

**KEY ESCROWING SYSTEMS  
AND  
LIMITED ONE WAY FUNCTIONS**

William T. Jennings  
Southern Methodist University  
& Raytheon E-Systems,  
Greenville Div.  
Phone : (903) 457-6756  
FAX : (903) 457-7640  
E-Mail : wtjl@esygv1.com

James G. Dunham  
Southern Methodist University  
Dallas, Tx. 75275-0335  
(214) 768-3484  
(214) 768-3883  
jgd@seas.smu.edu

**ABSTRACT**

*The topic of key escrow has received considerable attention in the literature as of recent. One problem with existing public proposals for multi-agency secret splitting is that they do not address concerns that individuals within those agencies might work in collusion to gain access to large amounts of valuable keying information. This work suggests a solution to prevent large scale, random abuse of privilege. The basis of this proposal is to add a limited one way work function to make the withdrawal process much more difficult than the deposit process, thus limiting the ability to make excessive numbers of withdrawals.*

KEYWORDS: Key Escrow, Algorithm, Clipper, Limited One Way Functions

**KEY ESCROWING**

The term *Key Escrowing* has recently emerged in the literature in reference to systems which are intended to provide the capability for cryptographic key storage and retrieval. The consideration of such systems was largely ignored in the literature until controversy arose over recent government proposals concerning public cryptographic standards.

In April, 1993, the Clinton Administration made a public proposal suggesting a NSA designed encryption/decryption cryptographic device, designated the Clipper Chip, to be made available for private sector use [1]. Attacks on networked computer systems reported are growing exponentially [2]. Additionally, the introduction of digital technologies coupled with the availability of private data encryption has reportedly made the task of legal wiretapping by law enforcement agencies very

difficult to perform. The introduction of the Clipper hardware is an attempt to solve both problems. The proposal, however, has met with a great deal of criticism [3, 4]. Included in the proposal was a requirement that the master keys for these devices be registered with the government. The registration has been referred to as Mandatory Key Escrowing (MKE).

As a consequence of the legal implications of MKE, it has been suggested that separate agencies or agents would be given separate components of the key. This would be accomplished using secret splitting techniques [5]. Only when the components are put back together may the key be successfully recovered and used. Each of these agencies would serve in the role of a trusted authority or "escrow agent" [1].

A *Key Escrowing* system is fundamentally different from cryptographic systems based on zero-knowledge techniques such as some password or authentication systems. A Key Escrowing system must provide a withdrawal capability rather than simply verification. The value of the keys stored is equal to the sum of all the contained keys. Thus for a national key escrow system, the economic value of stored keys would be immense. There are interesting technical questions that are raised. Barlow [1] has raised the issue of whether or not key depositories would themselves become the target of criminal or terrorist organizations, raising the uneasy question of what happens if the key depositories themselves are compromised. The key depositories, containing an enormous wealth of information, would serve as a high priority target for terrorists or hackers. Someone with access to keying information stored in large key escrow databases would be confronted with an tremendous temptation to browse through all therein. Current proposals would require collusion between escrow agencies or individuals within those agencies in order for abuse to occur. It is possible to envision circumstances whereby this might happen.

One response is to simply argue that key escrowing should not occur. It is however entirely likely that it will occur anyway. There are of course secret sharing schemes that are more elaborate and might require larger groups or more agencies to act in collusion to reveal the secrets. Creating more agencies and splitting the information up into more pieces does seem to provide a higher level of security. While it requires more individuals to act in a collusive manner, it does not change the basic nature of the problem's weakness.

It is apparent that, for such a system, what may be desired is a self-limiting or self-regulating algorithm to preclude the potential for wholesale abuse. Indeed this same argument has been offered by Bellare and Goldwasser [6]. Since keys are very frequently created (deposited), but rarely withdrawn, there is an inherent asymmetry to the problem that can be used to advantage. If it were as easy to withdraw keys as it is to deposit them,

then there is every possibility that keys may be withdrawn at an excessive rate. Therefore it would be preferable to design a system that inherently limits the rate of withdrawals to a pre-defined maximum rate. This provides a deterrent against a specific threat profile, that of the "casual key browser," or systematic abuser.

This application thus suggests that there should be a specific cost associated with each key withdrawal from a Key Escrowing depository. This then means that one may not simply randomly browse among what is in principle a very large data base of keys but must in general request specific keys to be withdrawn. What is proposed is a function designed in such a manner that the computational cost of key withdrawal greatly exceeds the cost of deposit in a controllable manner. Such a proposal may not in itself prohibit an authorized individual from asking for additional keys as well, but the numbers would be inherently self limiting. Additionally any continuing pattern of such behavior would be statistically detectable, since the cost (on average) for systematically requesting additional keying information would be detectable.

This subject is of wider interest than that of simply addressing issues concerning the government sponsored Clipper chip. There are indeed important commercial applications for Key Escrowing systems as well. Valuable corporate information that should be protected by strong cryptographic methods often is not. Recall that data thus encrypted is totally unrecoverable without the keys. Therefore it is highly desirable that commercial systems provide key escrowing capabilities to facilitate data recovery in the event that keys are lost.

### **KEY ESCROWING SYSTEM IMPLEMENTATION**

Traditional electronic approaches to the storage of vital key information normally involve keeping copies of keys in a trusted database, protected by one or more master keys. This master key is therefore more valuable than the other keys and is at least as important as the sum of all of the keys that it protects. Consequently, anyone in possession of a master key would be afforded complete and unlimited access to all of the information protected by all of the sibling keys stored using the master key. The advantage of having a master key is that there is only one key of which to keep track. The chief disadvantage is that a master key constitutes a single point defense. Compromise of a such a master key is therefore very critical. A single key database can be compromised by anyone in possession of the master key material.

It therefore would seem to be desirable to segregate data into multiple master key domains. This of course has the undesirable property of multiplying the number of master keys which must be safeguarded or protected in of themselves. Ultimately these keys

are protected in much the same manner as a single master. It also perhaps suggests that hierarchical approaches suffer similar maladies and do not really address the underlying problem.

There have been suggestions that solve this problem by using secret splitting techniques to provide complementary components for each key stored. These components would have to be put together to recover the original key. Components would be separated at time of creation and stored with alternate "trusted" agencies. These techniques not only offer protection from external attack on the database but also some protection from abuse from within a particular trusted agency. These ideas are inadequate in that they do not address concerns over the possibility of collusion between individuals within the agencies with access to the databases. It is recommended that additional measures are necessary to discourage abuse of the system and to provide additional opportunities for oversight.

### KEY ESCROW AND LIMITED ONE-WAY FUNCTIONS

Key Escrowing systems can be characterized as being strongly one way in their basic input/output bandwidth requirements. Many keys are created, but few are ever retrieved. Typically the input bandwidth far exceeds the aggregate output bandwidth, perhaps by many orders of magnitude. A balanced design for such a system might suggest that the algorithm for storage and retrieval match the actual bandwidth requirements. It would be advantageous to implement an algorithm that requires far less work to make a deposit than it does to make a withdrawal. It is proposed herein to refer to applicable functions that display asymmetric work requirements as being *limited one-way functions*. This proposed methodology is to use limited one-way functions to effectively limit the rate of withdrawals. We draw the distinction from normal one-way functions and hence use the term *limited* because we want to look at candidate functions which are not necessarily strongly one-way.

Candidates for useful functions should be provably asymmetric. Ideally there should be provable bounds on the ratio of the amount of work required to go forward versus the work required to go back. This is important because the effectiveness of the ability of the function to impose cost on the user is characterized by the upper and lower bounds on this ratio. Another aspect of this methodology deviates significantly from a classical cryptographic application. Since the key escrow database server has access to the plaintext key information by possessing the master key, but is simply being penalized by a work function for key withdrawal, the algorithm may legitimately only require that each transaction be accomplished taking a prescribed length of time, on average. This constitutes a significant shift of paradigm. The concept of the penalty is to limit or regulate the general flow of data out of the key escrow database. Hence, to satisfy the demands of this requirement, it

may only be necessary to determine the average or statistical complexities of the limited one-way function and it's inverse.

One possible candidate would simply be to use a suitable cryptographic technique with a limited key size. This is the most straight-forward approach. Conceptually it is very similar to partial key escrow techniques such as proposed by Shamir [7]. The difference in this case being that the entire key may be escrowed but the work may be imposed prior to accomplishing key withdrawal rather than after withdrawal from the escrow. The decryption (withdrawal) is accomplished either by brute force techniques or by directly breaking the key. Since suitable cryptographic techniques to accomplish this are based on solving NP-complete problems, there are not provable tight lower bounds on the work required to accomplish this. Additionally there may be a large differential in work required between the normal withdrawal technique (if implemented by brute force) and the backdoor path (breaking the key). Therefore there are not necessarily very tight controls on the work required to accomplish this.

We propose another example of how a suitable Limited One Way function might be implemented. The following is an extension to an algorithm originally proposed by Merkle [8]. Let us consider a case where we shall define (following Merkle's original terminology) the puzzle transmitted by Alice to Bob to be as follows: Alice generates, using the encryption keys, matched cryptogram/decryption key pairs  $(C_i, Kp_i)$ ,  $i = 1, 2, \dots, N$  corresponding to a set of messages  $\{M_i\}$   $i = 1, 2, \dots, N$  where  $i$  is simply an index used to identify which member of the set of pairs is referenced. The message  $M_i$  contains a corresponding token  $T_i$ . In this example the familiar RSA system is used to illustrate the concept. This is not however a general requirement. Thus Alice generates the puzzle set:

$$P = \{(C_0, Kp_0), (C_1, Kp_1), \dots, (C_i, Kp_i), \dots, (C_N, Kp_N)\} \quad (1)$$

where  $C_i$  is the  $i$ th cryptogram corresponding to the  $i$ th message  $M_i$ , and where  $Kp_i$  is the  $i$ th public key generated by Alice.  $Kp_i$  is used to encrypt  $M_i$  and corresponds to  $Ks_i$  which is the secret key retained by Alice. It is assumed that they share the commonly agreed upon encryption function. Alice communicates the set  $P$  to Bob.

Bob selects  $j$  at random, where  $j \in 1, \dots, N$ , then chooses the  $j$ th ordered pair,  $(C_j, Kp_j)$ , from the set  $P$ . Bob derives  $T_j$  from  $C_j$  by performing the decryption:

$$D(Kp_j, C_j) = M_j; T_j \subset M_j, \quad (2)$$

where  $D$  is the decryption function.

Bob then forms the message  $\mu_j = (T_j \ \&\& \ R)$ , the concatenation of the selected token and a randomly chosen vector  $R$ . Bob proceeds to form the response message  $S$ , such that:

$$S = E(Kp_j, \mu_j), \quad (3)$$

and sends  $S$  to Alice. Alice may then recover  $T_j$  by application of the secret key  $Ks_j$ . Alice does not know which of the  $N$  keys to use and thus must try keys randomly from the set of  $N$  until a match is made.

It is assumed that the channel between Alice and Bob cannot be tampered with but is not secure. An observer, Carol, may see both the initial message  $P$  and the response  $S$  from Bob. Carol therefore has all of the  $N$  public keys but does not have the corresponding secret keys. To "discover" the message Carol is faced with the problem of first deriving the  $N$  tokens, then forming  $N \cdot R$  messages of the form  $(T_j \ \&\& \ R)$ . Finally Carol must then encrypt these and compare the result to  $S$  in order to discover Bob's choice for  $j$ .

We should consider the amount of work imposed by this algorithm upon the various parties involved. The work that Carol is forced to perform is now greater however than that performed by either Bob or Alice. Carol does not have the benefit of having the decryption keys that are available only to Alice. Carol must try all  $\text{Avg}(N \cdot R)$  possibilities to discover the decision that was derived where we use the notation  $\text{Avg}(\ )$  to refer to the average complexity. Refer to this approach for obtaining the solution as the "front-door" approach.

Carol is at a disadvantage to Alice by a factor of  $\text{Avg}(R)$ , the amount of randomization information embedded in the problem. This is because Carol does not possess the decryption keys which are the sole property of Alice and are not revealed in the process. Carol is forced to try all  $\text{Avg}(N \cdot R)$  combinations until a match is found. Carol does, however, have an alternative possible attack. Carol may attempt to break  $\text{Avg}(N)$  decryption/encryption key problems, directly attempting to discover the secret keys. This approach to solving the problem is referred to as being the traditional "back-door" approach to solving the problem. The work associated with this approach thus represents an upper limit on the amount of work that Carol must perform. System parameters thus can be chosen such that Carol is forced to go in through the built in front door, because that is the only computationally viable path. Let the amount of work performed to directly break the key problem by brute force methods (the back-door approach) be represented by  $\text{Avg}(W_B)$ . Let the amount of work that Alice performs using trapdoor information to accomplish a decryption be represented by  $\text{Avg}(W_T)$ . We shall presume that for reasonable choices of system parameters that  $\text{Avg}(W_T) \ll \text{Avg}(W_B)$ . We can also reasonably presume that  $\text{Avg}(W_T) \approx W_E$  if the encryption and decryption processes are

symmetric. This assumption is true for instance of some public key cryptosystems such as RSA. The work that is now required by each party involved is given by:

$$W_{bob} = W_D = W_E, \quad (4)$$

$$W_{alice} = N * W_E = \text{Avg}(N * W_T), \quad (5)$$

$$W_{carol} = \text{Avg}(N) * W_D = \text{MIN}( \text{Avg}(N * R * E), \text{Avg}(N * W_B) ). \quad (6)$$

The work required by Bob to efficiently perform this calculation (assuming RSA) can be estimated to be  $Kn^2 \log n \log \log n$ , where  $K$  is a system dependent constant [9]. It was recently reported that the fastest single chip implementation for performing modular exponentiation is capable of evaluating 560 bit operations per 5.5 msec [10]. Consider an example system using this chip, using 560 bit numbers and taking  $N$  to be  $10^3$  and taking  $R$  to be  $10^5$ .

$$W_{bob} = W_D = 5.5 \text{ msec} \quad (8)$$

$$W_{alice} (\text{avg}) = N * W_E / 2 = 2.75 \text{ sec} \quad (9)$$

$$W_{carol} = N * R * W_E / 2 = 2.75 \times 10^5 \text{ sec} \approx 3.2 \text{ days}. \quad (10)$$

By using this method it is possible to control the amount of work that Carol must perform to solve the puzzle. In the example above, withdrawals could only occur in this system at the maximum rate of about 114 per year. This is reasonable assuming about 10 regional depositories. The number of court-ordered wiretaps for all federal, state, and local law enforcement purposes is approximately 1000 per year. Specifically there were 919 wiretaps authorized in 1992 and 976 in 1993 [5]. Carol is forced to perform a very large number of simple operations (on average) to resolve the answer. Because of this it is possible, by adding enough randomization, to take advantage of average computational complexity in determining the required work. This has a distinct advantage over implementing a single weak cryptofunction such as with a limited key size. The desired performance of the proposed algorithm can be controlled by adjusting the statistical parameters. This offers a greater degree of control over the results than that offered by the simpler approach.

To apply this algorithm to the problem of Key Escrow, we can consider a record made of the exchange between Alice and Bob (such as would be seen by Carol) as the material to be deposited in the escrow. In this scheme, Bob and Alice negotiate for a key exchange with Alice as the key requester and Bob as the key generator. Carol represents the recording/withdrawal mechanism. Prior to storage, the transaction is encrypted using a strong cryptographic technique and master keys used to protect the overall database. It is also practical to incorporate secret splitting mechanisms as well. Depending on the application Alice may either keep her secret puzzling keys or they may simply be discarded as part of the process. This escrowing process is illustrated in Figure 1.

Withdrawal of the keying material would involve retrieval of the transactions that had occurred between Bob and Alice first using the database master key for decryption to recreate the transaction. This transaction would then have to be "broken" in the manner that Carol would need to accomplish in order to discover Bob and Alice's agreement. Thus this second stage of decryption represents the controllable work function used to limit the rate of key withdrawal. This withdrawal process is illustrated in Figure 2.

### SUMMARY

Key Escrowing Systems have unique characteristics which distinguish them from other cryptographic systems. To address some of the unique requirements of these systems, the concept of limited one-way functions was introduced. This proposed technique is not intended to replace master keys or secret splitting techniques intended to preserve integrity of the data from external attack. Instead this additional layer of protection is intended to limit the ability of otherwise properly authorized individuals to withdraw keys at an excessive rate. An example algorithm was also introduced. It was suggested that this concept is a new tool available to deal with situations where the rate information retrieval is desired to be controlled or where some minimum time limit is to be asserted within a defined probability. This technique may be applicable to problems other than Key Escrow as well.

An additional layer of functionality is provided by the proposed algorithm in the form of a specific work function and hence economic costs associated with the function of key recovery from a key database. Master keys can still be used to provide the front door into the main database. Additionally this technique does not preclude the possibility of also using secret spitting techniques as well. The added value is that once inside the door, there is no free access to any and all information contained therein. The data contained can only be obtained by an authorized individual who in addition can afford to pay the price of retrieval. The specific benefit of this approach is that an inherently limiting or regulatory process is imposed. This means that general abuses can be limited, thus solving a fundamental problem that is not addressed by conventional cryptographic methodologies.

In a Key Escrowing system, exemplified by that proposed for a national communications system, inclusion of a methodology for limiting withdrawal bandwidth would provide the ability to prevent collusive parties from freely shopping among the keying information without imposed constraints. The system could be designed so that the withdrawal rate was adjusted such that only a reasonable number of withdrawals over a period of time would be possible. What is provided however is a greater level of privacy protection to the general populace from the possibility of widespread random monitoring of otherwise private transactions.

## REFERENCES

- [1] J. P. Barlow, "A Plain Text on Crypto Policy," *Communications of the ACM*, Vol. 36, No. 11, Nov. 1993, pp. 21-26.
- [2] S. Staniford-Chen, L. T. Heberlein, " Holding Intruders Accountable on the Internet," *Proceedings of the 1995 IEEE Symposium on Security and Privacy*. May, 1995, pg. 39-49.
- [3] R. W. Holleyman, "On the Export of Software with Encryption Capabilities", testimony presented at Key Escrow Meeting, NIST, Gaithersburg, Maryland, Sept. 6 1995.
- [4] U. S. Senator Patrick Leahy, "Statement of Patrick Leahy on Vice President Gore's Clipper Chip Letter," public letter, 21 July 1994.
- [5] Froomkin, Michael A., "The Metaphor is the Key: Cryptography, the Clipper Chip, and the Constitution," *U. Penn. Law Rev.* 709, 1995.
- [6] Denning, D. E., "Descriptions of Key Escrow Systems" unpublished, <http://www.cosc.georgetown.edu/~denning/crypto/Appendix.html> reference to technique attributed to Bellare and Goldwasser.
- [7] Denning, D. E., "Descriptions of Key Escrow Systems" unpublished, <http://www.cosc.georgetown.edu/~denning/crypto/Appendix.html> reference to technique attributed to A. Shamir.
- [8] Merkle, R. C., "Secure Communications Over an Insecure Channel," *IEEE Trans. on Information Theory*, 1976, IT-22, pp. 644-654.
- [9] Borodin, A., "Computational Complexity: Theory and Practice" from *Currents in the Theory of Computing*, Aho, A.V. editor, Prentice Hall, 1973.
- [10] Orup, H., "Simplifying Quotient Determination in High Radix Modular Multiplication," *Proceedings of the 1995 IEEE 12th Symposium on Computer Arithmetic*, July 1995, pp. 193-199.

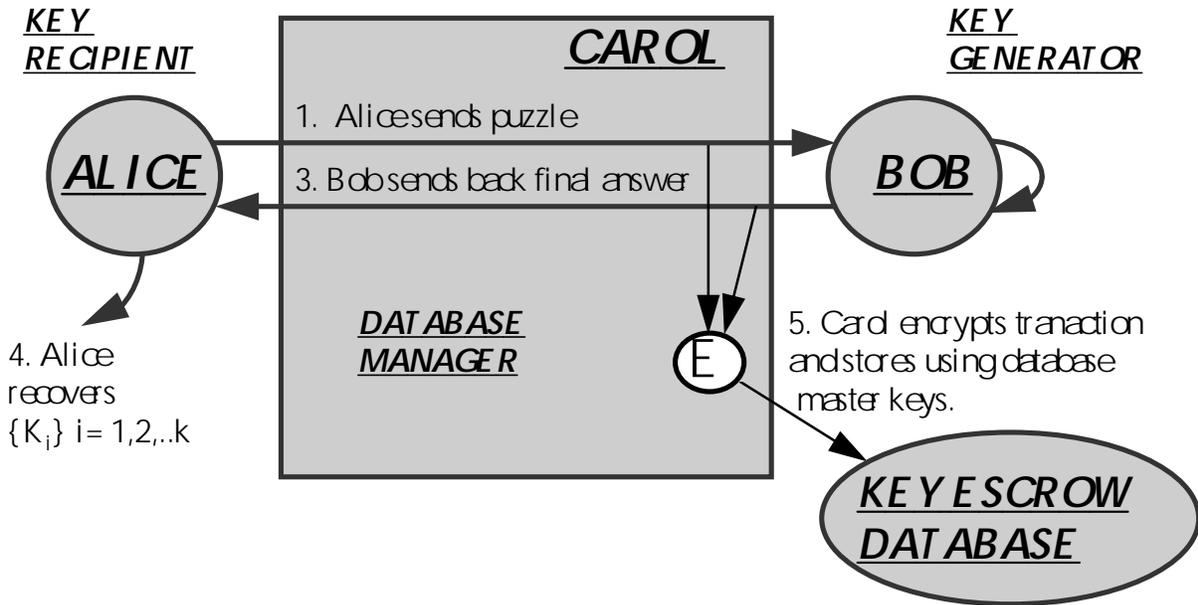


Figure 1 - Key Escrow Process

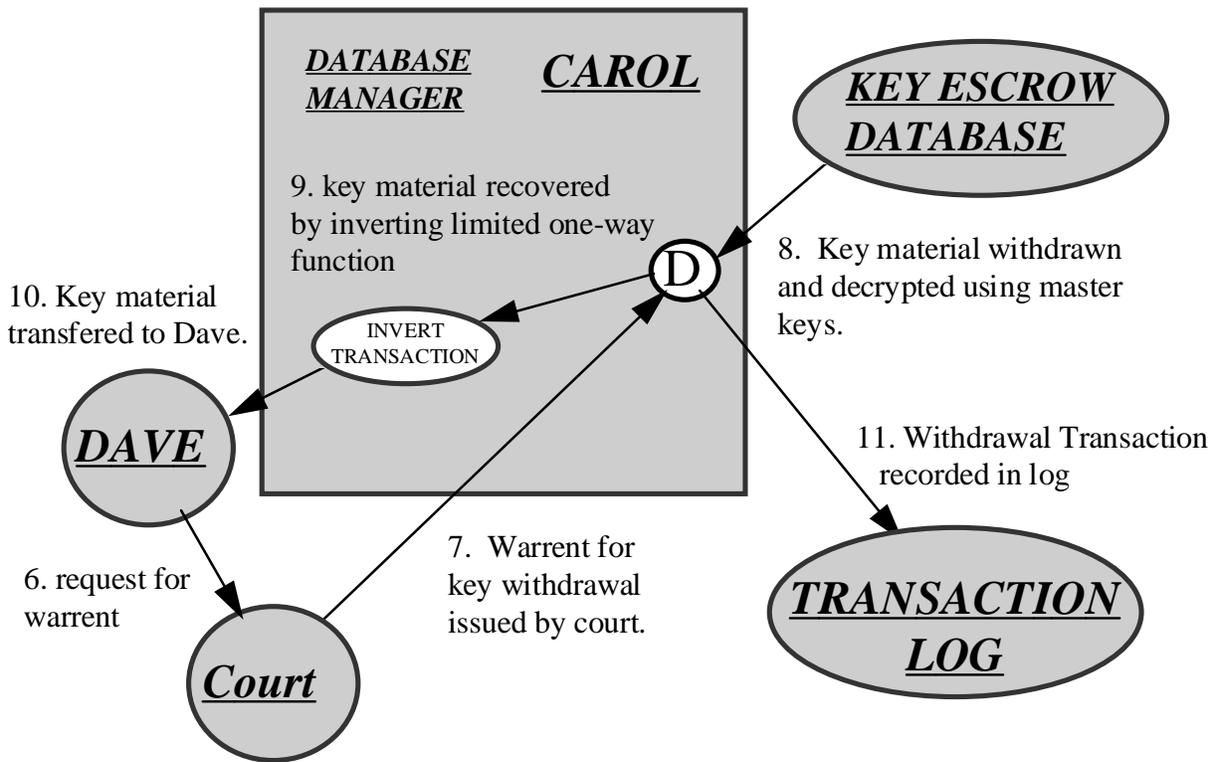


Figure 2 - Key Withdrawal Process

