Formal specification of authentication protocols

Olli Pottonen
olli.pottonen@tkk.fi

29.11.2007 (revised 3.12.2007)

T-79.5502 Advanced Course in Cryptology

Overview

- Proper use of cryptographic transformations (17.2)
- Formal specification and security proofs (17.3)

Cryptographic transformations

- **Encryption**: confidentiality, no data integrity (usually)
- Message M, key K: $\{M\}_K$
- Many authentication protocols misuse encryption.

Authentication via encryption-decryption

Example: Needham-Schroeder Public-Key Authentication Protocol:

1. Alice $\rightarrow$ Bob: $\{N_A, Alice\}_{K_B}$
2. Bob $\rightarrow$ Alice: $\{N_A, N_B\}_{K_A}$
3. Alice $\rightarrow$ Bob: $\{N_B\}_{K_B}$

Underlying assumption: only Alice can decrypt messages $\{M\}_{K_A}$, and only Bob messages $\{M\}_{K_B}$.

Lowe's attack

1. Alice → Malice: $\{N_A, Alice\}_{K_M}$

1'. Malice("A") → Bob: $\{N_A, Alice\}_{K_B}$

2'. Bob → Malice("A"): $\{N_A, N_B\}_{K_A}$

2. Malice → Alice: $\{N_A, N_B\}_{K_A}$

3. Alice → Malice: $\{N_B\}_{K_M}$

3'. Malice("A") → Bob: $\{N_B\}_{K_B}$

Result: Bob mistakes Malice for Alice.

Harmfulness of encryption-decryption

Alice acts as an decryption/encryption oracle, which Malice can use for breaking security.

Encryption does not provide data integrity

- Encryption is usually carried out block at a time.
- Malice may change some of the blocks and leave others untouched.
- For example, consider CBC: $C_0 \leftarrow IV$; $P_i \leftarrow \mathcal{D}(C_i) \oplus C_{i-1}$.
- If Malice changes block C_i, the decrypted blocks P_i and P_{i+1} are affected.
- P_{i+1} changes in predictable way.
- Malice needs encryption oracle to change also P_i in predictable way.

Back to Cryptographic transformations

- **Encryption**: confidentiality, no data integrity (unless non-malleable)
- Message M, key K: $\{M\}_K$
- **One-way transform** (MAC, digital signature): data integrity and message source identification, no confidentiality
- Message M, key K: $[M]_K$
- We assume $[M]_K = (M, prf_K(M))$, where prf_K is a keyed pseudo-random function.

Needham-Schroeder revisited

How to fix Needham-Schroeder Public-key Authentication Protocol:

1. Alice $\rightarrow$ Bob: $[\{N_A\}_{K_B}, Alice]_{K_A}$
2. Bob $\rightarrow$ Alice: $[\{N_A, N_B\}_{K_A}]_{K_B}$
3. Alice $\rightarrow$ Bob: $[\{N_B\}_{K_B}]_{K_A}$

Formal specification of authentication protocols - the Bellare-Rogaway Model

- Honest participant: polynomial-time function $\Pi(1^k, i, j, K, conv, r)$, where
  - k: the security parameter (key size)
    - Computation must be polynomial-time with respect to 1^k
  - i: identity of the participant
  - j: identity of the intended communication partner
  - K: long-lived symmetric key shared by i and j
  - $conv$: conversation, i. e., concatenation of all sent and received messages
  - r: random input generated by the participant

Formal specification (cont.)

- Execution of $\Pi(1^k, i, j, K, conv, r)$ yields
  - m: the message sent out - $m \in \{0,1\}^* \cup \{\text{no output}\}$
  - σ: decision - $\sigma \in \{\text{Accept}, \text{Reject}, \text{Undecided}\}$
  - α: the private output - $\alpha \in \{0,1\}^* \cup \{\text{no output}\}$

- $\Pi^s_{i,j}$ denotes participant i attempting to authenticate j in a session labeled by s.

Formal specification: Malice

- Malice has unlimited access to *oracles* $\Pi^s_{i,j}, \Pi^t_{j,i}$, with values of $i, j, s, t, conv$ supplied by Malice.

- The key K and random values r not known by Malice.

- Malice gets message m and decision δ, not private input α.

Formal specification: security definition

- **Matching conversations**: The messages received by $\Pi^s_{i,j}$ were sent by $\Pi^t_{j,i}$ in the correct order, and vice versa.

- $conv = (\tau_0, \text{``}'', m_1), (\tau_2, m'_1, m_2), \ldots, (\tau_{2t-2}, m'_t, m_t)$ and $conv' = (\tau_1, m_1, m'_1), (\tau_3, m_2, m'_2), \ldots, (\tau_{2t-1}, m_t, \text{``}'')$ are matching (here Alice sends the first and last messages).

- Malice wins, if $\Pi^s_{i,j}$ and $\Pi^t_{j,i}$ accept while not having matching conversations.

- Protocol is secure, if probability of Malice winning in polynomial time is negligible.

$MAP1$

Mutual Authentication Protocol 1 ($MAP1$):

1. Alice $\rightarrow$ Bob: $A \parallel R_A$

2. Bob $\rightarrow$ Alice: $[B \parallel A \parallel R_A \parallel R_B]_K$

3. Alice $\rightarrow$ Bob: $[A \parallel R_B]_K$

Recall that $[M]_K = (M, prf_K(M))$. K, R_A, R_B and $prf_K(M)$ have length $\Omega(k)$.

Proof of security

- First assume that $[M]_K = (M, rf_K(M))$, where rf_K is a truly random function.

- Alice accepts only if she sent $A \parallel R_A$ and received $[B \parallel A \parallel R_A \parallel R_B]_K$. Malice can guess $rf_K(B \parallel A \parallel R_A \parallel R_B)$ with negligible probability $\Rightarrow$ $rf_K(B \parallel A \parallel R_A \parallel R_B)$ was computed by Bob $\Rightarrow$ Bob received $A \parallel R_A$ and sent $[B \parallel A \parallel R_A \parallel R_B]_K$.

- Bob received $A \parallel R_A$ and sent $[B \parallel A \parallel R_A \parallel R_B]_K$. He accepts only if he receives $[A \parallel R_B]_K$. In that case the conversations are matching.

Proof of security (cont.)

- Now consider pseudorandom function prf_K. By definition, a secure pseudorandom function prf_K can not be distinguished from a truly random rf_K with non-negligible advantage.

- Consider the following algorithm for distinguishing prf_K and rf_K:
  - Charlie is given function g_K. He lets $[M]_K = (M, g_K(M))$ and simulates Malice and the oracles in the MAP1 protocol with function g_K. The assumption is that Malice succeeds in MAP1 with $g_K = prf_K$ with non-negligible probability, but if $g_K = rf_K$, then Malice's chances are negligible. If Malice wins, Charlies guesses "pseudorandom", otherwise Charlie guesses "random".

Proof of security (cont.)

- Now Charlie's advantage is $Adv(\text{Charlie})$

$$= |P(guess = pseudornd, g_K = prf_K) - P(guess = pseudornd, g_K = prf_K)|$$

$$= |P(g_K = prf_K)P(guess = pseudornd|g_K = prf_K)$$
$$-P(g_K = rf_K)P(guess = pseudornd|g_K = rf_K)|$$

$$= \frac{1}{2}|P(\text{Malice wins in MAP1}|g_K = prf_K) - P(\text{Malice wins in MAP1}|g_K = rf_K)|$$

$$= \frac{1}{2}|p_p(k) - p_r(k)|,$$

where $p_p(k) = P(\text{Malice wins in MAP1}|g_K = prf_K)$ and $p_r(k) = P(\text{Malice wins in MAP1}|g_K = rf_K)$.

Proof of security (cont.)

- By the first part of the proof, $p_r(k)$ is negligible. Hence, if $p_p(k)$ is non-negligible, then $Adv(\text{Charlie})$ is non-negligible, as shown on the next slide.

- So we have shown that since MAP1 is secure with truly random function, then MAP1 is also secure with pseudorandom function prf_K, for otherwise MAP1 could be used to construct an efficient algorithm for distinguishing between pseudorandom and random functions.

Proof of security (technical details)

- We must yet show that $|p_r(k) - p_p(k)|$ is non-negligible when $p_r(k)$ is negligible and $p_p(k)$ is not. Recall that function f is negligible if $1/f(k)$ is not polynomially bound.

- Since $1/p_p(k)$ is polynomially bound and $1/p_r(k)$ is not, $1/p_p(k) \leq \frac{1}{2}1/p_r(k)$ for large enough k. Thus for large enough k, $p_r(k) \leq \frac{1}{2}p_p(k)$, and $Adv(\text{Charlie}) = \frac{1}{2}|p_r(k) - p_p(k)| \geq \frac{1}{4}p_p(k)$, which is non-negligible.