Secure data storing in a pool of vulnerable servers

Marcin Gogolewski (Cryptology Center, AMU Poznań)

Mirosław Kutyłowski (Wrocław University of Technology)

Secure storing data

- contents protection: encryption (size?)
- protection against destroying: redundancy, random locations, distributed systems
- protection against administrators: anonymity
- active adversary

Active adversary

- may attack, overtake or destroy any location
- cannot brake strong cryptographic codes
- cannot influence random sources of the user

Environment

- shared communication channel providing anonymity
- a number of data servers
- data stored encrypted sufficiently
- public keys of servers known

Design goals

- minimize communication
- 100% success or failure

Naive solution

- user chooses k servers at random
- using public keys informs these servers about symmetric key
- transmit ion of data encrypted with symmetric key
- receipts
- protocol if something goes wrong

Onion solution - creating of an onion by Alice

1. Alice chooses $j_1, \ldots, j_k \leq n$ at random .
2. Alice chooses at random a symmetric key K, then a random key K_0 of the same length, and finally computes $K_1 = K$ XOR K_0
3. Alice chooses at random strings $SIG(M)$, $r_0, r_1, \ldots, r_k$, and $s_1, \ldots, s_k$ of a fixed length l.
4. The onion C_k is created by Alice. The *kernel* C_0 consists of

$$r_0, s_1, \ldots, s_k, K_1, SIG(M) \ .$$

Then for $i \leq k$ the onion C_i has the form

$$E_{P(j_i)}\left(r_i, s_i, \mathcal{F}, K_0, C_{i-1}\right)$$

$E_X(Y)$ = ciphertext obtained from Y with an asymmetric key X $P(u)$ = public key of server S_u, and $\mathcal{F}$ sufficiently long fixed sequence .

Alice sends

- the onion C_k,
- the message M encrypted with a key K using a symmetric encryption algorithm, with the string $SIG(M)$ in front of it.

Processing an onion by the servers

if X transmitted, then server S_i:

1. decrypts X with its private key, if the plaintext obtained has not the form

$$r, s, \mathcal{F}, L, C$$

   where r, s are strings of length l, L is the key for symmetric algorithm, and C is a ciphertext, or forms a kernel of an onion, then S_i stops processing X,

2. if decrypted ciphertext is not a kernel, then S_i associates s with key L and stores it for a later use, and publishes C on the bulletin board.

Processing an onion by the servers

3. if decrypted ciphertext is a kernel of an onion

$$r_0, s_1, \ldots, s_k, K_1, SIG(M)$$

then S_i truncates r_0 from the kernel and puts the kernel on the bulletin board.

Storing data by servers

- S_i detects on a bulletin board a truncated kernel containing a string s it has saved together with K_0 while processing an onion,
- it computes $K := K_0$ XOR K_1,
- when a ciphertext with $SIG(M)$ transmitted, it decrypts it with K and stores the result.

Conclusion

- optimal number of messages
- disrupting an onion - nobody stores data