

Computer and Network Security

©Copyright 2000 R. E. Newman

Computer & Information Sciences & Engineering
University Of Florida
Gainesville, Florida 32611-6120
nemo@cise.ufl.edu

Key Escrow and Secret Sharing (Pfleeger Ch. 3, KPS Ch. 7)

Safety Net

1 Need for Key Escrow

1.1 Government eavesdropping/forensic evidence

1. Law enforcement
2. State department

1.2 Removal of single point of failure

Individual keys used to encrypt critical organizational data - if user won't or can't produce key, the organization must still be able to access the data

2 Requirements

2.1 Access

Escrowed keys must not become available to nonauthorized parties

2.2 Protection against abuse

Keys must be escrowed in such a way that it is hard to abuse them by authorized parties

2.3 Limitations on use

It should be possible to limit the information revealed when an escrowed key is accessed

2.4 Information hiding

Even revelation of a part of escrowed key information should not provide useful information about the key

3 Approaches

3.1 Parts

The key K may be broken into parts, K_1, K_2, \dots, K_n , where $K = K_1|K_2|\dots|K_n$. All n pieces are needed to reconstruct the key, but if any one part is missing, the remainder cannot be reconstructed from the other parts.

However, this in effect reduces the keyspace for brute force search. Assuming all the pieces are the same length, if $n = 2$, then either secret holder can try to guess the remaining key bits. With a key of length of say 90 bits, only 45 bits must be guessed, which is feasible. If $n = 10$, then each piece is 9 bits long, and if 8 of the 10 collude, then only 18 bits are missing - trivial to attack by brute force.

In order for this scheme to work for key escrow, each piece must be long enough to make it infeasible to attempt brute force methods to discover the missing key bits.

3.2 Secret Sharing

Key escrow is a special case of secret sharing. These schemes are not supposed to reveal anything about the secret unless the threshold requirements are met.

4 Methods

4.1 XOR

- The original key K is XORed with a random number R to produce

$$L = K \oplus R.$$

- L and R are stored securely in different places under the control of different entities.
- Both entities must cooperate in order to reveal both L and R .
- Once L and R are known, then $K = L \oplus R$ may be found easily.

In essence, this is the approach taken by the US government for key escrow in Clipper.

This can be generalized to multiple secret sharers, any one of which can know nothing about the secret, but any two of which can recover the secret easily.

- Instead of one random number R , $m - 1$ random numbers R_1, R_2, \dots, R_{m-1} are generated.
- The first $m - 1$ portions are the random numbers,

$$P_i = R_i, \quad i = 1, \dots, m - 1.$$

- Now the secret K is XORed with all $m - 1$ random numbers, and an m th portion is created,

$$P_m = R_1 \oplus R_2 \oplus \dots \oplus R_{m-1} \oplus K.$$

- Any $m - 1$ colluders can only find either the XOR of the random numbers (and nothing about the key) or the XOR of the key with one of the random numbers.
- With all m pieces, the whole number may be reconstructed.
- This approach reveals the length of the secret, but nothing else, unless all the pieces are combined.

4.2 Threshold Schemes

An (n, m) -threshold scheme is a method by which n shares are generated and any m may be used to reconstruct the secret.

4.3 Blakely's Vector Scheme

Use the intersection of hyperplanes.

4.4 IDA

Rivest published a method for information dispersal that is a generalization of parity (used with the XOR scheme above).

The idea is to treat the secret as an $m \times b$ matrix (padding as necessary), with b depending on the length of the secret S

$$(b \geq m, b \geq |S|/m).$$

Another matrix, A , which is an $n \times m$ matrix with $n = m + r$ is formed with the special property that any $m \times m$ submatrix of A is invertible (using the elements of S as elements of a commutative ring, e.g., $\text{GF}(2^8)$).

Let

$$T = AS$$

(using matrix multiplication over the ring), noting that each row of T corresponds exactly to a row of A times S , so each row of T has "holographic" information from S in it.

Thus any $m \times b$ submatrix T' of T is generated by the corresponding $m \times m$ submatrix A' of A ,

$$T' = A'S$$

and that A' is invertible.

Hence,

$$S = A'^{-1}T' = A'^{-1}AS = S.$$

Now the IDA may be used as an erasure corrector (used for example when the rows of T are distributed over a RAID device and one or more of the component disks fails): simply find at least m good rows and use these to regenerate S (assuming that A is known, and the identity of the rows is known so that the submatrix of A can be found and inverted, and the submatrix of T can be formed properly).

It may also be used to share a secret, requiring that m of the $m + r$ partial secret holders agree to cooperate to reconstruct the secret.

4.5 Shamir

This approach uses polynomials. Any n -degree polynomial may be represented in a number of ways. The most familiar is the $n + 1$ coefficients, but it may also be represented by any $n + 1$ points $(x, P(x))$ on the curve. (Recall affine equations represented by slope and intercept, or by any two points.) In order to share a secret S , a polynomial is constructed such that $S = P(0)$ (or any other designated value in the domain). S is treated as a large unsigned integer here. The secret is shared by having at least $n + 1$ other integral points $(i, P(i))$ that are distributed to the partial secret holders. At least $n + 1$ of these must provide their information in order to correctly reconstruct P , and find $P(0)$.

The numbers can get very large doing this, so the whole operation can be performed over the ring of integers modulo p for some large prime p (greater than the secret S).

5 Issues in Secret Sharing

5.1 Cheaters

1. One gives a bogus share (prevents the secret from disclosure)
2. One waits to see the other's secrets then steals those shares
3. One sees at least m secrets, then quickly constructs a valid share

5.2 No central secret generator

Can we generate and share a secret without anyone knowing it until enough combine their pieces?

5.3 Sharing a secret without revealing the shares

5.4 Verifiable secret sharing

1. How can we verify that our shares are good without reconstructing the secret?
2. How can we be sure another has presented a valid share?

5.5 Secret sharing with prevention

Can there be a negative threshold $x \leq n = m$ such that any x shareholders can prevent the normal threshold m from working?

5.6 Secret sharing with disenrollment

How can we add/delete shareholders without starting all over again?