Exercise for Engineering Secure Software Systems

November 26, 2020: Exercises 2, 3

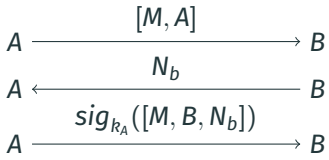
Henning Schnoor

Institut für Informatik, Christian-Albrechts-Universität zu Kiel

Protocol Omitted Last Week

A Secure Protocol: Simple Authentication

protocol



too complicated?

- why three messages?
- why is N_b needed?
- why must B be signed?

Bob's guarantees?

What can Bob be sure of after the protocol has successfully completed?

Omitting the Nonce

protocol

1. $A \xrightarrow{[M, A]} B$
2. $A \xleftarrow{\text{“OK, sign!”}} B$
3. $A \xrightarrow{sig_{k_A}([M, B])} B$

attack a real issue?

- A tenant
- B bank
- M money transfer for rent
- I landlord

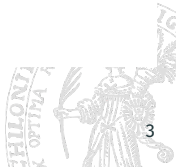
replay attack

1. $A \xrightarrow{[M, A]} B$
2. $A \xleftarrow{\text{“OK, sign!”}} B$
3. $A \xrightarrow{sig_{k_A}([M, B])} B$

1’. $I(A) \xrightarrow{[M, A]} B$

2’. $I(A) \xleftarrow{\text{“OK, sign!””}} B$

3’. $I(A) \xrightarrow{sig_{k_A}([M, B])} B$



Leaving out the Receipient (B)

protocol

1. $A \xrightarrow{[M,A]} B$
2. $A \xleftarrow{N_B} B$
3. $A \xrightarrow{sig_{k_A}(M, N_b)} B$

problem?

- A believes to have sent M to B with authentication
- C believes to have received M from A with authentication

MitM attack

1. $A \xrightarrow{[M,A]} I(B)$
1'. $I(A) \xrightarrow{[M,A]} C$
2'. $I(A) \xleftarrow{N} C$
2. $A \xleftarrow{N} I(B)$
3. $A \xrightarrow{sig_{k_A}([M,N])} I(B)$
3'. $I(A) \xrightarrow{sig_{k_A}([M,N])} C$

Discussion: Tasks for this week

Exercise

Task (DY closure and derivations)

In the lecture, the following lemma was stated (without proof):

If S is a set with $\mathsf{IDs} \cup \{k_a \mid a \in \mathsf{IDs}\} \cup \{\epsilon\} \subseteq S$ and $m \in \mathsf{DY}(S)$, then there is a derivation of m from S: $S = S_0 \to_{L_0} S_1 \to_{L_1} \ldots S_{n-1} \to_{L_{n-1}} S_n$ with $m \in S_n$.

1. Prove the above lemma.
2. State and prove an appropriate converse of the lemma.

Note: As in the lecture, you can assume that both S and m do not contain applications of hash functions, message authentication codes (MACs), or signatures.

Exercise

Task (minimal derivation properties)

In the video lecture on the computation of the Dolev-Yao closure, we proved a lemma characterizing shortest derivations.

1. Can you generalize this result to handle signatures, MACs, and hash functions?
2. Which properties does the modeling of cryptographic primitives have to satisfy for an analog of this result to hold?
3. Can you come up with a modeling of cryptographic primitives where this property does not hold?

Exercise

Task (DY algorithm correctness)

Prove that the algorithm for computing the DY closure (in its decisional variant **DERIVE**) as stated in the lecture is correct and runs in polynomial time. As in the lecture, restrict yourself to terms without applications of hash functions, signatures, or message authentication codes (MACs).

Preview: Tasks for next week

Exercise

Task (Formal Representation of the Woo Lam Protocol)

Study the authentication protocol by Woo and Lam (see slide 17 of the lecture from November 10).

1. Specify the protocol as sequence of receive/send actions, once in the intended execution between Alice and Bob, and once in a form that allows to model the attack introduced in the lecture.
2. Specify the attack on the protocol formally.
3. How can we modify the protocol in order to prevent this attack?

Exercise

Task (Otway Rees Protocol)

Consider the following protocol (Otway-Rees-Protocol):

1. $A \to B \quad [M, A, B, \mathrm{enc}^{s}_{k_{AS}}([N_a, M, A, B])]$
2. $B \to S \quad [M, A, B, \mathrm{enc}^{s}_{k_{AS}}([N_a, M, A, B]), \mathrm{enc}^{s}_{k_{BS}}([N_b, M, A, B])]$
3. $S \to B \quad [M, \mathrm{enc}^{s}_{k_{AS}}([N_a, k_{AB}]), \mathrm{enc}^{s}_{k_{BS}}([N_b, k_{AB}])]$
4. $B \to A \quad [M, \mathrm{enc}^{s}_{k_{AS}}([N_a, k_{AB}])]$
5. $A \to B \quad \mathrm{enc}^{s}_{k_{AB}}(\mathrm{FAIL})$

1. Why are the subterms M, A, and B in the second message sent both encrypted and as plaintext?
2. Why is the nonce N_b encrypted in message 2?
3. Is the protocol secure? (You do not need to give a formal proof of security or insecurity.)

Exercise

Task (Security Modeling Issues: Are we Missing Something?)

In the lecture, we defined security of a protocol as, essentially, unreachability of a state in which the adversary learns the constant FAIL. However, this FAIL-constant obviously does not have a correspondance in a real implementation of a protocol. In particular, the rules releasing the FAIL-constant are removed from the protocol in a real implementation. As a consequence, a potential security proof of a protocol in our formal model treats a different protocol than the protocol running in a real implementation.

Are there cases where this difference results in an insecure protocol that can be proven secure in our formal model? If this is the case, how can we circumvent this issue?