IMPLEMENTATION OF KEY ESCROW WITH KEY VECTORS TO MINIMISE POTENTIAL MISUSE OF KEY.

Professor W.J. Caelli
Professor D. Longley
Information Security Research Centre
Queensland University of Technology
GPO Box 2434
Brisbane QLD 4001
Australia

Keywords: key recovery, key escrow, key vector, key tagging, key management, authentication, signatures.

1. ABSTRACT

The concept of key escrow/key recovery schemes was introduced to overcome the perceived problems of law enforcement agencies when criminal elements protect their transmissions with cryptography. It was argued that such law enforcement agencies must have the potential for real time access to all cryptographic keys used for such privacy purposes, in order to ensure that criminal activities may be monitored, usually under appropriate court orders, and to thus allow such agencies to fulfil their respective charters.

The concept met with significant opposition, on both civil liberty and technical grounds. One major concern is that cryptography is not merely used to hide confidential data, it is now widely used for personal and system authentication in addition to non repudiation of electronic transactions. The ubiquitous public key cipher RSA may thus be employed to encrypt a message, or encrypt a symmetric cipher's secret key, e.g. DES key, which in turns encrypts a message, or authenticate the user to a host system or electronically sign a transaction. Handing over one's RSA secret key to a public official not only implies trusting that official not to misuse the knowledge of secret transmissions, it also implies trusting that official not to masquerade as oneself to computer systems, and not to misuse one's digital signature. The problem arises because in public key cryptography the private key, usually known only to its owner, may serve two functions: decryption of an encrypted message sent to the key holder, and the creation of a digital signature for a message, originated by the key holder, to provide for integrity checks and a guarantee of non-repudiation.

The potential misuse of keys submitted for key recovery could have significant social and legal implications for the widespread usage of electronic signatures, even if the private key is only ever used to encrypt a further secret key for a single key cipher, since there is no longer one unique owner of the key. This is a major concern for any national and international electronic commerce scheme.

The problem lies with the concept of handing over knowledge of the private key part of a dual key pair in any public key scheme, or the secret key used for message authentication and integrity check purposes in an electronic payments system. A law enforcement agency may state that it requires the **use** of the secret key for a very specific purpose, i.e. to decrypt a set of transmitted messages or to obtain a session key used for that purpose. The question arises; is it possible to hand over some information which enables the law enforcement agency to undertake only well specified monitoring activities and have a high level of assurance that there is no subsequent illicit use of a specific key? Moreover, is it possible to hold that agency accountable for all privileges handed to it?

The proposed solution to this problem is in the 'locking' of a key to its permitted usage and stated function. This facility has been provided by a techniques known as "key vectoring" or "key tagging", developed originally for symmetric key ciphers in the banking industry. Even with this approach it is not possible to eliminate the requirement for some trusted officials. However, it is possible to clearly identify such officials, and hold them accountable for their actions.

Essentially, the deployment of key vectors aims at restriction on the use of keying information, for example, copies supplied to a law enforcement agency through a key recovery centre. In the proposed scheme such keying information may only be exploited by the use of a specified tamperproof device, or chip. The key vectors instruct the cryptographic sub-system which particular cryptographic operations are permitted with particular keying information, through the incorporation of the vector technology into a chip or a trusted computing base (TCB) software system. Such technology is exemplified by the Hewlett-Packard "International Cryptography Framework" architecture [1]. The legitimate use of the supplied keying information is thus sealed into it; any attempt to use the key for other purposes will be inhibited by the unique hardware device or system level software. The identity of this device itself may be sealed into the keying information, thus further restricting the use of the keys. (The term hardware device or chip is employed in this paper from now on but, as indicated above, it is also possible that a trusted software solution could be implemented.)

It is also possible, with key vectors, to place a level of granularity on the monitoring itself. It would be feasible, for example, to specify that only messages received from a given sender, or transmitted to a given receiver, over a specified date period, could be monitored. In addition the specified law enforcement chip may produce an audit log of all activities. It would not be feasible to store this log on the chip itself. However, a condensed or hashed version could be stored on the chip. The law enforcement agency could then be asked to produce the audit log, and its accuracy and completeness checked against the hashed values stored on the chip.

The question arises, how is the keying information supplied to the law enforcement agency? It is suggested that a central government agency, receives the key components from the appropriate key recovery bodies, and produces the key vectored information to be supplied to the law enforcement agency. The central government agency is necessarily confined to use another hardware device or chip. This device may be used in a low privileged mode for essential checking purposes, e.g. it can confirm that supplied keying material does contain the secret key of a given user, without revealing the key. Privileged operations, e.g. to produce keying information to be handed to the law enforcement agency, would be restricted to two or more trusted officials. An audit log of these privileged operations would be secured by a similar mechanism to that described for the law enforcement agency.

In the final analysis, someone must be trusted to act according to the law. The proposals contained in this paper severely limit the range of operations that can be performed on supplied keying material, minimise the number of officials that must be trusted and can ensure that all such officials will be held accountable for their actions.

2. INTRODUCTION

There have been few information security proposals in recent years that have caused as much controversy as the United States centred *"Clipper/Key Escrow"* debate, following the announcement of the Clipper Chip initiative in April 1993. It has been consistently difficult to separate issues such as civil liberties (suspicion of national security and law enforcement agencies) from the intricacies of proposed technical solutions, which themselves were sometimes unconvincing and incomplete due to pressures to maintain confidentiality. The debate on government policy and cryptography, even among senior academics, has occasionally veered to the paranoiac.

This paper does not provide any views on the politics, ethics or morality of key recovery but attempts to provide a technical solution to one of the problems associated with the concept of handing over one's cryptographic keys, either original or derived, to public officials. This problem is associated with the ubiquity of modern cryptography. The whole key escrow/recovery debate has been directed to the question of confidentiality, but modern cryptography is increasingly used for personal authentication, integrity and non repudiation of electronic transactions. A citizen **may** conceivably be prepared to allow some law enforcement agency to read his or her electronic mail, in the interests of national security. However, that same person would normally be somewhat more reluctant to hand over, to some unknown public servant, the right to impersonate him or her and thus, potentially at least, "go on a spending spree with one's bank account".

This situation has arisen because public key ciphers do more than merely encrypt messages. An RSA public/secret key pair may be used to encrypt a message, encrypt a DES key which in turn encrypts a message, authenticate the owner in a challenge response dialog or sign an electronic check. If I am compelled to hand over my RSA key, because it is used to encrypt DES session keys on my email, then I wish to be sure that the

recipient cannot use it for authentication or signatures. Of course, I could use different RSA keys for these various purposes. However, I will still have to be sure that my “friendly but possibly foreign public servant” was not simply using my key exchange keys to sign fraudulent electronic checks.

One possible solution would be the use of certificates indicating the purpose of the public (and associated secret key). When signing a message I would hand over a specific “signing certificate”. I could then repudiate any signed transaction that was not accompanied by such a specific signing certificate. If my friendly foreign public servant signed an electronic check with my key exchange, RSA key I could tell an independent judge that the public key could not be linked to me for signing purposes because my opponent in court could not produce a signing certificate for that public key. This approach might be less than satisfactory if:

- the friendly but foreign public servant did not come within my local jurisdiction, or
- certification bodies did not universally include the role of the secret key in the certificate, or
- certification bodies were subject to the influence of friendly foreign public servants or
- I had a judge who lacked a detailed knowledge of public key cryptography.

The fundamental problem is that I would have to prove that a secret key used for decryption (as part of the normal key pair), that I publicly acknowledged, was never used or intended for signing purposes.

The solution to the ubiquity problem, described in this paper, is to seal the function of the secret key with the key itself , before it is handed over to the law enforcement agency. Hence the key handed over can only be used by the law enforcement agency for the *legitimate* purpose, possibly as given in appropriate court orders, to compromise user’s privacy. This technique of key tagging, and later key vectors, was developed for the banking sector, in order to counter attacks on complex key management schemes. The technique associates a *“key vector”* with a key, and the key vector specifies the operations that may be performed using that key.

The banking sector uses tamper proof *“security modules”* for cryptographic processing, so that an operator has the facility to perform cryptographic operations, but does not need to know the actual keys used. Session keys may be encrypted with master keys outside the module, and hence an attacker has no means of modifying the key vectors associated with the keys.

The use of key vectors thus requires that tamper proof cryptographic processing facilities be used by the law enforcement agencies. The agency is then handed the user key, and associated key vector, encrypted under a master key stored in the module. The module will only allow cryptographic operations using the key, as specified by the key vector. Trusted software implementations of this technique may also be created.

This key vector approach has a number of advantages. The use of the keys can be further restricted so, for example, only communications between two specified parties, or messages within a given time period, can be monitored, such limitations being specified by the key vector supplied with the key, and enforced by the hardware necessary to use the key. Moreover, the validity of the keys can be checked, without simultaneously supplying their use potential use in monitoring, and all officials can be held accountable for their privileged operations.

This paper provides a brief overview of key vectors and then describes a possible extension of this technique in key recovery environments.

3. KEY TAGGING AND KEY VECTORS

Key tagging was developed for complex banking key management schemes where tamperproof cryptographic modules stored master keys and performed secure cryptographic functions. In this environment anyone with logical access privileges to the module could perform necessary security operations - checking the integrity of messages, decrypting confidential fields, checking PINs (Personal Identification Numbers), etc. but were only given sufficient privileges to perform those limited functions. Hence the privileged users never gained knowledge of the cryptographic keys used, nor could they perform functions outside their privileges limit. For example, a bank official could decrypt an amount field, or check a PIN that was encrypted in the message, but could not gain sufficient knowledge to decrypt the amount field of another message, or learn a client's PIN.

There were commonly three classes of keys: master keys, key encrypting keys and session keys. The master keys were stored in the tamperproof module such that only very highly privileged users had any knowledge of such keys, usually in a shared parts manner. Alternatively such keys were generated and stored by automated and protected techniques. Key sharing schemes were commonly employed for the entry of these keys, such schemes required a conspiracy amongst such privileged users to gain knowledge of the actual keys used.

Session keys were used for various functions in the messages, a single message would often involve three session keys, and these keys were commonly encrypted under key encrypting keys when they were transmitted from sender to receiver. This whole system involved very complex keying arrangements with a multitude of key variants, i.e. keys based upon a master key, key encrypting keys themselves encrypted with master key variants, session keys encrypted with key encrypting key variants etc.

With such complex schemes there always existed the danger that an attacker could misuse the scheme and gain knowledge of the keys used. The most likely scenario was a bank official with limited access privileges, and access to encrypted messages, who used the cryptographic module functions in some unanticipated manner.

These attacks often involved the use of key management functions at variance with the designated role of the key. To take a simple and unrealistic example. In a single-key, cryptographic protocol, suppose there are three keys:

- Master key (KM) - stored in a cryptographic module;
- Key encrypting key (KEK) - used to encrypt a session key in transmission;
- Session Key (KSAB) - used to encrypt data from Alice to Bob during a particular session.

The convention used here is:

- eK(D) denotes data D encrypted with key K;
- eK1(K2) key K2 encrypted with key K1

The system is designed such that Bob may decrypt data from Alice, encrypted with KSAB, but cannot encrypt data with this key (e.g. masquerade as Alice).

Bob's module has two functions:

- Key translate - inputs: eKM(K1), e K1(K2)

  -output: eKM(K2)
- Decrypt - inputs: eKM(K1), eK1(DATA)
  -output: DATA.

NB These functions are valid for any K1, K2 and DATA but KM is a master key securely stored in the module hardware.

Suppose that Bob is supplied with eKEK(KSAB), sent by Alice, and eKM(KEK). Using a module function key translate function, with these inputs, Bob obtains eKM(KSAB). Bob now receives an encrypted message from Alice eKSAB(MESSAGE). The module decrypt function accepts inputs eKM(KSAB) and eKSAB(MESSAGE)

delivering the MESSAGE. Bob can, however, now fool the module by inserting eKM(KEK) and eKEK(KSAB) into the decrypt function and giving the key KSAB as output, and hence obtaining the facility to encrypt a message, allegedly emanating from Alice.

Banking Key management schemes were more secure than the example above but such schemes were also much more complex and extreme care had to be taken in the design of such schemes to avoid complex attacks (Longley and Vasudevan [2]).

The technique of key tagging and key vectors addressed this problem by specifying the function of the keys in such a manner that the cryptographic modules would inhibit illicit operations. For example, in the attack described above the functions of key encrypting key, and session key, would be associated with KEK and KSAB respectively. Decrypt operations would be restricted to data encrypted with session keys. The attempt to decrypt eKEK(KSAB) would be inhibited because the cryptographic module would recognise KEK as a key encrypting key and hence not a data session key.

The key tagging approach suggested by Jones [3] used the 8 parity bits of the DES key to specify any one of 256 possible functions - key encrypting key, session data key, session MAC (Message Authentication Code) key, session PIN key etc. Hence before session data key KSAB is encrypted with, KEK, the identifying bits for session data key are set in the hitherto unused parity bits. An attacker, supplied only with eKEK(KSAB), has no means of changing these bits.

The key vector approach described by Matyas [4], and Matyas and Longley [5], used a more versatile mechanism. In this case a 64 bit key vector is described for a host of key functions. In the simplest case a 64 bit key vector KV is formed, and a key KS is added modulo 2 ($\oplus$) to the key vector before it is encrypted with a key encrypting , or master, key. Hence:

- Key vector KV indicates the permissible operations using the key KS;
- KSV = KV $\oplus$ KS is the key vectored key corresponding to KS;
- KSV is encrypted with key encrypting key KEK giving eKEK(KSV);
- Whenever eKEK(KSV) is submitted for cryptographic processing a key vector KV1 is supplied;
- the module checks that KV1 permits the proposed operation;
- the module decrypts eKEK(KSV) to give KS $\oplus$ KV
- the module adds the result of the above operation to the supplied key vector forming ((KS $\oplus$ KV) $\oplus$ KV1);
- if KV1 = KV the above result is KS;
- the operation performs the necessary operations on KS.

The attacker may attempt to cheat by supplying a key vector (KV1) which does not conform to the permitted role of KS, in order to perform illicit operations. In this case the key used in the cryptographic operations will be KS $\oplus$ KV $\oplus$ KV1 and not KS, hence the cryptographic processing will not produce the desired results.

This process of key vectors hence enables the legitimate function of the key to be sealed into the key itself. The security of the key vectoring arises because users have no knowledge of the keys themselves and must use the encrypted messages in conjunction with cryptographic modules, storing secret master keys.

This approach may be extended to key recovery systems. In this case users are asked to handover their keys and completely trust all officials with access to such keys. The use of secure modules, combined with key vector methods, can not only minimise the number of trusted officials, it can also ensure that the actions of these officials can be securely monitored.

4. USE OF KEY VECTORS IN KEY RECOVERY

4.1 Protocol

In this proposed scheme the cryptographic module may be replaced by a smart card, a special purpose chip or trusted software. The essence of the approach is that the law enforcement agency is not supplied with actual keys, but rather with cryptographic processing modules and encrypted versions of the keys. These modules will contain master keys and the security of the scheme depends upon the technical infeasibility of extracting such master keys from the module. In the final analysis the security depends upon the cost/ effort involved in extracting such keys exceeding that of breaking the cryptographic algorithms employed. In the remainder of this paper the term chip is used to cover the three possibilities - smart card, tamperproof module(hardware and/or trusted software) and chip.

It is assumed that there are three parties in the scheme: Consumer, Central Government Agency and Law Enforcement Agency. The Central Government Agency is responsible for the administration of the scheme and for the protection of the citizen's rights to privacy, in addition to the effective operation of the law enforcement agency. This organisation may, however, take other forms depending upon the legal regime of a nation. In the context of this paper there is an implicit assumption that all parties trust a few selected officials of the Central Government Agency responsible for the operation of the scheme, but may not wish to extend this level of trust to all officials in the Law Enforcement Agency. At the very least someone must be trusted to ensure that the chips are correctly designed and manufactured. The authors do not comment upon the above-mentioned assumption.

The proposed key management scheme employs a combination of symmetric (secret key) and asymmetric (public key) ciphers. For notional convenience RSA-DES is assumed but the scheme is extensible to any equivalent ciphers. The user employs the secret/public keys of the RSA cipher for key management, signing messages, authenticating himself, verifying signed messages etc.

In the key management scheme the sender generates a DES session key, encrypts it with the receiver's public key and sends it to the receiver with the encrypted message. The key management scheme may be more complex and secure, using certificates, signing the encrypted session key, including nonces or time date stamps in the key exchange etc. From the viewpoint of the scheme it is merely required that a ciphertext block eKPR(KS||other data) is available where:

- || represents the concatenate operation;
- KPR is the receiver's public RSA key;
- KS is the DES session key.

The degree of sophistication possible in the scheme depends upon the key management system used in the end user encryption devices. Key vector schemes can provide for a simple limitation of the key recovery facilities to ensure that the law enforcement agency use the escrowed keys only for decryption of messages, i.e. they cannot use the supplied keys to falsely authenticate themselves as the user, nor to forge the legitimate user's signature. With more sophisticated end user key management schemes it would be possible to restrict the range of messages that could be decrypted. The scheme may be readily implemented with a public key authentication facility (PKAF) or a public key infrastructure (PKI).

It is assumed that various recovery systems have been set up to guard components of the user's secret RSA key and a judicial decision has been made that the Central Government Agency supply this key to the Law Enforcement Agency. It may also be that multiple public keys pairs have been created by users for various purposes: signature creation and verification, session key transfer etc. The discussions below refer to the private part of such key pairs. The Central Government Agency does not supply the key in clear, but rather an encrypted version of it, protected with a key vector. This approach requires that some chip at the Central Government Agency produces a ciphertext block which is a function of:

- a master key stored in a nominated law enforcement agency chip - KMLE;
- the law enforcement agency chip identifier - LEID;
- the key vector for message decryption - KVD;

- the user secret RSA key -KSU.

The security depends upon the technical infeasibility of extracting the law enforcement agency master key, KMLE, from the chip held by that agency. This matter is discussed in more detail below. The law enforcement agency enters the received ciphertext block, together with the key vector requesting decrypt operation, the monitored user key exchange message and encrypted data. The plaintext user message is then returned by the law enforcement agency chip.

If the law enforcement agency attempts to employ the user's RSA key for any purpose other than that specified in the key vector, the law enforcement agency chip will abort the transaction. If the law enforcement agency inputs a key vector for some other operation, e.g. signing a message, together with:

- a request for that operation;
- the ciphertext block (i.e. encrypted key protected with key vector) received from the central government agency;
- a message to be signed

then the chip will perform the requested function. However the key extracted from the central government agency ciphertext block will not be the user's RSA key, but rather some variant based upon the modulo addition of two different key vectors, i.e. the attempt to misuse the user's secret key will fail.

The outline scheme can be designed for a wide variety of RSA - DES key exchange protocols. Assuming for the purposes of this paper the most simplistic protocol is employed. Bob sends Alice a session key to be used in the subsequent conversation. The data from Bob to Alice comprises:

- eKPA(KS) - DES session key (KS) encrypted with Alice's public RSA key (KPA);
- eKS(data).

The law enforcement agency has monitored this data and requested Alice's secret RSA key (KSA). This is supplied to the law enforcement agency for use on an identified law enforcement agency chip, and key vectored for data decryption. Thus the central government agency supplies:

- eKMLE (KSA XOR KVD); the KVD bits indicate DES decrypt operation and law enforcement chip ID.

The law enforcement agency enters the following inputs into its designated chip:

- DES decrypt operation;
- KV1 for DES decrypt operation;
- eKMLE (KSA XOR KVD) supplied by Central Government Agency;
- eKPA(KS) and eKS(data) gained by monitoring;

The chip performs the following operations:

- decrypt eKMLE(KSA XOR KVD) with stored KMLE giving KSA XOR KVD;
- retrieves its stored chip ID and sets the appropriate bits in KV1 hence producing key vector KVD;
- (KSA XOR KVD) XOR KVD = KSA ;
- decrypt eKPA(KS) with KSA giving KS;
- decrypt eKS(data) with KS giving data.

If the law enforcement agency attempts to:

- perform the operation on a chip with a chip ID different from that specified by the central government agency

  or

- input a key vector for an operation other than decrypt DES

then the key vector produced by the chip, from the input key vector and chip ID, will differ from KVD supplied by the Central Government Agency and the subsequent key used in RSA operations will not be that of the user's secret key (KSA).

Note that this technique uses a single master key (KMLE) in all law enforcement agency chips and the central government agency need only store one such key. Nevertheless by inclusion of the chip ID in the key vector, only one specified law enforcement agency chip may be used in the decrypt operation. In the trusted computing scenario such keys would be protected under kernelised software systems at high trust levels.

The degree of control on the law enforcement agency may be increased by more sophisticated end user key management schemes employing key vectors on the session key. For example the law enforcement agency may only be permitted to decrypt data between certain parties and Alice over a certain date period. In this case the end user key management scheme sends messages

- eKPA(KS||sender ID||date)) - DES session key (KS), sender chip ID, date, encrypted with receiver' public RSA key (KPA);
- eKS1(data) where KS1 = KS ⇦ SenderID ⇦ date

The key management messages are produced by the sender's chip and sender ID/date are automatically inserted, i.e. the sender has no control on these quantities.

The central government agency may now form a more complex key vector, which includes the sender's ID and a date range over which messages may be decrypted. The law enforcement chip now forms the key vector based upon:

- requested operation - i.e. DES decrypt;
- internally stored chip ID;
- sender ID - additional input;
- date range - additional input.

The law enforcement agency now enters the following inputs to the chip:

- DES decrypt operation;
- KV1 for DES decrypt operation;
- eKMLE (KSA XOR KVD) supplied by Central Government Agency;
- eKPA(KS||SenderID||date) and eKS1(data) gained by monitoring;
- SenderID;
- date and date range

The chip sets up the key vector according: chipID, date range and senderID, and checks that the date is within the date range, and uses this key vector to retrieve the user RSA secret key - KSA. This key is used to decrypt the key exchange message eKPA(KS||Sender ID||date) as before. KS1 is formed by modulo addition of KS with the date and SenderID *entered with the chip input*. The data is now decrypted with KS1 and the plaintext is returned.

If the law enforcement agency attempts to misuse the supplied keying information by supplying monitored data between Charlie and Alice when only monitoring of messages between Bob and Alice is permitted then it may either enter Bob's or Charlies' SenderID with the chip input data. If Charlie's ID is entered then the key vector

formed by the law enforcement chip will differ from that supplied by the Central Government Agency, if Bob's SenderID is entered then the session key KS1 formed by the chip will differ from that used in transmission.

Similarly attempts to decrypt messages monitored outside the specified date range will fail. Suppose the law enforcement agency has approval to read messages between June and August 1997 and it attempts to decrypt a message sent in May. If the correct date of the message is entered (in date and date range chip input) then the attempt will fail; either because the message date is not in the permitted date range, or if the date range is altered to include May 1997, then it will differ from the key vector date range. If the entered date is set within the permitted date range, June - August, then the decryption will fail because the date, used by the chip in the computation of the session key KS1, will not correspond to that used in the original message key construction and encryption.

A potential objection to the security of the proposed scheme lies in the use of a common master key in all law enforcement agency chips. If one chip were successfully reverse engineered then the total security of the scheme would be lost. This is, however, a matter of implementation. A conventional approach would be to compute a master key for each chip based upon a one way function of some central government agency master key and the chip ID.

The concept of limiting the use of the keys to specified chips allows for much greater auditing of law enforcement agency actions. An audit log could be produced for each chip, with hashed values stored on the law enforcement agency chip. In this case the central government agency could audit the actions of the law enforcement agency. For a particular user, the chip employed to decrypt the message will be specified by the central government agency, this body could then request from the law enforcement agency a copy of the audit logs for that chip, plus the chip itself. The hashed logs could be compared with those stored on the chip to ensure that a complete set was submitted.

4.2 Who is Monitored ?

In the scheme described above the recipient of the messages can be monitored, because the recipients secret key is used in the key management scheme. In practice it is likely that all incoming and outgoing transmissions from a given unit will be subject to monitoring, although it may be that restrictions will be placed upon the law enforcement agency, allowing monitoring only between specified parties as discussed.

The key exchange scheme can be adjusted to meet these problems. For example whenever a key exchange message eKPB(KS, Alice) is received, then Bob's unit responds with a confirmation message eKPA(KS, Bob).

4.3 Hand-over of User Keys

In the above discussion it was stated that the central government agency chip would produce the user keys, encrypted with some master key and protected by key vectors. This leaves open the question of the hand-over of user keys to the central government agency and key recovery arrangements.

The use of a central government agency chip reduces the degree of trust that must be placed in the central government agency officials, such officials never have the opportunity to gain knowledge of the private keys. They are required to provide key vectors corresponding to the terms of the legal order for the message monitoring and such actions need to be subjected to audit routines protected by the chip itself.

One possible handover scheme is described here. It assumes that two officials each with a given password, or preferably with a biometric access token, are required to invoke the production of a key for handover to the law enforcement agencies.

The user chip has an inbuilt unique transfer key KMC which is computed from a manufacturer master key KMM and chip ID. The chip performs a one off operation:

- produces two key entities KC1 and KC2; KC1 and KC2 have the property that KC1 XOR KC2 = KSU||eKPU(Chip ID), where KSU, KPU are the user's RSA secret and public key pair;

- encrypts KC1 and KC2 with KMC and outputs eKMC(KC1), eKMC(KC2).

The chip outputs are sent individually to the central government agency which checks the validity of the key components; this action does not require a high level of privilege. The central government agency chip:

- receives the key components KC1, KC2, Chip ID and check component request;
- computes KMC from the master key KMM and Chip ID
- decrypts eKMC(KC1) and eKMC(KC2) and computes KC1 XOR KC2
- obtains KSU and eKPU(Chip ID) since KC1 XOR KC2 = KSU||eKPU(Chip ID);
- decrypts eKPU(Chip ID) with KSU;
- checks Chip ID with input value and outputs 'CheckOK' or 'CheckFail';
- computes a new transfer key KMCG based upon an internally stored master key and Chip ID
- encrypts KC1 and KC2 with a new key KMCG and outputs them, then erases KC1 and KC2.

The key components eKMCG(KC1) and eKMCG(KC2) are then stored by key recovery bodies. The new transfer key is used to allay fears that a key known to the manufacturer may be employed to decrypt key components. KMCG is in fact only available in the central government chip, and backup chip.

The above procedure ensures that the key components are genuine and the keys correspond to the Chip ID. However, given the restrictions on chip usage for privileged operations, e.g. supplying keys to law enforcement agencies, this administration will not result in the illicit disclosure of keys.

The central government agency chip may also be used to audit privileged operations, e.g. handover of keys to law enforcement agencies. Such operations require the input of passwords from two privileged officials, and all actions are thereafter logged. An audit log is produced and output by the central government agency chip, and a hashed version of this log is securely stored in the chip. At any subsequent date the officials may be asked to produce the log, its completeness being checked against hashed values in the chip.

The functions of the chip, including the storage of hashed logs, may become quite onerous. However, as indicated above the term chip refers to any secure, tamperproof, hardware device and there may well be a number of such chips, each dealing with a range of chip Ids.

4.4 Security Kernel

An obvious question is ' why bother with a key vector, why not simply supply a law enforcement chip with no facilities for authentication or signing.'. The difficulty with this approach is that it requires the proof of a negative - i.e. no other facilities are available. If the proposed chip functionality is at some later date implemented into a more comprehensive device, e.g. in a well protected software system, then there is no guarantee that the signing and authentication facilities will be absent. The key vector approach effectively provides for the design concept of a security kernel imposing a mandatory access control policy, i.e. no actions are allowed that conflict with the specifications of the key vector (See Fig 4.1).

The concept of using key vectors in conjunction with a security kernel, enforcing mandatory access control on the cryptographic processor opens up the possibility of implementing the system on a trusted computer base.

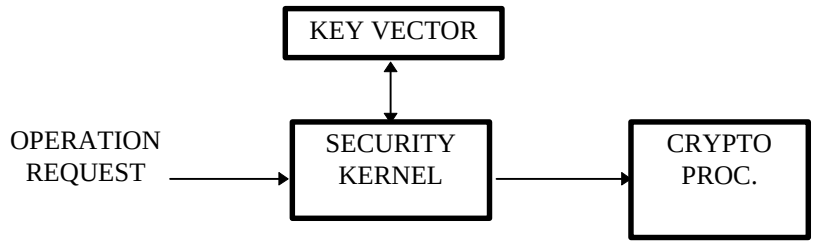

Fig. 4.1 KEY VECTOR INFORMS SECURITY KERNEL OF PERMITTED PROCESSES

5. CONCLUSIONS

The key recovery technique has been the subject of intensive debate for several years and the fundamental concern is the handing over keying information to government agencies and consequent loss of control over those keys. All key escrow schemes rely on trusting some government agency and this proposal is no different. However, with the proposed scheme is a significant reduction in the range of officials that need to be trusted, and a means of holding those few trusted officials accountable for their actions.

The proposal does not entirely eliminate the requirement for a degree of trust in the bodies responsible for designing and administering the keys. It is trusted that:

- the chip manufacturer does not reveal the manufacturer's key, if this key (KMM) is revealed then the key components supplied by the chip can be decrypted revealing the secret key;
- the chip manufacturer does not reveal the master keys (KMLE) used in the central government agency and law enforcement chips;
- the central government agency trusted officials do not conspire to produce a key vectors permitting illicit operations, e.g. key vectors for signing. Although such conspiracy may be detected by an analysis of the chip audit logs.

The system does, therefore, severely restrict the range of personnel with the capability to misuse the system, and it has the potential to hold such people accountable for their actions . To this extent it is a significant improvement on current systems where law enforcement officials may have a high degree of discretion on the use of keys handed over by key recovery bodies.

6. REFERENCES

1. Hewlett-Packard Praesidium, “International Cryptography Framework”, White Paper (Updated 18 November 1996), Hewlett-Packard WWW at http://www.dmo.hp.com/gsy/security/icf/wpaper.html

2. D. Longley and S Vasudevan, “Effect of Key Generators on the Automatic Search for Flaws in Key Management Schemes” , Computers and Security, Vol 13, No 4, 1994.

3. W. Jones, “Some Techniques for Handling Encipherment Keys” ICL Technical Journal, Vol 3, No 2, 1982.

4. S. M. Matyas, “Key Handling with Control Vectors”, IBM Systems Journal, Vol 30, No 2, 1991.

5. D Longley and S.M. Matyas, “Complementarity Attacks and Control Vectors”, IBM Systems Journal, Vol 32, No 2, 1993.