Entrust Technologies White Paper

# The Need for Separate Key Pairs for Symmetric Key Transfer and Digital Signature

Author: Warwick Ford
Date: February 1994
Issue: 1.0

© Entrust Technologies, 1997. All rights reserved.

The RSA cryptosystem has the interesting property that one key pair can, in theory, be used for both encryption (for example, for transferring a symmetric key) and for digital signature purposes. For example, if parties A and B want to communicate securely, and B has an RSA key pair, A can send an encrypted symmetric key to B by encrypting it under B's public key. Using the same key pair, B can sign a message to A; B generates the signature using B's private key and A verifies the signature using B's public key.

However, if one looks more closely into the full range of issues surrounding key management, it becomes apparent that such double-use of the one key pair is impractical. Even if RSA is the only algorithm used for symmetric key transfer and digital signature purposes, a party needs to have a separate key pair for each purpose. The reasons are summarized below.

Looking first at digital signature key pairs, the following key management requirements arise:

(a) In order to support non-repudiation, the private key of a key pair used for digital signature purposes must be stored, for its entire life, such that no party other than its assignee can possibly get access to it. It is commonly recommended, and sometimes mandated (for example, in the new ANSI X9.30 standard on public key techniques in banking), that a digital signature private key never leave the device in which it is used — the key is created, used, and destroyed within the one secure module.

(b) A digital signature private key is never backed-up — if it is lost, a new key pair is simply generated for the signer. Back-up would jeopardize the requirement in (a) above.

(c) A digital signature private key is never archived — there is no need for this, and archival would also jeopardize the requirement in (a) above.

(d) A digital signature public key generally *does* need to be archived. This key may be needed to verify old signatures at an arbitrary time after the corresponding private key has ceased active use.

(e) A digital signature private key must be securely destroyed when its active life terminates. If its value is disclosed, even a long time after it is no longer actively used, it may still be used to forge signatures on old documents.

Looking now at key pairs used for symmetric key transfer, different key management requirements arise:

(a) A key transfer private key often needs to be backed-up. This is because such back-up may be the only way to recover encrypted information. If

a key is lost (for example, due to equipment failure) it is not acceptable that all information held encrypted under that key also be lost.

(b) A key transfer private key may need to be (securely) archived. If information is stored in encrypted form for an indefinite period, it is necessary to ensure that the decryption key can be recovered at arbitrary times in the future.

(c) A key transfer public key never needs to be backed-up or archived. If the public key is lost, a new key pair can be established.

(d) A key transfer private key does not need to be securely destroyed when its active life terminates. On the contrary, points (a) and (b) above imply that it should *not* be destroyed.

Clearly, the above two sets of requirements are in major conflict. If one tried to use the same key pair for both digital signature and key transfer, it would be impossible to satisfy all the requirements.

In addition, the following arguments for using distinct key pairs for the two purposes can arise:

(a) There may be a need for different cryptoperiods for the two key pairs. For example, suppose a shared-purpose key pair is used much more frequently for key transfer than for digital signature. The key pair needs to be updated frequently because of the encryption requirements. However, it is undesirable to update digital signature keys very frequently, because of the certificate archival requirements.

(b) Not all public-key algorithms have the RSA property. For example, the DSA algorithm can be used for digital signature but not key transfer. If future algorithm flexibility is to be accommodated, it is better to design a system now assuming different algorithms are used for digital signature than for key distribution. This means building in an assumption of different keys for the different purposes.