Matching key recovery mechanisms to business requirements

Konstantinos Rantos and Chris J. Mitchell
Information Security Group,
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, UK.
Kostas@rantos.com, C.Mitchell@rhul.ac.uk

Abstract

This paper addresses the business needs for key recovery as a countermeasure to the threat of losing potentially valuable information. Several requirements essential for a sound key recovery mechanism are described, and the applicability of two main classes of existing key recovery schemes to a corporate environment is examined. Different requirements are identified for key recovery mechanisms for communicated and archived data, and a further study is made of the applicability of existing mechanisms to these two cases.

Keywords: key recovery, key management, disaster recovery, encryption.

1 Introduction

In the information age it has become vital for businesses and organisations to protect their most valuable asset, i.e. the information they possess, from unauthorised access both from outsiders and insiders. Encryption mechanisms are deployed, amongst other countermeasures, for this purpose. Use of these mechanisms, however, might lead to undesirable situations where access to encrypted data is not feasible due to loss of, or inaccessibility to, the encryption keys.

The resulting loss of important information might be very serious. Corporations will typically not wish to tolerate such a loss, especially if the inaccessible data hold potentially valuable information. Key recovery mechanisms (KRMs) can help overcome problems arising when encryption keys are lost, and hence prevent loss of information. KRMs allow authorised parties to retrieve cryptographic keys used for data confidentiality with the ultimate goal of recovering the encrypted data [1, 2].

The term key recovery (KR) or more specifically, key escrow, has attracted much unfavourable publicity mainly because of a number of government proposals for compulsory escrow of all private communications keys, see e.g. [3].

The intention of these proposals was to give governments the ability to decrypt intercepted communications to deal with criminal activities. However, this has been seen by a number of parties as a potential infringement of the rights of individuals and corporations to provide privacy for data stored and communicated electronically.

In a business environment, however, the situation is rather different. A company normally owns its information, and therefore the issues surrounding access to private communications through compulsory key escrow do not arise. KRMs deployed in a corporate environment can be thought of as part of routine disaster recovery planning.

This paper looks at threats that corporate information might face from loss of encryption keys, and at the various scenarios in which these threats might be realised. Based on this analysis, the business need for KR is outlined, and the concomitant requirements for a KRM are described. The applicability of two main types of KRMs to a business environment is then examined, and the pros and cons of these mechanisms when used for communicated and archived data are investigated. The need for this latter distinction arises from the fact that, as discussed below, different KR requirements exist for these two types of data. Note that possible legal requirements for access to business communications are not examined here.

2 Business needs for key recovery

Protection of information through the use of security mechanisms has become vital for business. Cryptographic keys, including key agreement keys, session keys used for encrypting communication sessions or stored data, and signature keys, are a crucial part of the security infrastructure protecting corporate data. Loss or unavailability of encryption keys will lead to an inability to access the encrypted information, a situation the corporation will typically not wish to tolerate. Within a business environment there are many cases where access to keys might be lost, arising from both deliberate actions and accidents. The former might originate from both outsiders and insiders, while accidents can be due to a failure of mechanisms.

As far as deliberate actions are concerned, it has been reported that more attacks to corporations' systems are likely to come from insiders rather than outsiders [4, 5]. This needs to be taken into account when establishing a cryptographic infrastructure offering services such as data confidentiality. Employees acting as the only holders of encryption keys might pose a threat to the corporation. Suppose a user's employment is terminated, and that the user is the only holder of keys used to encrypt business information. On leaving the company, the employee might withhold these keys, either deliberately or through simply forgetting to hand them to their legitimate owner. If there is no backup of these keys, and there is no way to recover or recompute them, then access to the information encrypted under them is infeasible (assuming that the cryptographic mechanisms used are strong enough to prevent a cryptanalytic means of decryp-

tion). Similar problems arise when an employee cannot be contacted because he is absent, e.g. on vacation. It is easier and safer to be able to recover the encryption key within the company's protected environment rather than having to contact the user, in some cases in insecure environments, bearing the risk of accidentally revealing the keys to untrusted third parties.

As far as failure of, or damage to, devices is concerned, this is always a threat in the business environment. More specifically, if encryption keys are stored in a damaged device, and there is no backup or other means to recover the keys, then data encrypted under the inaccessible keys will be lost. This device could be a hard disk storing a file of passwords used to derive keys, or a smart card containing a key or key component. In the latter case, losing the token itself is also not unlikely.

In [4] it is mentioned that AccessData of Orem, Utah, a company that provides software and services to companies that have lost access to encrypted data, "reported in 1995 that they received about a dozen and a half calls a day from companies with inaccessible computer data. About a third of these calls resulted from disgruntled employees who left under extreme conditions and refused to cooperate in any transitional stage by leaving necessary keys. Another half-dozen resulted from employees who died or left on good terms but simply forgot to leave their keys. The remaining third resulted from loss of keys from current employees."

In addition to the possibility of key loss, companies may wish to monitor employees' communications, either external or internal, e.g. to track leaks of information. This is especially necessary in hierarchical environments where exchange of proprietary information, even within the company's domain, needs to be monitored. Corporate monitoring of communications can also deter employees wishing to break security policies governing the flow of classified information [6]. Corporations may further wish to have access to encrypted communications for non-repudiation purposes in the event of a dispute, or even for running checks on incoming traffic, e.g. for viruses or for intrusion detection.

Although key recovery can be used to deter employees, it can also be used to promote the use of cryptography. Unless they are sure that the data they encrypt can be recovered even if they lose the decryption keys, employees may be reluctant to use encryption, hence leaving their data unencrypted.

Key recovery mechanisms have been devised to address these problems, and in some cases can overcome them quite efficiently. KRMs recover session keys used for encryption of data, which can then be used to decrypt the data [1, 6, 7, 8, 9]. KRMs are usually divided into key escrow and key encapsulation mechanisms. A typical key escrow mechanism involves the storage of keys or key-related information by one or more trusted agents, giving them the ability to recover the user's decryption keys. In a key encapsulation mechanism, a KR block or field is associated with the encrypted information, and this block contains the information necessary for an authorised entity to recover the data encryption key. The KR information generated by the user is typically attached to the encrypted data, and can be parsed only by an authorised entity such as the Key Recovery Agent (KRA).

3 Distinguishing between a business environment and law enforcement access

The term key recovery, or more specifically key escrow, has attracted much unfavourable publicity mainly because of a number of government proposals for compulsory escrow of all private communications keys, see e.g. [3]. The intention of these proposals was to give governments the ability to decrypt intercepted communications to deal with criminal activities. However, this has been seen by a number of parties as a potential infringement of the rights of individuals and corporations to provide privacy for data stored and communicated electronically.

In a business environment, however, the situation is rather different [10]. Corporations cannot tolerate loss of potentially important information through the unavailability of encryption keys. Further, and most importantly, a company normally owns its information, and therefore the issues surrounding access to private communications through compulsory key escrow do not arise. KRMs deployed in a corporate environment can be thought of as part of routine disaster recovery planning.

Moreover, the requirements for KRMs used for law enforcement access and those deployed in a business environment are slightly different, making this distinction important. As described in [10], the requirements for a KRM designed for law enforcement access include the following.

1. **Access without end-user knowledge or consent**. While law enforcement typically requires the monitoring of users' communications without the latter being aware, recovery of keys in a business environment does not necessarily have this requirement. The company, as the owner of the encrypted data, has the right to recover keys should an emergency situation arise, with or without the user's knowledge or consent.

2. **Ubiquitous adoption**. Governments have been seeking the ubiquitous adoption of "key recovery for all encryption, regardless of whether there is benefit to the end-user" [10]. In a business environment, however, deployment of a KRM will be in a restricted controlled environment where users can benefit from the existence of KRMs, as these can help prevent loss of access to their data.

3. **Fast access to plaintext**. Law enforcement access to encrypted communications demands service availability round-the-clock, and seeks the ability to obtain decryption keys quickly (in some cases within one or two hours). This is to help monitor fast-moving criminal or terrorist activities. Within a corporation, however, time restrictions are not expected to be such an important requirement when the KRM is used for archived data and the corporation can typically tolerate longer response times.

Granularity of keys is another strong requirement for KRMs deployed for law enforcement needs as it limits LEA access to those communications authorised by a valid warrant [11]. In a business environment, however, granularity is of

minor importance as the corporation will typically have the right to access any corporate data.

A discussion of the need to distinguish between business and law enforcement requirements can also be found in [12]. Note, however, that in [12] the authors claim that *enforceability*, i.e. a requirement that the users cannot circumvent the KRM, is not strongly required in a business environment. We believe that this is not always true as enforceability (or *non-circumventability*, as it is identified in this thesis) is necessary for the prevention of rogue user attacks. The existence of a controlled environment, i.e. an organisation's infrastructure, does not necessary mean that the users cannot manipulate the generated key recovery information (in some KRMs), thus circumventing the key recovery functionality.

In the following section the requirements on a KRM deployed specifically in a business environment are described in detail.

4 Requirements for KRMs deployed in a business environment

Although key recovery mechanisms address problems arising from loss of decryption keys, they should always be deployed with extreme care. If the mechanism is not properly deployed it can seriously weaken security, as KR provides an alternative means of access to encryption keys that may be easier for an attacker to exploit than the original computation process. Thus, the fundamental security requirement for any KRM is that the effort to exploit and break the cryptographic infrastructure with KR added should not be less than the effort required if the cryptographic infrastructure lacks KR functionality. Moreover, the KRM deployed should not weaken the cryptographic mechanisms used. In particular, it should not necessitate the use of specific mechanisms and algorithms which may be weak.

Another obvious requirement is that honest users and agents should be able to successfully use the KRM, and, if possible, the deployed mechanism should be transparent to users and acceptable by users and agents. Moreover, the mechanism should not be vulnerable to rogue user attacks.

The following list gives requirements for a KRM deployed in a business environment. It is not unlikely, however, that most of the these requirements also apply in a LEA environment. For instance, some of these (in particular requirements R2, R3, R4, and R5) are identified in [11] where the authors consider them as a general framework for analysis of key recovery systems regardless of the environment that they are used. Detailed requirements regarding a general functional model of a key recovery system are also proposed by the National Institute of Standards and Technology in [13].

R1. *Non-circumventability*: The KRM should be infeasible to circumvent, i.e. users must not be able to use the cryptographic mechanisms while bypassing the KR information generation process. Further, the user should not be able to generate invalid KR information or alter/delete it after its generation.

It should be noted that a lot of mechanisms do not meet this requirement, making them vulnerable to rogue user attacks.

R2. *User completeness*: Honest users should succeed in making use of the KRM to produce valid KR information. This requirement covers any need for the availability of a server that will be involved in the KR information generation process, or of any public key material required by the KRM during this process. User completeness should also satisfy the users need for being able to use the KRM to recover their keys without the agent's intervention. This is particularly relevant to the use of a KRM for archived data.

R3. *Agent completeness*: An authorised entity complying with the company's rules and policy should succeed in recovering the required keys. Any authorised attempts to recover keys, successful or not, should be logged for audit purposes.

R4. *User soundness*: As a result of the first requirement, any attempt to misuse the protocol by a dishonest user should be prevented, or at least be detectable. More specifically, a rogue user should not succeed in establishing a secure communication or encrypting archived data while producing invalid KR information without being detected and logged for audit purposes.

R5. *Agent soundness*: Any attempt by an agent to misuse the protocol or recover keys without complying with the company's policy should at least be detectable. Within this category of misuses fall both attempts by unauthorised entities and unauthorised attempts by authorised entities. Agent soundness can be achieved by the use of well-established access control mechanisms. Any unauthorised attempt to use the recovery process, successful or not, must be logged for audit purposes.

R6. *User acceptability*: The protocol should be acceptable to users. However, in a corporate environment this property is less important, since the use of the mechanism will typically be specified within the corporate security policy, and hence part of the conditions of employment. Nevertheless, factors that will help acceptability include: making clear the benefits of its use, its flexibility, availability, compatibility and interoperability with existing schemes, and, probably most importantly, its efficiency of use (see property 11 below).

R7. *Agent acceptability*: The protocol should be acceptable to entities authorised to recover keys. The main factor that will lead to acceptability is the protocol's efficiency, i.e. it should be easy for agents to recover keys efficiently and quickly by using the protocol in accordance with the corporate security policy.

R8. *Policy compliance*: The protocol should operate within the corporate security policy, and should satisfy all relevant legal restrictions. Within the

latter fall any constraints imposed by laws within the domain the corporation operates, such as those covering cryptographic algorithm use and export.

R9. *Flexibility*: The KR scheme should not prohibit the use of well-known and secure cryptographic mechanisms. This is closely related to the main security requirement that the KRM should not weaken or introduce any vulnerabilities to the cryptographic infrastructure by imposing restrictions on the mechanisms used.

R10. *Interoperability*: The mechanism should be capable of dealing with dissimilar KRMs, or cryptographic mechanisms that do not have KR functionality. This is particularly important for communications enhanced with a KR capability. A thorough analysis of the interoperability problem and a proposed solution is given in [14].

R11. *Mechanism transparency*: The KRM should be transparent to end users in that it should not introduce any significant computational overhead, or demand user interaction when the user employs the cryptographic mechanisms.

R12. *Negligible cost*: Deployment of the KRM should introduce only a negligible increase in the overall cost of the cryptographic services used and the security infrastructure. In particular, the cost of using the mechanism should not exceed the value of the information encrypted using an inaccessible key.

Although, as previously mentioned, some of these requirements are also identified as requirements for a LEA environment, others are of no relevance to a KRM deployed for LEA access. One example of such a requirement is interoperability of key recovery mechanisms. This is because law enforcement access typically seeks the wide (even global) deployment of a KRM which is potentially not expected to interact with other schemes. However, the variety of existing KRMs and their deployment in various business environments is expected to cause interoperability problems in encrypted communications.

Other requirements differ from those for LEA access in the degree that they should be met. For instance, while user acceptability has proven to be one of the crucial requirements for the deployment of a KRM for LEA needs, it is not expected that it will be an important factor in a business environment, assuming that the benefits from the use of a KRM will be made clear to the users. Cost is another issue that corporations are likely to consider, in contrast to governments which can typically tolerate higher expenses.

The above list of requirements can also serve as criteria for evaluating a KRM. All of them are likely to be essential for a sound mechanism, but factors that can influence their criticality should also be taken into account. For example, a slight weakening of the user and agent acceptability requirements is allowable if all the other requirements are satisfied and there is no alternative.

The target of a KRM, that is whether it is used for communicated or archived encrypted data, can also affect the above requirements.

There are other issues involved in using KRMs. KRMs require the existence of an infrastructure supporting key management techniques meeting the requirements of the mechanism. Clearly the administration and cost of the required infrastructure are factors that cannot be ignored. Moreover, if the mechanism is not appropriately deployed a number of security issues might arise. Storage of any KR information should be carefully protected to prevent unauthorised access. If appropriate access control restrictions on the recovery process are not enforced, an adversary without the required privileges might be able to recover keys and monitor communications or decrypt stored encrypted files. More important still, if the infrastructure lacks the existence of an audit log mechanism where all attempts to recover keys are monitored and reviewed, then the system could be significantly more vulnerable to compromise than it would be if it lacked KR functionality.

These requirements may be more difficult to meet if the corporation decides not to perform key recovery internally but rather to outsource the service. While there are certain risks associated with outsourcing the service (the agent might relax its security policies, go bankrupt, or even be bought out by a competitor while retaining the ability to recover keys [10]) this solution might be more attractive to small and medium-sized enterprises that are not willing to deploy their own infrastructure.

5 Classification and assessment of existing mechanisms

A variety of KRMs have been proposed by both the commercial sector and academia. Denning [1] gives a description of a wide range of KRMs identified as key escrow encryption systems, while [2] classifies the existing KRMs into several types. However, as previously mentioned, KRMs in the information security literature are usually divided into two types: *key escrow* and *key encapsulation* mechanisms.

5.1 Classification of key recovery mechanisms

In a typical **key escrow mechanism** an escrow agent holds a copy of all or part of the user's keys. According to [2], key escrow involves storing keys or key parts directly with one or more escrow agents, while in [13] a key escrow mechanism is described as a method of KR in which the secret or private keys, key parts, or key-related information to be recovered are stored by one or more Key Escrow Agents. As a result, each user has to escrow with his agent his private keys, or each session key that he uses. The agent is a TTP that operates within the corporation, or an external TTP with which the corporation has a contractual agreement. A typical key escrow agent, external or internal, could be an on-line TTP acting as a Key Distribution Centre (KDC) or a Key Translation Centre

(KTC), which keeps a copy, with the user's consent, of all keys that the user establishes.

In another scenario the user escrows an initial value, namely a Master Key, with his agent, which is subsequently used for the generation of all session keys (for example using a hash function and a time-stamp). An alternative is when the user escrows with his agent the private key of an asymmetric key pair that can be used to compute the secret session keys [15]. An example of the latter is the JMW scheme [8]. As can been seen from the above, a wide range of KR solutions can be classified as key escrow mechanisms. All of them are characterised by the storage of key-related information with a trusted agent that gives the latter the ability to recover all the user's decryption keys.

In a typical **key encapsulation mechanism** the user encloses the KR information (e.g. session keys or key parts) in an encrypted KR block which is made available to the agent(s) with which the user is associated, and can be decrypted only by this agent(s). The KR block is typically encrypted using the agent's public encryption key, and attached to the encrypted data as a Key Recovery Field (KRF) [9]. In a more general definition, [13], a key encapsulation mechanism is described as "a method of key recovery in which keys, key parts, or key related information are encrypted specifically for the KRA function and associated with the encrypted data", where the KRA function is "a key recovery system function that performs a recovery service in response to an authorised request". As with key escrow mechanisms the agent can be internal or external to the corporation.

There are also KR mechanisms which it is difficult to categorise into one of the above classes. There are, for instance, key escrow schemes which also require the transmission of a KRF for the KRA to be able to recover the keys. An example is a KRM where the user escrows his private decryption key with his agent, and uses the public key for the transmission of any key related material. Such a scheme can be considered as a **hybrid** mechanism, but for the purposes of this paper it is a key escrow scheme as it involves escrowing of key-related material, regardless of the transmission of a KRF. There are also key encapsulation schemes for which the KRF is not restricted to the transmission of session keys encrypted under the KRA's public encryption key. As an example consider the KRM proposed by Maher in [16]. Although the author claims that the proposed scheme is a key escrow mechanism, using the above definition the scheme should be classified as a key encapsulation mechanism, since the user does not escrow any key material with his KRA but he rather makes this information available to the KRA function.

5.2 Assessment of key escrow mechanisms

When key escrow mechanisms are used in a business environment, there is typically a need for a large storage capacity for the escrowed information and, more importantly, this information must be protected from unauthorised access. The latter is one of the main drawbacks of key escrow mechanisms, as pointed out frequently by their opponents. As mentioned in [10], the storage of all keying

material at a single point makes it a significant point of attack, and introduces a major vulnerability to the key escrow mechanism if appropriate countermeasures are not in place. The main protection mechanism that can be employed for this purpose is strong access control, ensuring that only authorised personnel can access the escrowed key material and recover user keys. Access control should be enforced in conjunction with appropriate audit log mechanisms that will enable the monitoring of all attempts to access the escrowed keys and make use of the recovery process. Dispersion of the key material to multiple locations, resistance to agents' collusion, and residual work factor can be used as additional countermeasures. If all these protection mechanisms are in place, supported by an appropriate security policy, the likelihood of misuse of the KRM can be minimised.

Key escrow mechanisms are likely to be integrated into a cryptographic infrastructure, i.e. the KR functionality will be closely related to the key establishment process. For these mechanisms to work properly and not to face interoperability problems (requirement R10), there is a need for a common infrastructure meeting the mechanism's requirements. In other words, such mechanisms usually demand the existence of specific key establishment protocols, a requirement that can cause interoperability problems in communications sessions, making them difficult to deploy world-wide. As a consequence, if they are deployed by a corporation they may constitute a barrier to encrypted internet communications with other organisations. Even if the KR information can be generated solely at the user's end, with no requirement for interaction with the peer, the dependence on the cryptographic infrastructure would require the communicating parties to use cryptographic mechanisms compatible with the KRM.

Such mechanisms are, however, potentially more appropriate for intranet communications, where it is easy to establish a common infrastructure, and also for archived data encryption. If the KRM is part of the cryptographic infrastructure, and dependent upon it, circumventing it will typically require the rogue user to circumvent the whole cryptographic functionality, and hence not use the provided mechanisms. In controlled environments, such as a corporation, it is not infeasible to restrict the user's resources, and hence requirements R1, R3, and R8 can be efficiently met. This is not the case for key encapsulation mechanisms as will be described later, or even for those key escrow mechanisms that are less dependent on specific key establishment protocols.

When assessing key escrow mechanisms it is useful to make a further distinction between those that at least sometimes require the on-line participation of a TTP acting as an escrow agent and which assists in the session key establishment process (such as the JMW scheme [8]), and those that do not. In the first category fall mechanisms such as the ones where the escrow agent acts as a *key distribution centre* (KDC) or *key translation centre* (KTC), and in the meantime escrows all the keys that the users establish. For this class of key escrow mechanisms, an on-line server, which must typically be able to deal with a large number of simultaneous requests, will be involved in all, or at least a significant number of, the key establishment processes. The agent's on-line par-

ticipation makes these mechanisms the most difficult to circumvent, and rogue user attacks are difficult to mount. Furthermore, they give the agents more control over the KR information than with other mechanisms, a scenario that fits the business model (assuming that agents are internal to, and managed by, the organisation). These properties are particularly relevant to requirements R1, R3, and R7, which this type of key escrow mechanism can efficiently satisfy. Compromise of this server, however, would have unpredictable consequences. An adversary in control of the server functionality would typically be able to recover all the established keys and decrypt communicated or archived data. Moreover, agent unavailability would mean that users are unable to encrypt data, hence requirement R2 will not always be satisfied. Thus, protection of the server against unauthorised use and denial of service attacks becomes a fundamental issue.

In the second category fall mechanisms that are less dependent on an on-line escrow agent. For these mechanisms, the user escrows certain key-related information, typically during the initialisation phase, which enables the agent to recover session keys subsequently generated by the user. As an example, consider a scheme where the sender encrypts the session key under the recipient's public key, while the corresponding private key is escrowed with the user's agent. Although there is no need for the agent to be on-line for these mechanisms, avoiding any availability requirements, they are no more flexible than mechanisms of the first category, since they also typically need an infrastructure to assist in all the cryptographic computations. For instance, in the given example users need to possess each others' valid certificates. This means that a complete certificate management scheme is required, including an inter-organisational public-key infrastructure (PKI) (e.g. a certificate repository, and means to generate and manage certificate revocation lists).

Cost (requirement R12) is another important consideration, especially in a commercial environment. Although deployment costs might be acceptable, long term administrative costs cannot be ignored. Key escrow mechanisms require provisions to protect the escrowed key material, and in that respect are potentially expensive. Although the cost might be significantly reduced if an external agent is used, as previously mentioned there are clear potential disadvantages of such an approach.

Summarising the above, key escrow mechanisms can efficiently satisfy *non-circumventability* (R1), especially in cases where the key recovery agent controls the key establishment process. As a result *agent completeness* (R3) and *policy compliance* (R8) will also be satisfied. Problems, however, might arise if an on-line agent is used for the key establishment process, whose unavailability can affect *user completeness* (R2) and *user acceptability* (R6). Finally, key escrow mechanisms are likely to suffer from interoperability problems, and hence not satisfy requirement R10, and their cost (R12) cannot be considered negligible.

5.3 Assessment of key encapsulation mechanisms

When used in a corporate environment, the majority of key encapsulation mechanisms appear to be more flexible than key escrow schemes. Being independent of the key establishment technique means that protocols can easily be adapted to them and, unlike key escrow schemes, they are unlikely to suffer from key management related interoperability problems and hence, they can satisfy requirement R10. The data encryption keys will typically be encrypted under KRM-specified public key(s), with no restrictions on their nature or generation (requirement R9). Interoperability, however, can be affected in communication sessions by the interaction the mechanism might require with the remote party prior to generation and/or for verification of the KR information. This might include mechanism-specific public keys that the originating party needs to generate the KR information, or any verification checks the peer is required to perform on the received KR information prior to decryption.

Unlike some key escrow mechanisms, a key encapsulation infrastructure would typically not require a high powered on-line server, as there is no need for on-line interaction during the KR information generation process (of course, such a server may be necessitated by other key management requirements). Therefore, user acceptability (requirement R6), as far as the availability of the KRM is concerned, will always be satisfied. Deployment of such a mechanism in a corporate environment, however, might necessitate checks on transmitted KR information to ensure users comply with the company's policy (requirements R3, R4, and R8). This is because key encapsulation mechanisms are vulnerable to *cut-and-paste* attacks, where a rogue user can alter or delete the KR information after its generation to disable subsequent key recovery by an authorised entity. To prevent this, and hence satisfy requirements R1, R3 and R7, authorised entities could run checks on the generated KR information to ensure that the KRM has been properly used (assuming that such checks are supported by the KRM); such checks could be made at random, or only if rogue activity is suspected.

One way of preventing rogue user attacks is to check intercepted KR information. If the intercepted information is invalid, the transmitted data could be prevented from leaving the organisation's domain. Rogue user attacks can also be prevented by requiring the validation of the KR information by the receiving party prior to decryption of the received data. This latter solution, however, requires trust in the receiving entity, which is not always the case, especially if the latter is not within the company's domain. Thus, a drawback of key encapsulation mechanisms is that, in order to ensure that the mechanism is not circumvented, there is a potential need for on-line checks on the generated KR information. This checking can prevent both single rogue user attacks that are not prevented by the mechanism itself, and double rogue user attacks where the colluding entities agree to make use of the organisation's cryptographic mechanisms but bypass the key recovery process by tampering with the key recovery information.

On-line checks on the key recovery fields might be trivial if the key recovery

agent operates within the company's domain. In that case a server that resides behind the organisation's firewall can be used for this purpose. However, verification of the KRF might be harder if the agent is external to the organisation. Only checks can be performed in this case, and then only if the information is intercepted during its transmission, or if all communications are routed through an agent's server. This is because the data may already have left the company's control and have reached their destination. Thus, in this case, *detection* of rogue users remains possible, but the capability for *prevention* is much more limited. Of course, it may be the case that detection is sufficient, for actions can be taken against rogue users as soon as a single instance of system misuse is detected. However, in situations where misuse must always be prevented, it will probably be appropriate to have an internal rather than an external agent.

Note that existing key recovery mechanisms typically do not support third party validation of KR information, a property that can help meet requirements R1, R3, R4, and R8. A model designed for this purpose was proposed in [17] but, as described in [18], the proposed scheme suffers from superencryption attacks. Simple techniques can be employed, e.g. demanding the encryption of a data field known to the agent with the key used for encrypting the rest of the data. Decrypting this field with the key contained in the KRF would give the agent an indication as to the validity of the KRF. Even with these techniques, however, mechanisms can still be vulnerable to rogue user attacks. Probably the most effective solution is to decrypt the data, and search for expected patterns that should, or should not (in the case of malicious software), be present in the transmitted data.

Key encapsulation mechanisms are even more vulnerable to cut-and-paste attacks when the mechanism is used on encrypted archived data. It would typically be too costly to check the validity, either on-line or off-line, of every KRF produced for every encrypted file. Therefore, a rogue user could tamper with the KRF by simply deleting or modifying the field after its generation, thereby disabling any authorised KR attempt. This will typically constitute a breach of policy (requirement R8), and action could be taken against that employee, who might even lose his job. However, from the company's perspective the data are lost, and hence the KRM has failed. In such a case, it would be more appropriate to use a key escrow mechanism which will give the company the ability to have more control over the generated keys.

Another relevant issue is that, in key encapsulation mechanisms, the management of the keys is left with the user. In hierarchical environments this property might cause problems if different agent public keys are used to protect KRFs for different levels of classification of information. The user will have to decide which key he should use for encrypting the key recovery information, depending on the sensitivity of the data. This might lead to confusion, and even accidental or deliberate misuse of the KRM. This is not always the case with key escrow mechanisms, where the escrow agent can manage the generated data encryption keys.

Key encapsulation mechanisms are not inherently more secure than key escrow mechanisms. Although for the latter there is a need to protect all the

escrowed key material, and unauthorised access to it would typically give the attacker access to data encryption keys, the compromise of the agent's private decryption key in a key encapsulation mechanism would have the same unacceptable consequences.

Finally, key encapsulation mechanisms appear to have potentially lower management costs than key escrow schemes (requirement R12). For such a mechanism there only needs to be a cryptographic infrastructure. There is no need for on-line participation of the agent, which potentially requires a high powered server (unless the corporation demands on-line checks on generated KRFs), or for the secure storage of all the escrowed keys. However, these cost estimates might be altered, depending on the mechanism's implementation. For instance, if it is decided to deploy user smart cards, then the cost of issuing a card for each employee may not be negligible.

Summarising the above, key encapsulation mechanisms are more susceptible to rogue user attacks than key escrow schemes, and thus they do not always satisfy the *non-circumventability* requirement (R1). Susceptibility to rogue user attacks is likely to have the same effect on *agent completeness* (R3), *agent acceptability* (R7), and *policy compliance* (R8). However, as key encapsulation mechanisms can typically work with any key establishment protocol and key encryption algorithm, they are quite *flexible* (R9), and they cause less *interoperability* (R10) problems than key escrow mechanisms. Finally, their cost of deployment (R12), although it cannot be considered negligible, is likely to be less than key escrow mechanisms.

6 Distinguishing between communicated and archived data

The above analysis of KRMs in a business environment has not considered the target data. There are certain issues, however, that need to be addressed as far as the target of the KRM is concerned. This arises from the fact that there are different requirements for KRMs for archived data and KRMs for communicated data.

The majority of existing key recovery mechanisms were designed for use with communicated data, and with the objective of giving access to LEAs. Giving user access was typically not a design requirement mainly because users would not benefit from such a property. As mentioned in [10], "there is hardly ever a reason for an encryption user to want to recover the key used to protect a communication session". If the session key is lost during an encrypted session, a new session can be established and a new key can be negotiated. This, however, does not rule out business demands for access to encrypted communications.

With communicated data, interoperability of the deployed KRMs is the most important requirement. Otherwise, use of dissimilar mechanisms might prohibit the establishment of a secure communication session.

With archived data, the requirements are rather different, making the dis-

tinction essential. With archived data, the focus of the design of the KRM should also consider the users' needs for recovery of data. In other words, apart from fulfilling third parties' needs, the KRM should also be the users' means for access to lost keys. It would be a waste of communications resources and processing power if the user had to contact his agent whenever he wants access to his encrypted data or requests recovery of a lost key. Moreover, interoperability is no longer an issue, as encryption of archived data will typically only involve one entity, and hence only one KRM. As a result of this, however, the mechanism will be more susceptible to rogue user attacks where the user might deliberately delete or alter the generated KR information.

Electronic mail is a special case because it has the characteristics of both communicated and archived data. If the decryption key for an encrypted e-mail is lost, access to the email will be infeasible unless the sender re-sends the message. The ability to recover keys used for encrypting e-mail could be of a potential benefit to the user.

Therefore, the differences that necessitate the distinction between KRMs used for communicated data and those used for archived data are as follows.

- **Interoperability**. While interoperability is a critical requirement for KRMs used for encrypted communicated data it is of no importance to KRMs for archived data.
- **Susceptibility to rogue user attacks**. In encrypted communications, attacks where a rogue user tampers with the generated KR information can be prevented by requiring the receiving party to verify it prior to decryption (although this might not always be an efficient countermeasure against these attacks). This is not possible with archived data, however, as during a typical encryption of archived data there will be only one entity involved.
- **Users' ability to recover their keys unaided**. While users typically do not benefit from being able to recover keys used for their encrypted communications, the situation is rather different for archived data.

7 Key recovery mechanisms for communications

The issues surrounding key recovery for communicated and archived data are somewhat different, as previously mentioned. In this section we look at the requirements that surround KRMs for communicated data, and investigate the applicability of the two main categories of KRMs. In the next section a similar analysis is performed for encrypted archived data.

7.1 Requirements

As far as KRMs for *communicated data* are concerned, there is clearly a need for the deployed mechanism to inter-operate with the one used by the peer that

might reside inside or outside the company's domain. In other words, how likely is the mechanism deployed to prohibit the establishment of secure communications with an entity incapable of dealing with the specific KRM? When KRMs are used for communicated data it is crucial that both communicating parties are capable of dealing with each other's mechanism requirements. As far as KR is concerned, there are two main interoperability scenarios for a communication session between entities A and B, assuming that entity A resides within a company's domain whose policy demands use of a KRM for all encrypted communications.

1. The two entities A and B make use of KRMs KRM_A and KRM_B respectively. If KRM_A and KRM_B are dissimilar, the two parties might not be aware of the semantics of each others' KRM, including the requirements for KR information generation, and the format of the fields that carry it. As a result the two entities might not succeed in establishing a communication session, and, even if they do, any KR information fields in the incoming traffic might cause a problem.

2. If entity A uses KRM_A and B does not make use of KR, will the two entities be able to communicate securely, and will the first entity be able to make use of its KRM? Moreover, will entity A, and more specifically its company, accept encrypted incoming traffic that does not make use of a KRM?

Assuming that KR is required for all encrypted communicated data, it is clear that interoperability restrictions on its use would be undesirable. A couple of schemes that promise to provide interoperability between dissimilar mechanisms have been proposed by IBM [6] and the Key Recovery Alliance [15]. As far as the latter is concerned, although it promises to promote interoperability between dissimilar schemes, in many cases it fails to do so. Furthermore, the proposed scheme is vulnerable to rogue user attacks [19]. Another slution to the interoperability problem is provided in [14].

Therefore, the requirements that should be fulfilled by a KRM specifically deployed for communicated data, are as follows.

1. The KRM should give authorised entities the means to recover session keys used to encrypt the exchanged data, as well as keys used to encrypt communications-related data. This is vital for the company, as it may legitimately want to keep track of certain outgoing communications, to use this information for non-repudiation purposes in the case of a dispute, or even monitor incoming traffic. Thus, individual keys used by employees, or established during a communication session, must be recoverable using a KRM. This includes keys generated by entities outside the company and used to protect messages sent to the company.

2. The KRM should provide the ability for on-line and real-time interception and decryption of the communications. That is, if suspicious communica-

tions take place between an employee and an outsider then the management should be in position to monitor them. This includes the ability to verify the validity of the KR information generated by the employee, or even decrypt the communicated data at the time they cross the company's domain.

3. The KRM should be interoperable. That is, use of the mechanism should not prohibit the establishment of a secure communication session. Problems are likely to occur if the two communicating entities use entirely different mechanisms. In another scenario, one of the two parties might not make use of key recovery at all. As a consequence, the receiving party might discard the received data because it is simply not aware of the semantics of the additional KR fields.

The above issues cover both communications within an intranet, and with the outside world. In an ideal scenario the interoperability and portability of the mechanism among various domains should not be affected by government restrictions. This is essential in cases where the organisation's communications or boundaries span regions and therefore multiple national restrictions might apply to the KRM and the cryptographic products used.

7.2 Applicability of existing KR mechanisms

As previously mentioned, most key escrow mechanisms depend on a common infrastructure. While this might not be a problem in a corporate environment, i.e. for intranet communications, it is a major drawback when using the mechanism for communications that span company domains. The main reason is that there would be a need for the peer to deploy the same, or at least compatible, key establishment protocols, or the establishment of secure communications is likely to be prevented.

Use of a key encapsulation mechanism, on the other hand, would typically require the addition of data fields to the communicated data, to carry the KR information. A communicating party not wishing to incorporate KR can simply modify its existing infrastructure so that additional fields are discarded without being interpreted. This allows secure communication to take place, and provides KR for the communicating party that wants it. However, such a configuration allows rogue users to mount cut-and-paste attacks, as the other party would not check the validity of the KRF.

There are, however, certain issues affecting the above interoperability scenarios which depend on the nature of the KRM, and which apply to both key encapsulation and key escrow mechanisms, independently of the key establishment protocol. Specifically, in considering certain interoperability requirements, we need to divide KRMs into two classes depending on how the KR information is generated. The needs of the underlying key establishment protocol and the dependence of the KRM on it are not taken into account.

In the first class of KRMs are those that do not require co-operation with the peer to generate KR information. Each party generates KR information

merely for its own needs, and the two entities might thus be able to make use of distinct mechanisms. If neither party requires verification of the generated KR information (this is a matter of policy within each party's domain) then there is no interoperability issue.

In the other class are mechanisms where the sender generates the KR information both for himself and for the receiver. This is typically required in single message communications such as email or file transfer, where the remote party requires KR for the data encryption key. For this class of mechanisms the two parties have to use identical or at least interoperable mechanisms. In other words, the receiving entity should be able to provide the sender with the cryptographic material required by the sender's KRM for the generation of the KR information. In the case of key encapsulation mechanisms this will typically include certified public keys under which the sender will encrypt the KR information.

Policy demands might also affect the interoperability of KRMs. If the company requires that every incoming encrypted message should have a KR capability, e.g. so that malicious software checks can be run on the content before the message reaches the ultimate receiver, it is clear that the sender must generate KR information interpretable by the receiving organisation. Furthermore, if the corporate policy demands verification of the KR information prior to decryption of the received data, e.g. for non-repudiation purposes, the receiving party should be able to handle the specific KRM used for the generation of this information.

8 Key recovery mechanisms for archived data

In this section the requirements of KRMs for archived data and the applicability of existing mechanisms are examined.

8.1 Requirements

As far as *archived data* are concerned, there will typically only be one entity involved, which stores and retrieves data at distinct points in time [20]. Therefore, the keys used by this entity to encrypt stored company data are likely to be possessed only by this entity. Hence, decryption of archived data by a third party will be infeasible unless the third party either contributes to the key generation process, and hence has access to the generated data encryption keys, or is provided with material to enable key recovery.

Unlike KRMs for communicated data, there will be no interoperability requirements for a KRM used for encrypted archived data, as it will not interact with any other KRMs. However, because of this, some KRMs will be subject to single rogue user attacks, as there will be no other entity that can check the validity of the generated KR information. A rogue entity making use of the KRM might be able to manipulate the generated KR information, and thereby disable any subsequent KR. While this kind of attack is typically prevented in

a data communication environment by requiring the receiving party to verify the KR information, this is infeasible for archived data encryption. Thus, non-circumventability becomes a crucial requirement for KRMs for archived data. That is, the user should not be able to disable KR by altering or deleting the generated key recovery information.

8.2 Applicability of existing KR mechanisms

We start by making the same distinction as in section 5.2, that is, between mechanisms that require the existence of an on-line TTP and those that do not.

1. Mechanisms that require the on-line participation of the escrow agent appear to be the most secure from the non-circumventability point of view. As they are typically integrated within the cryptographic mechanisms, circumventing the KRM would require the user to avoid making use of the cryptographic mechanisms, and hence make no use of encryption. Yet, as already mentioned, these mechanisms suffer from the requirement for a high-powered server that will participate in the key generation process.

2. KRMs that are less dependent on on-line agent participation are weaker as far as circumventability is concerned. Within this class, however, fall some mechanisms that do not have this problem. For example, consider a KRM where a user escrows with his agent a Master Key, which he subsequently uses, together with some additional data that the agent is assumed to know in advance, to compute the session key. Such a scheme does not require the existence of an on-line TTP. Yet, the TTP has direct access to the generated key, and this is the characteristic distinguishing it from other mechanisms of the same category.

Both these types of mechanisms suffer from the problems previously mentioned regarding key escrow mechanisms. That is, deployment of such a scheme would require the existence of large storage devices for the administration of the escrowed keys, which require the use of strong security mechanisms for their protection.

Key encapsulation mechanisms used for encrypted archived data have the advantage that neither an on-line agent nor storage of any KR material by the user's agent is required. All the user needs to do is encrypt the session key together with some mechanism-specific credentials under the public key of the user's agent, and attach the generated field to the encrypted data. However, these mechanisms are vulnerable to cut-and-paste attacks where a rogue user alters or deletes the generated KR information preventing access to the encrypted data by authorised entities. So, as far as circumventability is concerned, these mechanisms appear to be more vulnerable than key escrow.

Another issue is that most existing KRMs, especially key encapsulation schemes, do not offer the user the ability to recover the keys himself, hence forcing him to use the agent to recover the keys on his behalf. The reason is that, in most cases, the user has no means to recompute the data encryption key.

For example, consider a key encapsulation mechanism where the KRF contains the data encryption key encrypted under the agent's public key. Whenever the user wants to access an encrypted file using the KRM, e.g. when the user loses the key, he has to ask the agent to recover the decryption key from the KRF. Problems will then arise when the user works off-line, or if there is no connection with the agent because of a network failure; the user will be unable to use the KRM to recover his keys because of agent unavailability. This property might make a KR scheme much less acceptable to the users.

A mechanism that does not suffer from the above problems, i.e. a mechanism not requiring an on-line server, the storage of KR information, or escrowed keys, and that would not be vulnerable to cut-and-paste attacks, would be of considerable potential value. Moreover, it would also be helpful if the mechanism could enable the user to recover keys on his own without intervention of the agent.

9 Conclusions

An analysis has been made of the requirements for KRMs applied in a business environment, and the applicability of existing mechanisms was investigated. As there is no panacea to the key recovery problem, careful analysis of the business needs is necessary to identify appropriate solutions. A further distinction was made between requirements for KRMs for communicated data, and requirements for KRMs for archived data, and it was shown that mechanisms providing KR functionality for communicated data might not be ideal for encrypted archived data, and vice versa. A KRM that does not suffer from the problems identified in the existing mechanisms would be of potential practical benefit.

References

[1] D.E. Denning and D.K. Branstad. A taxonomy of key escrow encryption systems. *Communications of the ACM*, **39(3)**:34–40, March 1996.

[2] M. Smith, P. van Oorschot, and M. Willett. Cryptographic information recovery using key recovery. *Computers & Security*, **19(1)**:21–27, 2000.

[3] The White House. Directive on public key encryption management, 1993.

[4] D.E. Denning. *Information Warfare and Security*. Addison Wesley, 1998.

[5] M.R. Smith. *Commonsense Computer Security*. McGraw-Hill, 1994.

[6] IBM SecureWay. Towards a framework based solution to cryptographic key recovery. http://www-4.ibm.com/software/security/keyworks/library/.

[7] N. Jefferies, C. Mitchell, and M. Walker. Trusted third party based key management allowing warranted interception. In *Proceedings: Public Key*

Infrastructure Invitational Workshop. MITRE, McLean, Virginia, USA, **NISTIR 5788**, September 1995.

[8] N. Jefferies, C. Mitchell, and M. Walker. A proposed architecture for trusted third parties. In E. Dawson and J. Golic, editors, *Cryptography: Policy and Algorithms — Proceedings: International Conference, Brisbane, Australia*, pages 98–104. Springer-Verlag (LNCS 1029), Berlin (1996).

[9] S.T. Walker, S.B. Lipner, C.M. Ellison, and D.M. Balenson. Commercial key recovery. *Communications of the ACM*, **39(3)**:41–47, March 1996.

[10] H. Abelson, R. Anderson, S.M. Bellovin, J. Benaloh, M. Blaze, W. Diffie, J. Gilmore, P.G. Neumann, R.L. Rivest, J.I. Schiller, and B. Schneier. The risks of key recovery, key escrow, and trusted third party encryption. http://www.cdt.org/crypto/risks98.

[11] L.R. Knudsen and K.M. Martin. In search of multiple domain key recovery. *Journal of Computer Security*, **6**:219–235, 1998.

[12] J.G. Nieto, K. Viswanathan, C. Boyd, and E. Dawson. Key recovery system for the commerical environment. In E. Dawson, A. Clark, and C. Boyd, editors, *Information Security and Privacy – ACISP 2000*, pages 149–162. Springer-Verlag (LNCS 1841), Brisbane, Australia, 2000.

[13] National Institute of Standards and Technology. requirements for key recovery products, November 1998. Available at http://csrc.nist.gov/keyrecovery/.

[14] K. Rantos and C.J. Mitchell. Key recovery scheme interoperability – a protocol for mechanism negotiation. In B. Honary, editor, *Cryptography and Coding - Proceedings of the 8th IMA International Conference, Cirencester, UK, December 2001*, pages 268–276. Springer-Verlag (LNCS 2260), Berlin (2001).

[15] S. Gupta. A common key recovery block format: Promoting interoperability between dissimilar key recovery mechanisms. *Computers & Security*, **19(1)**:41–47, 2000.

[16] D.P. Maher. Crypto backup and key escrow. *Communications of the ACM*, **39(3)**:48–53, March 1996.

[17] E.R. Verheul and H.C.A. van Tilborg. Binding Elgamal: A fraud-detectable alternative to key escrow proposals. In W. Fumy, editor, *Advances in Cryptology–EUROCRYPT'97*, pages 119–133. Springer-Verlag (LNCS 1233), Berlin (1997).

[18] B. Pfitzmann and M. Waidner. How to break fraud-detectable key recovery. *ACM Operating Systems Review*, **32(1)**:23–28, 1998.

[19] K. Rantos and C.J. Mitchell. Remarks on KRA's key recovery block format. *Electronics Letters*, **35**:632–634, 1999.

[20] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1997.