. The KEA is trusted not to reveal the secret keys of the users to unauthorized parties. For additional security, the power of the KEA can be distributed among a set of trusted escrow agents. This will not add substantial complexity to the key recovery process, as the services of the KEA will be invoked only in special circumstances when all other key recovery attempts fail

[3] The SP could be, for example, a server-side module in a domain-like (e.g. as in Windows 2000) client-server security architecture for the corporate intranet, where archived or communicated data can be filtered out inside the domain controller.

3.2 The Data Recovery Component

We employ a *hybrid* key recovery mechanism that combines *traditional long-term key escrow* in which private decryption keys are escrowed, with *key encapsulation* for encryption.

For long-term keys, we can go along the usual model for key escrow mechanisms [11], in which the communication is encrypted with the decryption key escrowed to trusted agents. The agents are trusted to safeguard their shares and to enable decryption when necessary. For fault-tolerance, *threshold* [4] key recovery techniques may also be employed in designing the KEA. Users may encrypt their personal communications and archived data using the long-term public keys.

For time efficient access to corporate data we adopt a key encapsulation model [18]: The sender, say Alice, selects a session[4] key S and encrypts her message M by using an appropriate symmetric algorithm with secret key S. Then, Alice encrypts S with the public key of the receiver, say Bob, as well as with the public key of the Message Recovery Service (MRS). Given PK_{Bob} and PK_{MRS}, being the public encryption keys of Bob and MRS respectively, Alice sends to Bob the ciphertext:

$$C = [M]_S \,|\, [S]_{PK_{Bob}} \,|\, [S]_{PK_{MRS}} \,|\, \text{binding data}$$

where || denotes concatenation and $[]_K$ denotes encryption with key K (symmetric/public). The *binding data* is a non-interactive proof of correctness (in *zero-knowledge* [17]) that the session keys contained in the second and third component of C are the same. This can be checked (on-line or off-line) by any independent monitor (e.g. the SP), for verifiability. Binding techniques that support virtual addressing of session keys to several MRS's have also been proposed in the literature [18].

3.3 The Key Recovery Mechanism

This proceeds as follows:

1. Alice sends the ciphertext C to Bob. The ciphertext is intercepted by the SP and temporarily logged. The SP checks that C has the specified format, and that the data recovery component of C: $[S]_{PK_{Bob}} \,|\, [S]_{PK_{MRS}} \,|\, \text{binding data}$ is not defective: that is, that *binding data* is indeed a

[4] A new key may be selected for each encryption, or at a regular basis.

proof that the first two components $[S]_{PK_{Bob}}$ and $[S]_{PK_{MRS}}$ are encryptions of the same secret session key S. Defective ciphertexts are destroyed, and not forwarded to Bob.

2. If corporate data cannot be accessed or if the Policy Enforcement Agency (PEA) decides the traffic to Bob is suspect, the PEA requests a ticket from the TGS to recover the plaintext of ciphertexts addressed to Bob. During normal operation, in case of a key loss, Bob may also ask such a ticket from the TGS to decrypt any archived data created by him (in this case Alice=Bob), or any communicated data sent to him by Alice. If the request is justified, a time-bounded ticket T is issued. T includes an identifier ID_{Bob} for Bob.
3. The PEA presents the ticket T to the SP and obtains all ciphertext addressed to Bob from logs of the SP (including Bob's archived data), for the period referred to in T.
4. The PEA will forward the ciphertext C (as well as all other ciphertexts addressed to Bob within the time-bounded period), along with T to the Message Recovery Service (MRS). The MRS will use its secret decryption key to recover the session key S, and hence the plaintext M. The MRS sends $\{M, ID_{Bob}\}$ to the PEA. If the PEA is satisfied that M is indeed the intended plaintext then the key recovery process is completed. However, if there are any reasons to believe that M contains hidden information, say M', that is encrypted with the public key of Bob, then it sends $\{M, ID_{Bob}\}$ to the Key Escrow Agency (KEA).
5. The KEA (or a threshold of KEA's, if threshold decryption is used) will use Bob's decryption key to recover the plaintext M' from M. The KEA sends $\{M', ID_{Bob}\}$ to the PEA. At this point, the key recovery process is completed.

Under our assumption that all long-term secret decryption keys are escrowed, and that the PEA has at least as much cryptanalytic power as Bob, any ciphertext will be decrypted directly, or with the help of the KEA.

4 Conclusion

In this paper we overviewed various approaches for key recovery and proposed, at high level, a hybrid model that can be used in the corporate environment to deal with double encryption attacks. In our model we require that all long-term decryption keys are escrowed to a Key Escrow Agency (KEA). To exclude viewing the KEA as a single point of attack we expect the KEA to be highly distributed (as first proposed by Micali [11]). This will introduce an increased complexity during the key generation and recovery process. However the KEA is not involved during normal operation, but only in special circumstances when all other recovery attempts fail (e.g. during a double encryption attack). Users may encrypt their personal communications and archived data using the long-term public keys. For routine key recovery of corporate data we employ efficient *key encapsulation* techniques: corporate data are encrypted with session keys that are virtually addressed to a Message Recovery Service (MRS), which, when presented with a ticket, will assist recovering a message in a timely manner. In this way our model is equitable as it allows quick around-the-clock access to critical corporate plaintexts while protects long-term keys from being easily manipulated by policy enforcement agencies.

References:

[1] H. Abelson et al. The Risks of Key Recovery, Key Escrow, Trusted Third Party & Encryption. Digital Issues No. 3, pp. 1-18, 1998.

[2] M. Burmester, Y. Desmedt, and J. Seberry. Equitable Key Escrow with Limited Time Span (or How to Enforce Time Expiration Cryptographically). In Advances in Cryptology - ASIACRYPT '98, LNCS 1514, Springer-Verlag, pp. 380-391, 1998.

[3] D. Denning and D. Branstad. A Taxonomy of Key Escrow Encryption Systems. Comm. of the ACM, Vol. 39(3), pp. 34-40, 1996.

[4] Y. Desmedt and Y. Frankel. Threshold Cryptosystems. In Advances in Cryptology - CRYPTO '89, LNCS 435, pp. 307-315, 1989.

[5] FIPS PUB 185, Escrowed Encryption Standard. US Department of Commerce, February 1994.

[6] Y. Frankel and M. Yung. Escrow Encryption Systems Visited: Attacks, Analysis and Designs. In Advances in Cryptology - CRYPTO '95, LNCS 963, pp. 222-235, 1995.

[7] R. Gennaro, P. Karger, S. Matyas, M. Peyravian, A. Roginsky, D. Safford, M. Zollet and N. Zunic. Two-Phase Cryptographic Key Recovery System. Computers & Security, Elsevier Sciences Ltd, pp. 481-506, 1997.

[8] N. Jefferies, C. Mitchell and M. Walker. Trusted Third Party based Key Management allowing Warranted Interception. Public Key Infrastructure

Invitational Workshop, MITRE McLean, Virginia, USA, September 1995.

[9] S. Kim, I. Lee, M. Mambo and S. Park. On the Difficulty of Key Recovery Systems. In Information Security Workshop ISW-99, LNCS 1729, Kuala Lumpur, Malaysia, pp. 207-224, November 1999.

[10] B. J. Koops. Crypto Law Survey-Overview per Country. Version 20.0, March 2002. At: http://cwis.kub.nl/~ frw/people/koops/cls2.htm

[11] S. Micali. Fair Public Key Cryptosystems. In Advances in Cryptology - CRYPTO '92, LNCS Vol. 740, Springer-Verlag, pp. 113-138, 1993.

[12] F. Peticolas, R. Anderson and M. Kuhn. Information Hiding-A Survey. In Proceedings of the IEEE, Special Issue on Protection of Multimedia Content, IEEE Vol. 87(7), pp. 1062-1078, 1999.

[13] B. Pfitzmannm and M. Waidner. How to Break Fraud-Detectable Key Recovery. EUROCRYPT-'97. Rump Session, Konstanz, Germany, May 13, 1997.

[14] Regulation of Investigatory Powers Act 2000. www.homeoffice.gov.uk/ripa/

[15] A. De Santis, Y. Desmedt, Y. Frankel and M. Yung. How to Share a Function Securely. In 25th Annual Symposium on Theory of Computing, ACM Press, pp. 522-533, 1994.

[16] A. Shamir. Partial key Escrow: A New Approach to Software Key Escrow. Key Escrow Conference, Washington, D.C., September 15, 1995.

[17] B. Schneier. Applied Cryptography - Protocols, Algorithms and Source Code in C. 2nd Edition, 1996.

[18] E. Verheul, C. Henk and C. van Tilborg. Binding ElGamal: A Fraud-detectable Alternative to Key-Escrow Proposals. In Advances in Cryptology - EUROCRYPT '97, LNCS 1233, pp. 119-133, 1997.