

## Merkle's Puzzles

See:

Merkle, *Secrecy, Authentication, and Public Key Systems*, UMI Research press, 1982

Merkle, *Secure Communications Over Insecure Channels*, CACM, Vol. 21, No. 4, pp. 294-299, April 1978

## Merkle's Puzzles

Merkle's puzzles

שיטת החידות של מרקל

1. The first hint that two parties have computational advantage over attackers
2. Exchanges keys over insecure channels
3. Uses puzzles

## Puzzles

- A **Puzzle** is a cryptogram, which is designed to be breakable
- Breaking the cryptogram reveals the puzzle information hidden in the plaintext
- A cryptogram can be encrypted using any secure cipher  $E$ . Examples:  $E = DES$ ,  $E = AES$
- The complexity of solving the puzzle can be chosen by selecting the size of the puzzle keys. For example, for  $2^{20}$  complexity, 20-bit puzzle keys can be used (the other key bits of  $E$  are fixed to some agreed value)
- The plaintext of the puzzle should include redundancy to allow the users to solve it. Such redundancy is included by incorporating an agreed fixed value  $S$ , whose length suffices to ensure uniqueness of the solved puzzle key

## Puzzles (cont.)

**Definition:** A **puzzle** is  $E_{PK}(S\|ID\|K)$  where

- $\|$  denotes concatenation
- $PK$  is an  $n$ -bit puzzle key
- $S$  is an agreed fix value used in all the puzzles, whose length is at list  $n$  bits. It ensures uniqueness of the puzzle keys
- $ID$  is an  $n$ -bit puzzle identifier, unique for each puzzle
- $K$  is a random value, whose size equals the size of the required common key — a keys of one of the puzzles will become the common key
- $E$  is a block cipher with  $n$ -bit (or longer) keys, and sufficiently large blocks
- $ID, K$  are kept secret, and the only way to recover them is to solve the puzzle

## Puzzles (cont.)

**Remark:**

We use two kinds of keys:

- The puzzle key  $PK$  is the key under which the puzzle is encrypted
- $K$  is hidden in the puzzle, and becomes later the result of the protocol

$n$  is a security parameter that controls the difficulty of solving the puzzle

## The Protocol

Basically the protocol is:

1. Alice generates a table of  $N = 2^n$  keys

ID	K
$ID_1$	$K_1$
$\vdots$	$\vdots$
$ID_N$	$K_N$

2. She sends the table to Bob where each row is hidden in a puzzle
3. Bob selects a row and tells Alice the ID of that row
4. Alice fetches the  $K$  of that row

## The Protocol (cont.)

1. A, B wish to select a common secret key
2. A, B agree on  $n$  and  $S$ ,  $|S| \geq n$
3. A generates  $N = 2^n$  puzzles  $P_0, P_1, \dots, P_{N-1}$ , where  $P_i = E_{PK_i}(S\|ID_i\|K_i)$ ,  $PK_i, K_i$  are randomly chosen, and  $ID_i$  is a unique identifier of the puzzle
4. A sends all the puzzles to B. The attacker  $E$  can listen to all the communication
5. B receives  $N$  puzzles, and selects one puzzle  $P_i$  randomly
6. B solves  $P_i$  by trying all  $N$  possible puzzle keys  $PK$  and verifying the redundancy  $S$ . B recovers the puzzle key  $PK_i$ , and the secret values  $ID_i, K_i$
7. B sends  $ID = ID_i$  to A; A identifies the puzzle  $P_i$  by  $ID$
8. A, B agree that  $K = K_i$  is the common secret key

## The Protocol (cont.)

**Complexity:**

- A invests  $O(N)$  time for generating  $N$  puzzles
- B invests  $O(N)$  time for solving one puzzle
- The communication complexity is  $O(N)$
- An attacker has to invest  $O(N^2)$  time to solve the puzzles

### The Protocol (cont.)

#### Parameters:

- $n = 20$ ,  $N = 2^{20} \approx 1000000$  is sufficiently small such that computing and transmitting  $O(N)$  puzzles, and solving one puzzle, can be done relatively fast, but recovering the common key by an eavesdropper takes  $10^{12} \approx 2^{40}$  steps
- In order to have security for periods of years and beyond, we need to choose  $n > 32$

### The Protocol (cont.)

#### The Legal Users' Advantage:

- Merkle's puzzles suggest that the legal users have computational advantage over attackers
- The advantage is quadratic ( $N$  for legal users versus  $N^2$  for attacker)
- When a high security is required, such as  $n \geq 32$ , the legal users have to invest a lot of time in the protocol
  
- Is there another scheme with an exponential advantage?  
We will discuss it in the next lecture

### Implementation Notes

First notice that in most ciphers, the block size may not be large enough to contain  $S\|ID\|K$ .

Therefore, some implementation changes may be necessary.

We now show that although the protocol is secure, a careless implementation can be totally insecure.

Implement the puzzles using DES, assuming  $n = 32$ . Let the puzzle be

$$\text{DES}_{PK}(S), \text{DES}_{PK}(ID), \text{DES}_{PK}(K).$$

**This is insecure:** The attacker can encrypt  $\text{DES}_{PK}(S)$  in advance under all possible  $PK$ 's, correlate the first words of the puzzles to the  $PK$ 's, and compute the  $ID$  for each puzzle. It can reduce the complexity to  $O(N)$ .

### Implementation Notes (cont.)

**Possible solution:** Encrypt the first word under  $K$  instead:

$$\text{DES}_K(S), \text{DES}_{PK}(ID), \text{DES}_{PK}(K)$$

**This is also insecure:** After receiving  $ID$ , the attacker can encrypt  $ID$  under all possible  $PK$ 's, correlate the puzzles and the  $PK$ 's, compute  $K$  and verify correctness of  $S$ . The total complexity is also  $O(N)$ .

### Implementation Notes (cont.)

**A Better Solution:** Encrypt first two words under  $K$ :

$$\text{DES}_K(S), \text{DES}_K(ID), \text{DES}_{PK}(K)$$

**Or for  $S \neq 0$ :**

$$\text{DES}_{PK}(S \oplus K), \text{DES}_{PK}(ID \oplus K), \text{DES}_{PK}(K)$$

**Or:** Use a cipher  $E$  with a sufficiently large block size, such as AES, where  $PK, S, ID$  are 32-bit values, and  $K$  is a 64-bit value. In this case a puzzle is simply

$$\text{AES}_{PK}(S\|ID\|K).$$

But we cannot select a 128-bit  $K$  in this implementation.

However, in order to distribute a 128-bit key, we can perform this implementation twice.

### Additional Notes

- The puzzles do not have to be secret to ensure a common secret key. Each user  $A$  can publish a set of puzzles in a public file, that everybody can read, but not modify. Then, every user  $B$  can select a puzzle and share a secret key with  $A$
- $B$  can authenticate  $A$  by sharing a key and asking  $A$  to encrypt some value that  $B$  selected. Only  $A$  can succeed, assuming the public file manager verifies ownership correctly. Even the manager cannot recover the keys!
- Mutual authentication:  $A, B$  can share two keys  $K_A$  and  $K_B$ , one using puzzles of  $A$  and one using puzzles of  $B$ , and then use  $K_A \oplus K_B$  as the common secret key