Remarks on KRA's Key Recovery Block Format

Konstantinos Rantos* and Chris Mitchell

Information Security Group,

Royal Holloway, University of London,

Egham, Surrey TW20 0EX, UK.

K.Rantos@dcs.rhbnc.ac.uk, C.Mitchell@rhbnc.ac.uk

26th January 1999

Abstract

The introduction of a plethora of key recovery (KR) schemes and the lack of a standard has led to interoperability problems between dissimilar mechanisms. To overcome these problems the Key Recovery Alliance (KRA) has proposed a common Key Recovery Block (KRB) format. This paper identifies some cases that the mechanism fails to achieve its objectives.

Keywords: key recovery, interoperability.

1 Introduction

The deployment of cryptographic mechanisms for data confidentiality has resulted in the introduction of key recovery (escrow) services. These will poten-

*This author's work is supported by the European Commission (TMR Marie Curie Research and Training Grant ERBFMBICT983274).

tially be used both by companies as well as by individuals who do not want to lose their encryption keys and therefore have no access to their data. Many KR schemes have been proposed so far by industry and academia. An overview of some of these mechanisms is given in [1]. The existence of various KR mechanisms, with different characteristics and more important not compliant to a standard, has led to interoperability problems.

Consider the case where an entity A using a KR mechanism KRM_A wants to communicate with an entity B using a KR mechanism KRM_B. Assuming that both entities wish to establish a secure communication with KR capability, both of them have to make use of their KR mechanisms. The problem that arises in such a case is whether these two mechanisms can interoperate, i.e. whether "KRM_A can set up a KR enabled cryptographic association with KRM_B"[2]. The problem becomes more visible when one or both of the entities want, or are obliged due to policy restrictions, to validate the KR information received from the peer entity prior to decryption of the ciphertext. That is, both entities want to make sure that the peer entity makes proper use of the KR mechanism.

Due to the increasing number of KR schemes commercially available, and not compliant with a standard, it will be infeasible for KRM_B to know the semantic details of all the mechanisms and be able to parse the KR information received from A. The KR information in many KR mechanisms is within a Key Recovery Field (KRF). In [2] two main types of KRFs are identified. The first one is found in a key *escrow* mechanism and basically contains the session key encrypted with the recipient's public key. Such a mechanism requires access to the recipient's

private key which in turn enables the recovery of the session key. In a key *encapsulation* mechanism, however, the session key is encrypted with the public key(s) of the Recovery Agent(s). This type of mechanism allows the Recovery Agent to directly recover the session key.

The differing requirements of the available KR mechanisms and the varying formats of their KRFs (when one is used) have led to interoperability problems. In many situations the KRFs cannot be interpreted and therefore cannot be used at all. This might lead to suspension of communications as in many cases the communicating entities' policies might explicitly require validation of the KR information prior to decryption. Thus, it becomes obvious that a mechanism is needed that will enable the entities to validate the KR information received.

2 KRA's model

To overcome the interoperability problems described in the previous section the Key Recovery Alliance (KRA) has proposed a mechanism which, when adopted, enables communications between dissimilar KR mechanisms. The mechanism introduces a Key Recovery Block (KRB) that "serves as a container for a single KR mechanism-specific KRF" [2]. The KRB according to [2] achieves two main objectives; it *"provides a means to identify the KR mechanism used to construct the KRF"*, and, *"provides a range of validation techniques, including those that allow validation of the KR information in generic, KR mechanism-independent ways"*.

2.1 Description of the KRB and its validation mechanisms

In this section brief descriptions of the KRB and the KRB validation techniques are given. For full details and explanations of the mechanism see [2]. The KRB consists of the following fields:

- **KRB Version Number**.
- **KRB Length**: The number of words in the entire KRB.
- **Object Identifier (OID) for KRF**: The OID for the KR mechanism used to generate the KRF, as registered with a central authority.
- **Reserved**: A 16-bit field reserved for future use.
- **KRF Length**: Number of words in the KRF.
- **Key Recovery Field**: The proprietary KRF whose format and contents are indicated by the OID.
- **Validation Field Type**: Identifies one of the following seven techniques used to compute the Validation

    1. **NONE(Type 0)**: No Validation Field Value (VFV) is calculated; KRF validation is unnecessary at the decrypting side.
    2. **SEMANTIC(Type 1)**: No VFV is calculated; the KRF should be validated semantically using the mechanism-specific algorithm.
    3. **PROTOCOL(Type 2)**: No VFV is calculated; the KRB need not be checked for validity since the carrier protocol provides integrity protection for the KRB.

4. **CONF-HMAC-SHA-1-96(Type 3)**: The VFV is a hash of the KRB using HMAC and SHA-1 and the confidentiality key associated with the KRF.
5. **CONF-HMAC-MD5-96(Type 4)**: The VFV is a hash of the KRB using HMAC and MD-5 and the confidentiality key associated with the KRF.
6. **INTEG-HMAC-SHA-1-96(Type 5)**: The VFV is a hash of the KRB using HMAC and SHA-1 and the integrity key associated with the KRF.
7. **INTEG-HMAC-MD5-96(Type 6)**: The VFV is a hash of the KRB using HMAC and MD-5 and the integrity key associated with the KRF.

- **Validation Field Length**: Number of words in the VFV.
- **Validation Field Value**: It is calculated over the entire KRB.

3 Problems identified in this mechanism

The mechanism proposed by the KRA, as mentioned above, promises to promote interoperability between dissimilar mechanisms. However, two problems can be identified within this mechanism. The first relates to difficulties arising from the generation of proprietary KRFs while the other concerns the fact that in many cases the mechanism fails to provide interoperability.

3.1 KRF generation

The paper makes the silent assumption that a KRF has already been generated and therefore it is always available for the KRB generation. However, this is feasible only in a very limited number of combinations of the various KR schemes. In the case of non-interoperable mechanisms, the sender might not be able to generate a KRF because the receiver's end does not fulfill the requirements of the sender's KR mechanism. Consider the case, in its simplest approach, where the KRF generation requires the encryption of the secret session key (data encryption key) using the public key of the receiver (because the sender is using a key escrow mechanism). If the receiver does not have a public key because he is not using key escrow but a key encapsulation mechanism (or a public key that he might have does not fulfill the requirements of the sender's KR mechanism) the sender cannot oblige the receiver to get one. Therefore the sender cannot generate a KRF.

3.2 Interoperability issues

Among the Validation Field Types proposed in [2] there are five (types 2–6) which provide validation of the KRB. According to [2], using the generic validation mechanisms supported by the KRB, entity B would be able to validate the proprietary KRF sent by A and vice versa, "even though B did not understand how to parse the KRF". However, the placement of the KRF within the KRB does not enable this validation. The receiving entity still has to parse the KRF to check its validity. The integrity, confidentiality, and/or authentication of the

KRB and therefore of the content of the KRB, the KRF, does not guarantee the latter's validity.

As a consequence, the validation techniques proposed are vulnerable to a single rogue user scenario. Consider the case where two parties communicate using dissimilar (non-interoperable) KR mechanisms. Sender A generates a KRF for the receiver B who is not able to verify the proprietary KRF using the method required by A's KR mechanism because he is using a dissimilar KR mechanism. Assuming further that A is a rogue user, then the following scenario might take place: A generates a non-valid KRF. However, the Validation Field Value is generated using the valid session key (validation field types 3-6) which the receiver knows in advance. A genuine receiver B will validate correctly the KRB, as this was correctly generated by A. However, the validation technique and therefore the validation of the KR information has just failed because the KRF is not genuine and B has no means to verify that (we assume that the two mechanisms are not compliant and therefore B does not know the semantic details of KRM_A). Thus B will not be compliant with his policy which requires validation of the KRF prior to decryption of the encrypted data.

Moreover a Key Recovery Agent that A is associated with will not be able to recover the key. This is because the KRA will not try to recover the key using the KRB but using the KRF contained in it. However, the KRB was accepted as valid by B. In other words the KRB does not play the main role of the KRF which is to give information on the session key. It is only a mechanism that can provide integrity, confidentiality, or/and authentication but it is not a

"mechanism for verifying the validation of the enclosed KRF" [2, page 7]. There is no way to recover the key from the KRB using only the information provided by the KRB itself (excluding the KRF because this can be manipulated only by the users that deploy the same product). In the case of Validation Field Type 7 the same problem holds, as the Validation Field Value is a digital signature on the KRB which can carry a non-valid KRF. The only types that do not suffer from this problem are types 0 and 1 which demand "no validation" and "mechanism specific validation" respectively.

Therefore the solution proposed here does not achieve one of the two major objectives mentioned above, which is to provide "the ability to validate the KR information in a way that does not require knowledge of the exact semantic properties of the KRF". The solution fails to achieve its objective in situations where it is most desirable, i.e. in environments where there is a lack of trust.

The mechanisms where the KRF comprises the key exchange block do not suffer from the above problem because in such a case the KRF has to be processed to obtain the decryption key. However, these mechanisms face the problem where the sender cannot generate a KRF and if one is generated the receiving end might not be able to parse it and get the session key. The KRB proposed does not offer an alternative to this situation since the receiving end cannot obtain the session key using interoperable components.

4 Conclusions

This paper has looked at two problems inherent in the KRA proposal for interoperability of dissimilar key recovery mechanisms. The mechanism proposed offers an efficient way of carrying the KRF with confidentiality, integrity, and/or authentication but not a way of validating the KRF itself. Thus the proposed scheme fails to achieve one of its main objectives.

References

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

[2] Sarbari Gupta. A common key recovery block format: Promoting interoperability between dissimilar key recovery mechanisms. A document prepared for the Key Recovery Alliance, http://www.kra.org, May 1998.