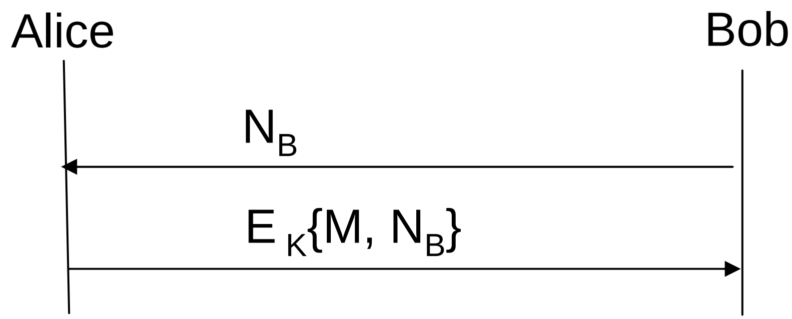
Formal Analysis: MAP1

Kaisa Nyberg

6.10.2006

Textbook: W. Mao. Modern Cryptography T&P; 17.2-3

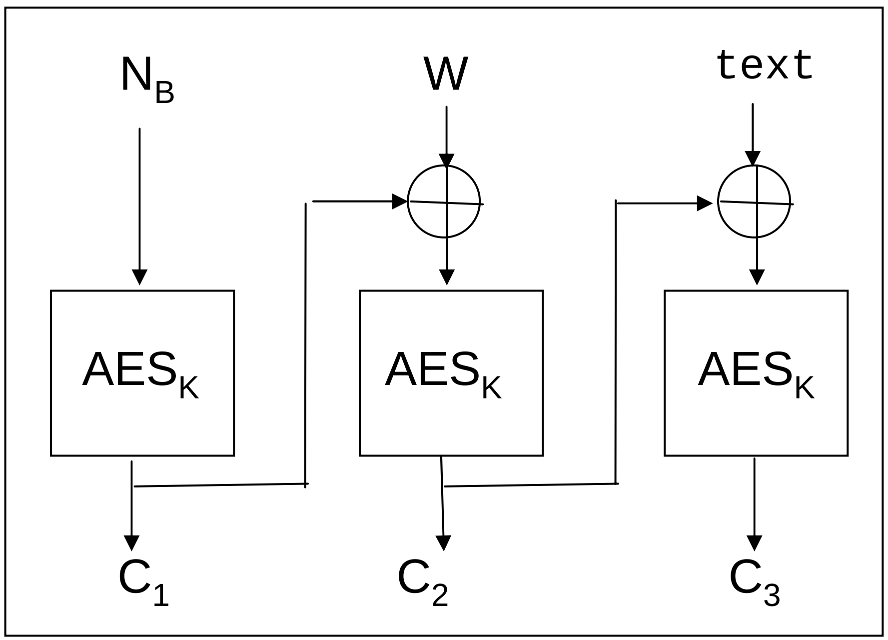
Challenge-Response



- Potentially harmful: May provide to Malice *Oracle Services* to compute function E_K with unknown secret key K
  - *encryption oracle if* E_K *is encryption*
- Insufficient: Encryption does not provide integrity

Non-integrity of CBC encryption

- Bob wants to verify the liveness of Alice's love and receive a fresh new key
- Alice's message M = W ||"I love you" , where W is a128-bit key
- Encryption is CBC with 128-bit block cipher (AES)
- N_B is a 128-bit value; $(C_1, C_2, C_3) = E_K(N_B, M)$




```
text = 49 20 6c 6f 76 65 20 79 6f 75
   Δ = 00 00 04 0e 02 00 00 00 00 00
text'= 49 20 68 61 74 65 20 79 6f 75
```

- Malice changes the second ciphertext block to $C_2' = C_2 \oplus \Delta$
- After decryption Bob reads

  M' = W' || "I hate you"

  where W' is a random 128-bit value

Formal model for a symmetric key protocol

- Parties A and B share a protocol Π and a secret key of length k
- The i th run of the protocol is labelled as Π^i
- Malice uses A and B as oracles and can run with them simultaneously more than one protocol runs and use any legal identities in its communication. Malice uses A as a black box (oracle) $\Pi^r_{A,B}$ and B as a black box oracle $\Pi^s_{B,A}$
- Malice can make A and/or B initiate the protocol runs or initiate runs by himself.

Matching conversations

Let

$$\tau_0 < \tau_1 < \tau_2 < \ldots \tau_{2t-2} < \tau_{2t-1}$$

be a time (counter) sequence recorded by party A when it converses with B. Let

$$conv = (\tau_0, m'_0, m_1), (\tau_2, m'_1, m_2), \ldots, (\tau_{2t-2}, m'_{t-1}, m_t)$$

be the conversation recorded by A. We say that party B has a matching conversation $conv'$ with A if $conv'$ has the form

$$conv' = (\tau_1, m_1, m'_1), (\tau_2, m_2, m'_2), \ldots, (\tau_{2t-1}, m_t, m'_t).$$

Here the first message is a received one, and the second message is a sent one. In particular, $m'_0 = m'_t = \text{empty}$.

Security definitions

- The *accept* condition is defined by each oracle's own view of the conversation.
- Definition. We say that $\Pi(1^k; A,B\})$ is a secure mutual authentication protocol between A and B if the following statement holds except for a negligible probability in k : oracles $\Pi^r_{A,B}$ and $\Pi^s_{B,A}$ both reach the *accept* decision if and only if they have matching conversations.
- If protocol is correct, and the parties have matching conversations then they reach the accept state.
- Definition. We say that Malice wins if both $\Pi^r_{A,B}$ and $\Pi^s_{B,A}$ reach the *accept* decision while they do not have matching conversations.

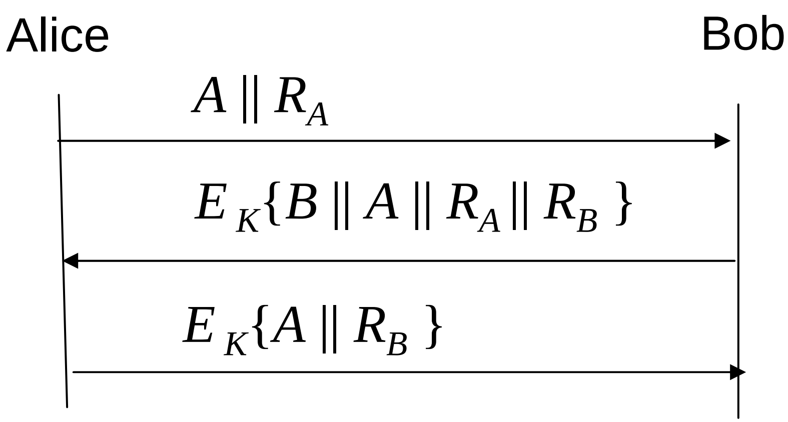
  Note: Sometimes it is more appropriate to say that Malice wins if at least one of the oracles reach the accept state.
- Definition. We say that $\Pi(1^k;\{A,B\})$ is a secure mutual authentication protocol between A and B if Malice cannot win with a non-negligible probability in k.

Pseudorandom function family

- Protocol analysis makes use of idealized cryptographic primitives that are formally defined to satisfy certain cryptographic properties
- Example: Keyed pseudo-random function prf_K

  Definition: A function family $\{prf_K\}$ with key length k is a *pseudorandom function family,* if any adversary A (whose resources are bounded by a polynomial in k) cannot distinguish between a function prf_K (where K is chosen randomly and kept secret) and a purely random function only with negligible probability. That is, a function f is chosen to be either prf_K for a random K or a purely random function with the same input domain and output range. Next A gets to ask the value of f on a number (bounded polynomially in k) of points. Nonetheless A should be unable to tell whether f is random or pseudorandom.
- A and B are said to share a purely random function if for each input A and B (after computation the function) get the same randomly selected output.

MAP1



Denote:

$E_K\{M\} = M \| prf_K\{M\}$

tag $prf_K\{M\}$ has k bits

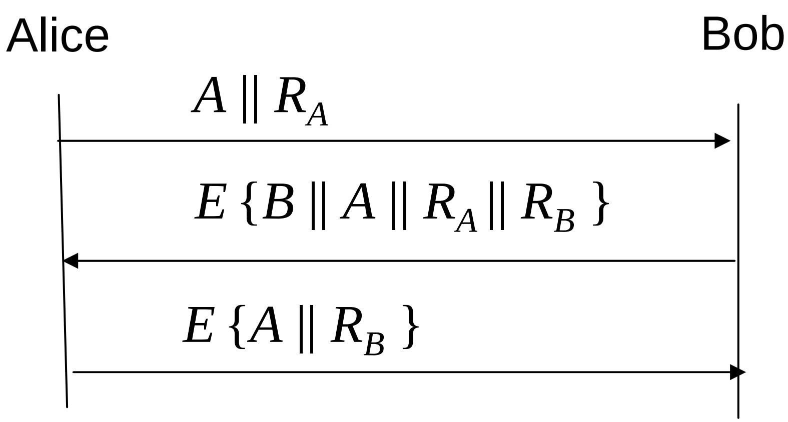
$$conv_A = (\tau_0, empty, A \| R_A), (\tau_2, E_K\{B \| A \| R_A \| R_B\}, E_K\{A \| R_B\})$$

$$conv_B = (\tau_1, A \| R_A, E_K\{B \| A \| R_A \| R_B\}), (\tau_3, E_K\{A \| R_B\}, empty)$$

Consider two experiments:

- Exp_0: MAP1 is run with prf_K replaced by a truly random function g with k -bit output shared by Alice and Bob
- Exp_1: MAP1 is run with prf_K

Exp_0



Denote:

$E\{M\} = M \| g\{M\}$

$g\{M\}$ has k bits

$$conv_A = \left(\tau_0, empty, A \| R_A\right), \left(\tau_2, E\{X \| A \| R_A \| R_X\}, E\{A \| R_X\}\right)$$

Because of R_A Alice sees that $E\{X \| A \| R_A \| R_X\}$ cannot have been created by anybody else than Bob with probability larger than 2^{-k}

$$conv_B = \left(\tau_1, Y \| R_Y, E\{B \| Y \| R_Y \| R_B\}\right), \left(\tau_3, E\{Y \| R_B\}, empty\right)$$

Bob sees that $E_K\{Y \| R_B\}$ cannot have been created by anybody else than Alice with probability larger than 2^{-k}. Bob accepts only if in $conv_B$ the identity Y is the same at τ_1 and τ_3.

Exp_1 and the distinguisher

- Exp1 = MAP1 with a keyed prf_K
- Assume now that Malice is good at MAP1 and can win with a probability larger than 2^{-k}
- Then Charlie can run a polynomial-time test and use Malice to distinguish pseudo-random functions from truly random functions as follows.
- Denote $f_0 = g$, $f_1 = prf_K$. A coin δ is flipped and Charlie is given f_δ. Then Charlie implements all oracles Malice needs to run its attack against MAP1 using f_δ as the function to compute tags. Assume that Malice wins in MAP1 with probability $p > 2^{-k}$. If Malice wins, Charlie's guess is $\delta = 1$, otherwise his guess is $\delta = 0$. Then Charlie's advantage is

$$\text{Adv(Charlie)} = \Pr[\text{guess} = 1 \mid \delta = 1] - \Pr[\text{guess}=1 \mid \delta = 0]$$

$$= \Pr[\text{Malice wins in MAP1}] - \Pr[\text{Malice wins at random}]$$

$$\geq p - 2^{-k} > 0$$

Discussion

- Security proof in *random oracle model* uses an idealized version of a cryptographic function
- Advantage: Protocol properties can be analyzed independently from the properties of the cryptographic primitives
- Disadvantage: The separation may break important dependencies and interactions between the protocol structure and the cryptographic primitives.