HOW TO KEEP A SECRET ALIVE
EXTENSIBLE PARTIAL KEY, KEY SAFEGUARDING, AND THRESHOLD SYSTEMS

David Chaum

Center for Mathematics and Computer Science (CWI)
Kruislaan 413
1098 SJ Amsterdam, The Netherlands

PREVIOUS WORK

Partial key, key safeguarding, and threshold techniques appear to be another example of similar good ideas springing up in several places at nearly the same time—each with a different name and associated terminology. The use of partial key techniques actually appeared in print first in a technical report [Chaum 79] before the key safeguarding techniques were presented at a conference [Blakley 79], and before the threshold schemes were submitted for publication [Shamir 79]. (In fact the author received comments on the technical report from Shamir, along with a draft of the threshold scheme.)

The essential idea of all three techniques is that someone who knows a secret number can form other numbers from it, such that it is easy to compute the secret number if you know any fixed k of the other numbers, but knowing less than k of the other numbers gives no clue about the secret number.

Feistel proposed dividing a key into parts such that the key could be recovered by forming the bitwise exclusive-or of all the parts [Feistel 70]. Shamir quotes a problem from a combinatorics text in which all fixed-size subsets of a set of scientists have sufficient physical keys to unlock a file cabinet protected by multiple padlocks [Liu 68]. The partial key technique, which works with cryptographic keys, was an improvement over the approach of the padlocks, because it allowed each trustee of keys to hold only one key, instead of many keys. It might be quite convenient in practice for small numbers of trustees, and can provide unconditional security. It appeared as a single paragraph and footnote in a proposal for distributed computer systems that can be trusted by groups who don't necessarily trust each other. The secret sharing and threshold techniques are far more elegant than the original partial key technique. They allow large systems in practice, and also can provide unconditional security. A number of similar

G.R. Blakley and D. Chaum (Eds.): Advances in Cryptology - CRYPTO '84, LNCS 196, pp. 481-485, 1985.

schemes have appeared subsequently.

Since no standard terminology seems to have emerged, the following will be used: in a partial key system, the *system creator* divides the *key* into *partial keys* (Blakley's shadows, Shamir's pieces) that are transmitted to various *trustees* (Blakley's guards), such that any *quorum* (Shamir's threshold) of trustees is sufficient to *recover* the key, and less than a quorum of trustees is insufficient.

The present work describes techniques allowing any partial key system to adapt for survival in the face of changing availability of trustees and even changing needs for system parameters.

MOTIVATION

All trustees may not always remain able and willing to recover the key in a partial key system. For example, a computer acting as a trustee and storing a partial key may be destroyed by natural or other disaster, or a human trustee may be hit by a truck. Less sever causes can also easily be imagined, such as hardware or software failures, or a person suffering from loss of memory. A trustee that will never participate in recovering a key will be said to be *lost;* a trustee able and willing to participate will be said to be *present.*

Clearly it is prudent to consider scenarios in which trustees become lost. If less than a quorum of trustees remain after one or more trustees is lost, then it has become impossible to recover the key. Problems could also result from a substantial loss that leaves only a quorum of trustees, or relatively few more than a quorum, for reasons similar to those for having the greater number of trustees in the original system. For example, a loss of sufficient trustees to prevent recovery might then become too likely or too easy to cause, or some cooperating trustees might gain significant power from their ability to prevent recovery of the key.

A solution to the problems of loss of trustees is presented in the next section. The section after that considers causes of unavailability of trustees other than simple loss.

REPLACING A TRUSTEE

The problem of loss of trustees is solved by allowing new trustees to *replace* lost trustees. If some trustee is lost, and a quorum of trustees is present, then the quorum can give a replacement trustee the same ability that the lost trustee had to participate in the partial key system; the replacing trustee is given the slot of the lost trustee replaced.

Any partial key technique can be used in a way that allows such replacement. The ability to allow replacement can be "built-in" when the partial key system is first created, or it can be "added-on" later separately by each trustee. The added-on approach is considered first for a single trustee and next for all trustees. Then the built-in approach is considered.

Suppose you as a trustee wish to make provisions that would allow your own replacement,

should it ever become necessary. The essential idea of the solution is that you create your own partial key system to allow the other trustees to recover your partial key. You divide your partial key into *sub*-partial keys. The quorum parameter you use is the same as that in the original partial key system, and the number of partials you generate is one less than the number of original trustees. Then you transmit a different sub-partial to each other trustee. (You might use cryptographic techniques, e.g., to provide secrecy and authentication of sub-partials transmitted.) If you should become lost, heaven forbid, and some present quorum wants to replace you by filling your slot with a replacing trustee, then each member of the quorum would separately transmit to the replacing trustee the sub-partial they received from you. (Again these transmissions might be cryptographically protected.) When the replacement gets a quorum of your sub-partials, the replacement is able to recover your partial—and has become able to participate in the original system in your place.

If other trustees try to provide for their own replaceability in a similar way, problems may eventually arise since replacements won't have access to sub-partials, and consequently won't be able to help make subsequent replacements. Consider a solution in the homogeneous case of a set of trustees who each provide every other trustee with sub-partials. After a trustee receives sub-partials from all the other trustees, the trustee encrypts them together using the trustee's partial key as a key in say a conventional cryptosystem. This collection of all the sub-partials received by a trustee, encrypted by the trustee's partial key, will be called an *extender.* Thus, once a replacement gets sufficient sub-partials to allow recovery of the partial, the replacement can use the partial to decrypt the extender and obtain the full collection of sub-partials that was available to the replaced trustee.

Unlike the key and partials, extenders are public. But to be of use they must of course be accessible. One way to treat extenders is to allow them to be copied freely, and assume that this provides adequate protection against all copies becoming inaccessible. Another way is for trustees to keep copies of extenders.

The built-in approach promised above is easily seen by noticing that the original system creator knows all the partials and can thus form sub-partials, and from them the extenders. The system creator might, for example, transmit the sub-partials to the trustees along with the partials, and make the extenders public.

ADDING A NEW TRUSTEE

It might be desired to add new trustees without replacing any specific trustee. One reason might be just to expand the reliability of the system. Another reason is to be able to contend with *missing* trustees, i.e. those that are not present and not known to be lost. For example, a trustee might not be reachable because of a communication failure or other circumstances, or whether a trustee will recover from some disabled state may not be known with certainty. A kind of reversible replacement, even if such were possible, may not be the best approach. Consider for example the case where a missing trustee returns just making a quorum, but

recovery is impossible since the replacing and returned trustees together can contribute only one partial. Thus it may be desirable to compensate for missing trustees by *adding* new trustees.

Now a technique for allowing new trustees to be added is presented. It is a built-in approach, requiring the system creator to make provisions for the additions when the system is created. The total number of trustees that can be added must be fixed before the system is created (but see the next two sections).

The technique is essentially the same as the built-in approach of the previous section, except that some partials are created that are not issued to trustees initially, and extenders need only cover these partials. The system creator provides each initial trustee with the usual partial and also with a different sub-partial for each trustee slot that can be added. A quorum of the sub-partials for a particular such slot allows recovery of the partial for that slot. When the sub-partials are transmitted to a new trustee, the new trustee uses them to recover the appropriate partial. This partial allows the new trustee to participate in recovering the original key—just as any other trustee. Systems providing for the addition of more than one new trustee could use extenders to ensure that new trustees are themselves as capable as the other trustees of allowing new trustees to be added. Thus, it would be sufficient for the system creator to form an extender for each new trustee slot, such that each extender contains sub-partials for all the other new trustee slots. If the order in which new slots will be filled is fixed when the system is created, then the extender for a particular new slot need only contain sub-partials for the new slots that appear after it in the order. (A similar technique is buried in [Chaum 82].)

REPLACING A SYSTEM

It might be desired to replace a partial key system that is in use by a new partial key system—without using a mutually trusted party like the original system creator. There are several reasons for wanting to do this: to change the parameters of the existing system, such as the quorum size and the number of trustees; to change the set of trustees; or to restore the built-in ability to add trustees if the original provisions become depleted. Of course the extent of effective change may be limited because some trustees may not be relied on to destroy their old partial keys. If the number of trustees not destroying old partials is less than the old quorum, then the situation is effectively the same as if all had destroyed their partials. A quorum of trustees acting together will be able to *replace* an existing system with a new system.

The essential idea is the same as with replacing or adding a trustee, except that the parameters and trustees of the sub-partials may differ from those of the replaced system. What is in effect created is an old quorum of partial key systems, each of whose parameters and trustees are the same, and each of whose keys is a partial of the replaced system. Thus, a new partial is actually a *collection* of partials, one from each of the old quorum who established the new system. A new quorum of such collections is sufficient to recover an old quorum of old partials, and thus the original key. Extenders are used only to provide for adding trustees in the new system, with each trustee issuing them as the system creator did in the previous section. Of course it should

be ensured that old partials will not be destroyed until the new system is in place.

If a newer system is to replace a new system, then each collection of the new system would be encrypted under a key created by the trustee holding the collection, and then made public or otherwise protected like extenders. The key created by each trustee would be used as the key in a partial key system, with the newer parameters, whose partials would be transmitted to the newer trustees. (A similar technique is also buried in [Chaum 82].) For arbitrary fixed maximum system size and parameters, the time and space requirements of a series of systems succeeding each other in this way are linear in the length of the series.

OPEN QUESTIONS

More natural and efficient mechanisms for extensibility seem possible.

CONCLUSIONS

Various ways to allow the trustees of a key to adjust to changes in their own membership and external circumstances have been described. The techniques appear to be quite flexible and potentially quite useful in practice.

REFERENCES

(1) Blakley, G.R., "Safeguarding Cryptographic Keys," Proceedings A.F.I.P.S. 1979 National Computer Conference, vol. 48, June 1979, pp. 313-317.

(2) Chaum, D., "Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups," Memorandum No. UCB/ERL M79/10, University of California, Berkeley, CA, February 22, 1979.

(3) Chaum, D., "Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups," dissertation, Computer Science, University of California, Berkeley, CA, June 1982.

(4) Feistel, H., "Cryptographic Coding for Data-Bank Privacy," Research Report RC 2827, IBM T. J. Watson Research Center, Yorktown Heights, NY, March 1970.

(5) Liu, C.L., Introduction to Combinatorial Mathematics, McGraw Hill, NY, 1968.

(6) Shamir, A., "How to Share a Secret," Communications of the ACM, vol. 22, no. 11, November 1979, pp. 612-613.