Bellare-Rogaway protocol verification model

T-79.5502 Advanced Course in Cryptology

Samuli Larvala

Overview

- Introduction
- The model
- Two-party mutual authentication
- Security analysis
- Two-party authenticated key exchange
- Conclusion

Introduction

- M. Bellare, P. Rogaway "Entity Authentication and Key Distribution," CRYPTO '93
- Describes the first provably secure protocol for entity authentication and key distribution
- Entity authentication is the process by which an agent in a distributed system gains confidence in the identity of a communcation partner
- Key distribution gives the partners a session key for message confidentiality, integrity and other needs.

Previous Work

- Needham-Schroeder authentication protocol
- Propose, attack, fix, attack, fix, ...
- Encryption-decryption paradigm
- Confidentiality vs. data integrity

Traditional Needham-Schroeder Symmetric-key Authentication

- Traditional Needham-Schroeder relies on encryption

$$\begin{aligned}
&\text{Alice} \rightarrow \text{Trent} : \text{Alice}, \text{Bob}, N_A \\
&\text{Trent} \rightarrow \text{Alice} : \{K, N_A, \text{Bob}, \{K, \text{Alice}\}_{K_{BT}}\}_{K_{AT}} \\
&\text{Alice} \rightarrow \text{Bob} : \text{Trent}, \{K, \text{Alice}\}_{K_{BT}} \\
&\text{Bob} \rightarrow \text{Alice} : \{N_B\}_K \\
&\text{Alice} \rightarrow \text{Bob} : \{N_B - 1\}_K
\end{aligned}$$

- Without integrity checking the data can be modified although encrypted

Refined Needham-Schroeder Symmetric-key Authentication

- The refined Needham-Schroeder authentication minimizes use of encryption using message authentication

$$
\begin{aligned}
\text{Alice} \rightarrow \text{Trent} &: \ \text{Alice}, \text{Bob}, N_A \\
\text{Trent} \rightarrow \text{Alice} &: \ [\{K\}_{K_{AT}}, N_A, \text{Alice}, \text{Bob}]_{K_{AT}} \\
&\quad\ [\{K\}_{K_{BT}}, T, \text{Alice}, \text{Bob}]_{K_{BT}} \\
\text{Alice} \rightarrow \text{Bob} &: \ [\{K\}_{K_{BT}}, T, \text{Alice}, \text{Bob}]_{K_{BT}} \\
\text{Bob} \rightarrow \text{Alice} &: \ [N_B]_K \\
\text{Alice} \rightarrow \text{Bob} &: \ [N_B - 1]_K
\end{aligned}
$$

- Confidentiality service is provided at the minimum level

Bellare-Rogaway Model

- All communication between the parties is under the control of the adversary who can read, create, modify, delay, replay messages
- The adversary can initiate new authentication sessions at any time
- Each party will be modeled by an oracle which the adversary can run
- The oracles never directly interact with one another
- The protocol is secure if the only way that an adversary can get a party to accept is by faithfully relaying messages (benign adversary)

Authenticating participants

- Players are modeled by a function $\Pi(1^k, i, j, a, \kappa, r)$:

1^k	Security parameter - $k \in \mathbb{N}$
i	Identity of the initiator - $i \in I$
j	Identity of the responder - $j \in I$
a	Secret information - $a \in \{0,1\}^*$
κ	Conversation so far - $\kappa \in \{0,1\}^*$
r	Random input of the sender - $r \in \{0,1\}^\infty$

- I is a set of identities which defines the players who can participate in the protocol
- The adversary is not a player ($\notin I$)
- The function Π runs in polynomial time

Player Function Response

- The execution of $\Pi(1^k, i, j, a, \kappa, r)$ yields a response (m, δ, α):

m	Next output message - $m \in \{0,1\}^* \cup \{*\}$
δ	The decision of the oracle - $\delta \in \{\mathrm{A}, \mathrm{R}, *\}$
α	Private output to the player - $\alpha \in \{0,1\}^* \cup \{*\}$

Key Generator

- The protocol also includes a key generator $\mathcal{G}(1^k, i, r_G)$ for generating keys:

1^k	Security parameter - $k \in \mathbb{N}$
i	Identity of the protocol participant - $i \in I \cup \{E\}$
r_G	infinite string - $r_G \in \{0,1\}^\infty$

- Generates keys for all the protocol participants
- In this protocol players share a common secret key $\mathcal{G}(1^k, i, r_G) = \mathcal{G}(1^k, j, r_G)$

The Protocol

Running the protocol in the presence of an adversary E, using security parameter k, means performing the following experiments:

- Choose a random string $r_G \in \{0,1\}^\infty$ and set $a_i = \mathcal{G}(1^k, i, r_G)$, for $i \in I$, and set $a_E = (1^k, E, r_G)$
- Choose a random string $r_E \in \{0,1\}^\infty$ and for each $i, j \in I$, $s \in \mathbb{N}$, a random string $r^s_{i,j} \in \{0,1\}^\infty$
- Let $\kappa^s_{i,j} = \lambda$ for all $i, j \in I$ and $s \in \mathbb{N}$
- Run adversary E on input $(1^k, a_E, r_E)$. E queries (i, j, s, x) and oracle $\Pi^s_{i,j}$ computes $(m, \delta, \alpha) = \Pi(1^k, i, j, a_i, \kappa^s_{i,j}.x, r^s_{i,j})$, answers with (m, δ) and $\kappa^s_{i,j}$ gets replaced by $\kappa^s_{i,j}.x$

Conversations

- The Adversary's i-th query to an oracle is said to occur at time $\tau = \tau_i \in \mathbb{R}$. For $i < j$ we demand that $\tau_i < \tau_j$
- The conversation κ of oracle $\Pi^s_{i,j}$ is a sequence of messages ordered by time $\tau_1 < \tau_2 < \cdots < \tau_R$ for some $R \in \mathbb{N}$
- Oracle $\Pi^s_{i,j}$ has conversation $K = (\tau_1, \alpha_1, \beta_1), (\tau_2, \alpha_2, \beta_2), (\tau_3, \alpha_3, \beta_3), \ldots, (\tau_m, \alpha_m, \beta_m)$
- If $\alpha_1 = \lambda$, $\Pi^s_{i,j}$ is an initiator oracle
- If α_1 is any other string, $\Pi^s_{i,j}$ is a responeder oracle

Matching Conversations

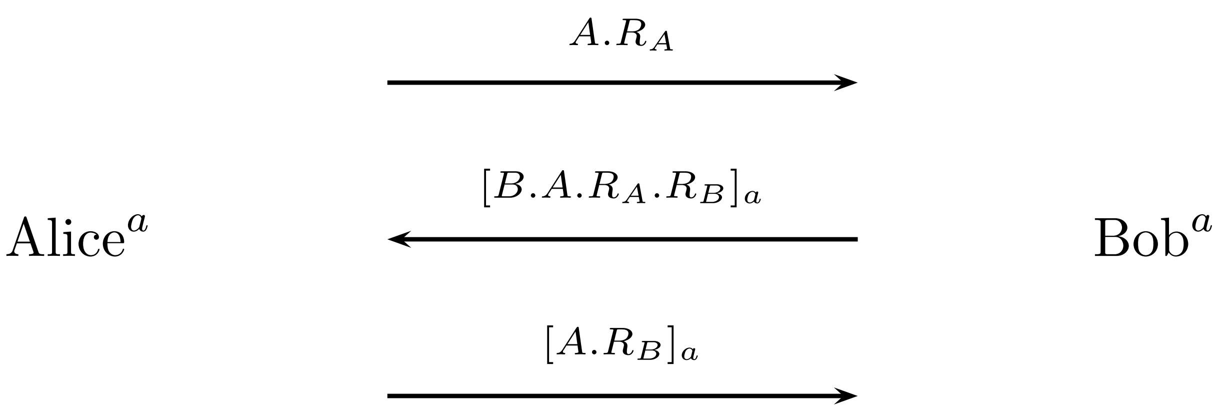
- Consider two oracles $\Pi^s_{i,j}$ and $\Pi^t_{j,i}$ engage in a conversation
- If
  $\kappa^s_{i,j} = (\tau_0, \lambda, m_1), (\tau_2, m'_1, m_2), (\tau_4, m'_2, m_3), \ldots, (\tau_{2t-2}, m'_{t-1}, m_t)$
  and
  $\kappa^t_{j,i} = (\tau_1, m_1, m'_1), (\tau_3, m_2, m'_2), (\tau_5, m_3, m'_3), \ldots, (\tau_{2t-1}, m_t, \lambda)$
  parties i, j have a matching conversation

Mutual Authentication

- Two parties i, j accept when they have a matching conversation
- No-Matching$^E(k)$ is the event that there exists i, j, s such that $\Pi^s_{i,j}$ accepted yet there is no $\Pi^t_{j,i}$ with matching conversation
- Π is a secure mutual authentication protocol if for any polynomial time adversary E
  1. If oracles $\Pi^s_{i,j}$ and $\Pi^t_{j,i}$ have matching conversations, both oracles accept
  2. The probability of No-Matching$^E(k)$ is negligible

MAP1

- Let f_a be a pseudo random function $\{0,1\}^{\leq L(k)} \to \{0,1\}^{l(k)}$ specified by key a and $L(k) = 4k$ and $l(k) = k$
- For any string $x \in \{0,1\}^{\leq L(k)}$ define $[x]_a = (x, f_a(x))$ to denote the authentication of message x



MAP1 is Secure

- Suppose f is a pseudorandom function. MAP1 based on f is a secure mutual authentication protocol
- Running the adversary E with MAP1 using a PRF f_a is the real experiment
- Running the adversary E with MAP1 using a truly random function g is the random experiment
- The probability that the adversary E is successful in the random MAP1 experiment is at most $T_E(k)^2 \cdot 2^{-k}$ where $T_E(k)$ denotes the polynomial bound on the number of oracle calls made by E

The Random MAP1 Experiment (part 1)

Claim: The probability that the initiator oracle $\Pi^s_{A,B}$ accepts without a matching conversation is at most $T_E(k) \cdot 2^{-k}$

Proof: Suppose at time τ_0 oracle $\Pi^s_{A,B}$ send the flow R_A. Let $\mathcal{R}(\tau_0) = \{R'_A \in \{0,1\}^k : \exists \tau, t$ such that $\Pi^t_{B,A}$ was given R'_A at time $\tau < \tau_0.\}$. If $\Pi^s_{A,B}$ accepts, then at time $\tau_2 > \tau_0$ is must have receive $[B.A.R_A.R_B]_g$ for some R_B. The probability that E can compute it is at most 2^{-k}. The output came from oracle $\Pi^t_{B,A}$ which received R_A. The probability of this happening before τ_0 ($R_A \in \mathcal{R}(\tau_0)$) is at most $[T_E(k) - 1] \cdot 2^{-k}$. If it happened after τ_0 then we have a matching conversation. The probability that $\Pi^s_{A,B}$ accepts without a matching conversation is at most $T_E(k) \cdot 2^{-k}$.

The Random MAP1 Experiment (part 2)

Claim: The probability that the responder oracle $\Pi^t_{B,A}$ accepts without a matching conversation is at most $T_E(k) \cdot 2^{-k}$

Proof: Suppose at time τ_1 oracle $\Pi^t_{B,A}$ received the flow R_A and responded with $[B.A.R_A, R_B]_g$. To accept, $\Pi^t_{B,A}$ must receive $[A.R_B]_g$ at time $\tau_3 > \tau_1$. The probability that E can compute it is at most 2^{-k}. The initiator must be a $\Pi^s_{A,C}$ oracle. The interaction with E has the form $(\tau_0, \lambda, R'_A), (\tau_2, [C.A.R'_A.R'_B]_g, [A.R'_B]_g)$ for some $\tau_2 > \tau_0$. Except for probability 2^{-k} there is a $\Pi^u_{C,A}$ oracle which output $[C.A.R'_A.R'_B]_g$.

The Random MAP1 Experiment (part 2 cont.)

Proof (cont.): If $(u, C) \neq (t, B)$, the probability that $R'_B = R_B$ is at most $[T_E(k) - 2] \cdot 2^{-k}$ and thus the probability that $[A.R'_B]_g$ leads $\Pi^t_{B,A}$ to accept is at most $[T_E(k) - 2] \cdot 2^{-k}$. Suppose $(u, C) = (t, B)$. It follows that $\tau_0 < \tau_1 < \tau_2 < \tau_3$, $R'_A = R_A$ and $R'_B = R_B$ and we have a matching conversation. The probability that $\Pi^t_{B,A}$ accepts without a matching conversation is at most $T_E(k) \cdot 2^{-k}$.

Conclusion: The probability that there exists an oracle which accepts without a matching conversation is at most $T_E(k)$ times the bound obtained in the claims, thus $T_E(k)^2 \cdot 2^{-k}$

The Real MAP1 Experiment

Claim: Real MAP1 is secure

Proof: Suppose adversary E has a non-negligible probability to succeed in the real MAP1 experiment. We will construct a polynomial time test T which distinguishes random functions from pseudo-random functions. T receives $g : \{0,1\}^{\leq L(k)} \rightarrow \{0,1\}^k$ which is chosen by fliping a coin C. If $C = 1$ let g be a random function, else pick a at random and let $g = f_a$. T's job is to predict C with some advantage. T runs E for MAP1^g. T simulates all oracles $\Pi^s_{i,j}$. If E is successful, T predicts $C = 0$ (g is pseudorandom), else T predicts $C = 1$ (g is random). T's advantage is $\text{Adv}(T) = \frac{1}{2}\text{Adv}(E)$. Thus an efficient attack on real MAP1 leads to a distiguihser of random and pseudorandom functions.

Authenticated Key Exchange

- The intent of an AKE will be to authenticate entities and to distribute a session key. When a player accepts, his private output will be interpreted as the session key.
- We strengthen our adversary E so that he can query a session key $\alpha_{i,j}^s$ of oracle $\Pi_{i,j}^s$
- Initially oracles are unopened, until the adversary asks for the session key.
- An oracle $\Pi_{i,j}^s$ is fresh if it has accepted, is unopened and there is no opened oracle $\Pi_{j,i}^t$ which engaged in a matching conversation with $\Pi_{i,j}^s$

Authenticated Key Exchange Security

- At the end of a secure AKE the adversary should be unable to distinguish a fresh session key from a random element over $\{0,1\}^k$
- Protocol Π is a secure AKE if Π is a secure mutual authentication protocol and in addition it is true that:
  1. In the precense of a benign adversary, oracles $\Pi^s_{i,j}$ and $\Pi^t_{j,i}$ accept with $\alpha^s_{i,j} = \alpha^t_{j,i}$
  2. In the precense of any polynomial time adversary E the advantage of distinguishing a given session key from a random output from $\{0,1\}^k$ should be negligible.
- We can modify MAP1 to a secure AKE:

AKEP1

$$\text{Alice}^{a_1,a_2} \qquad \begin{array}{c} \xrightarrow{\quad A.R_A \quad} \\[2em] \xleftarrow{\quad [B.A.R_A.R_B.\{\alpha\}_{a_2}]_{a_1} \quad} \\[2em] \xrightarrow{\quad [A.R_B]_{a_1} \quad} \end{array} \qquad \text{Bob}^{a_1,a_2}$$

Conclusion

- Bellare and Rogaway provide a framework for proving authentication protocols
- Matching conversations is a useful paradigm for proving protocol security
- MAP1 is a secure mutual authentication protocol and AKEP1 is a secure key exchange protocol
- Proofs rely on the existance of PRFs that are indistinguishable from truly random functions