

Markle's Puzzle

The approach in Markle's puzzle is to trade communication overhead for computation overhead.

3-9.pdf
939.3 KB

ⓐ ⓑ ⓑ

5

3.8 Keyed Hash

→ In this the file is hashed with some secret key.

3.9 Markle's Puzzle

Step 1: Generate 128 bit keys.

128 bit keys = $k_1, k_2, k_3, \dots, k_n$.

64 bit keys = $s_1, s_2, s_3, \dots, s_n \rightarrow$ secret keys.

sequence # = $t_1, t_2, t_3, \dots, t_n$.

Step 2:

Select symmetric Algo 64 bit key used encrypt
secret key.

~~Est~~ $E_{s_1}(t_1, k_1, h(k_1)) = m_1$

$E_{s_2}(t_2, k_2, h(k_2)) = m_2$

\vdots

$E_{s_n}(t_n, k_n, h(k_n)) = m_n$

Step 3: Alice will dump $m_1 \dots m_n$ to Bob.

Step 4: Bob randomly pick a message m_i .

Step 5: Bob uses brute force to open m_i using all keys.

Step 6: Bob opens message and choose seq. t_i and sends to Alice.

Now both sides will know k_i .

(IMQ 1)

→ If the attacker wants to open the message he has to try for n times
since there are n messages & 64 bit key
∴ bad guy has to crack $n \cdot 2^{64}$

bob has to crack $\sim 2^{64}$

→ If Alice can generate 2^{36} messages

Attacker computation overhead = $2^{36} \cdot 2^{64}$
 $= 2^{100}$

∴ If Bob need 2^{36} seconds to crack, Bad guys have 200 seconds.

Block 5 - Random Numbers.

Dual Encryption:

K_i^A try to establish a key: 128 bit key. K_i^B
trade communication overhead for computation overhead.

generate many (k_i)
128 bit key: k_1, k_2, \dots, k_n
64 bit key: s_1, s_2, \dots, s_n
sequence #: t_1, t_2, \dots, t_n

$E_{s_1}(t_1, k_1, h(k_1)) = m_1$
 $E_{s_2}(t_2, k_2, h(k_2)) = m_2$
 \vdots
 $E_{s_n}(t_n, k_n, h(k_n)) = m_n$

m_1, m_2, \dots, m_n

randomly pick message m_i
B brute force to atk m_i
64 bit t_i, k_i, h

A, I choose t_i

Dual Encryption

A pub_A pri_A

whoever pub_A

B pub_B pri_B

A, B. $(Sign_{pri_A}(E_{pub_B}(msg)))$

A, B $(E_{pub_B}(Sign_{pri_A}(msg)))$

Bob