Security II - Security Properties

Stefano Calzavara

Università Ca' Foscari Venezia

April 24, 2020

Introduction

We studied how to model cryptographic protocols in applied pi-calculus

- but how can we formulate security properties?
- secrecy: the attacker should not be able to learn confidential parts of protocol messages
- authentication: a subtle property, which ensures that the sender and the receiver "agree" on the exchanged data and their respective roles

Secrecy

Definition

The process P preserves the secrecy of M iff, for all the opponents O, we have that $P \mid O$ never outputs M on a public channel.

This also covers the case where M is not directly leaked by P, but can be reconstructed by O, because O can then output M on a public channel.

Do the following processes preserve the secrecy of n?

- $(\nu k)\,(\overline{c}\langle \mathrm{senc}(n, k)\rangle)$

Secrecy

Definition

The process P preserves the secrecy of M iff, for all the opponents O, we have that $P \mid O$ never outputs M on a public channel.

This also covers the case where M is not directly leaked by P, but can be reconstructed by O, because O can then output M on a public channel.

Do the following processes preserve the secrecy of n?

- $(\nu k)\,(\overline{c}\langle \mathrm{senc}(n, k)\rangle)$
- $(\nu n)\,(\nu k)\,(\overline{c}\langle \mathrm{senc}(n, k)\rangle)$

Secrecy

Definition

The process P preserves the secrecy of M iff, for all the opponents O, we have that $P \mid O$ never outputs M on a public channel.

This also covers the case where M is not directly leaked by P, but can be reconstructed by O, because O can then output M on a public channel.

Do the following processes preserve the secrecy of n?

- $(\nu k)\,(\overline{c}\langle \mathrm{senc}(n, k)\rangle)$
- $(\nu n)\,(\nu k)\,(\overline{c}\langle \mathrm{senc}(n, k)\rangle)$
- $(\nu n)\,(\nu k)\,(\overline{c}\langle \mathrm{senc}(n, k\rangle.\overline{c}\langle k\rangle)$

Violating Secrecy

Pick the process $P \triangleq (\nu n)\,(\nu k)\,(\overline{c}\langle \mathrm{senc}(n,k)\rangle.\overline{c}\langle k\rangle)$

The secrecy of n is violated by the following opponent:

$$O \triangleq c(x).c(y).\mathrm{let}\ z = \mathrm{sdec}(x,y)\ \mathrm{in}\ \overline{a}\langle z\rangle$$

We can show that:

$$\begin{array}{lll} P \mid O & \rightarrow & (\nu n)\,(\nu k)\,(\overline{c}\langle k\rangle \mid c(y).\mathrm{let}\ z = \mathrm{sdec}(\mathrm{senc}(n,k),y)\ \mathrm{in}\ \overline{a}\langle z\rangle) \\ & \rightarrow & (\nu n)\,(\nu k)\,\mathrm{let}\ z = \mathrm{sdec}(\mathrm{senc}(n,k),k)\ \mathrm{in}\ \overline{a}\langle z\rangle \\ & \rightarrow & (\nu n)\,(\nu k)\,\overline{a}\langle n\rangle \end{array}$$

Strong Secrecy

Our simple definition of secrecy has two main problems:

1. no implicit flows: we have discussed that secrets can be leaked bit by bit, we can't capture that only part of the secret is revealed
2. limited expressiveness: what if the secret is a public value, like in the case of e-voting protocols?

There are also stronger definitions of secrecy in the literature, based on the notion of observational equivalence.

Example

A protocol run where Alice votes for Bob is observationally equivalent to a protocol run where Alice votes for Charlie.

Authentication

Authentication is harder to formulate than secrecy

- non-injective agreement: the parties must agree on their respective identities, their role in the protocol and the content of the message
- injective agreement: same as above, but the recipient must also be able to verify the freshness of the message

Example

Assume that A sends a payment order M to B. Non-injective agreement requires that B authenticates A as the sender of M. Injective agreement also ensures that B cannot accept M multiple times (no replay attacks).

Correspondence Assertions

We decorate the protocol code with events, also called correspondence assertions in traditional literature

- begin(A, B, M): A sends to B the message M
- end(A, B, M): B accepts from A the message M

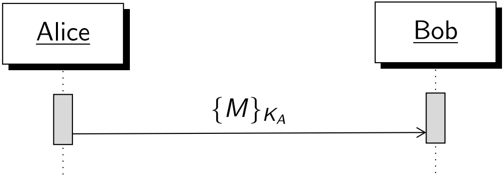
We assume that the attacker's code cannot contain end() events

Definition

The process P satisfies non-injective agreement iff, for all the opponents O and runs of $P \mid O$, each end(A, B, M) is preceded by a begin(A, B, M).

We require a distinct begin(A, B, M) for injective agreement!

Example: Injective vs Non-Injective Agreement



$$
\begin{array}{lcl}
A & \triangleq & \mathrm{begin}(a, b, M).\overline{b}\langle \mathrm{sign}(M, K_A)\rangle \\
B & \triangleq & !b(x).\mathrm{let}\ y = \mathrm{ver}(x, \mathrm{pk}(K_A))\ \mathrm{in}\ \mathrm{end}(a, b, y) \\
S & \triangleq & (\nu K_A)\,(A \mid B)
\end{array}
$$

This protocol satisfies non-injective agreement, but violates injective agreement: $O \triangleq b(x).\overline{b}\langle x\rangle.\overline{b}\langle x\rangle$

Challenge - Response Handshakes

We now study three different challenge-response schemes:

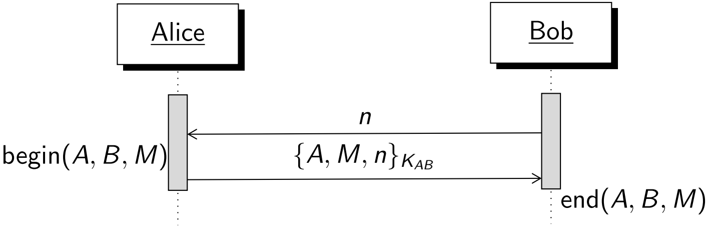
- plain-cipher (PC): challenge in clear, response encrypted
- cipher-plain (CP): challenge encrypted, response in clear
- cipher-cipher (CC): both challenge and response encrypted

Common idea: prove your identity by encrypting/decrypting

However, these schemes enjoy different security properties!

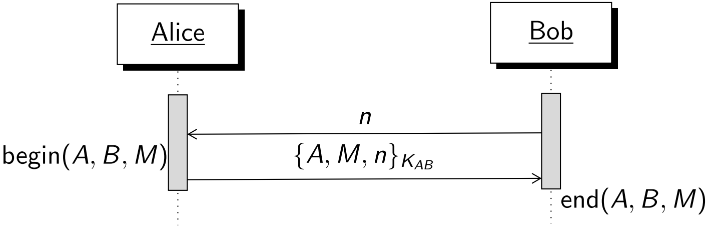
PC Handshake - Symmetric Key

Which authentication property is satisfied by the protocol?



PC Handshake - Symmetric Key

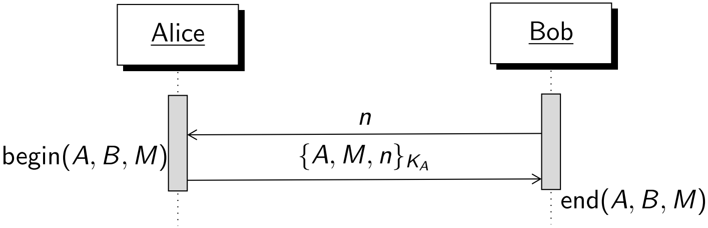
Which authentication property is satisfied by the protocol?



Answer: injective agreement begin$(A, B, M), \ldots,$ end(A, B, M)

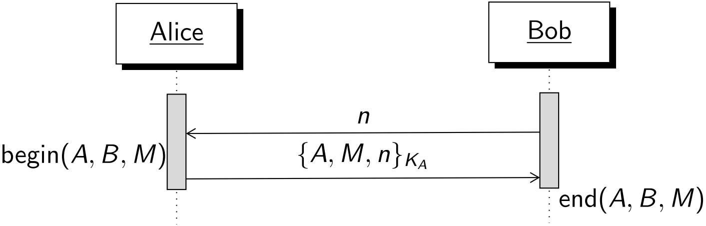
PC Handshake - Asymmetric Key

Which authentication property is satisfied by the protocol?



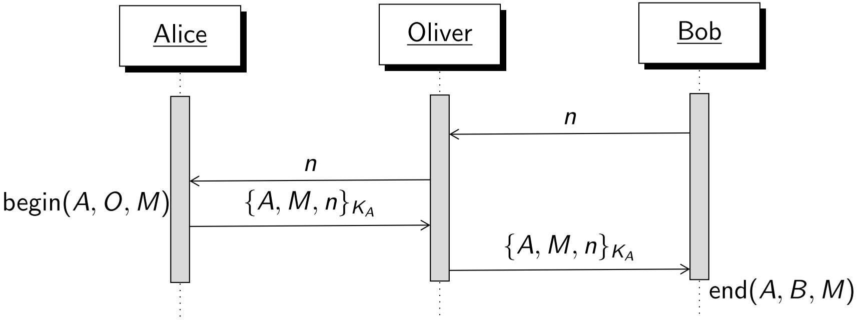
PC Handshake - Asymmetric Key

Which authentication property is satisfied by the protocol?



Answer: none! The second message is the same for Bob and Oliver!

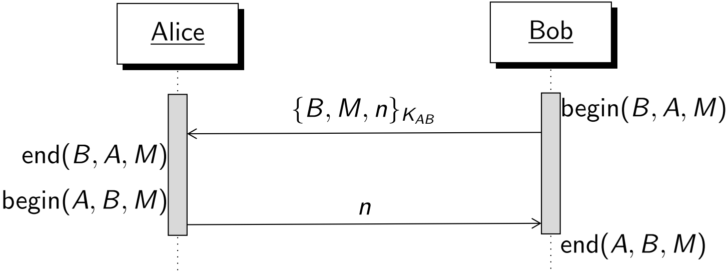
Breaking Authentication



Fix: in the second message replace the identity of the sender A with the identity of the recipient O

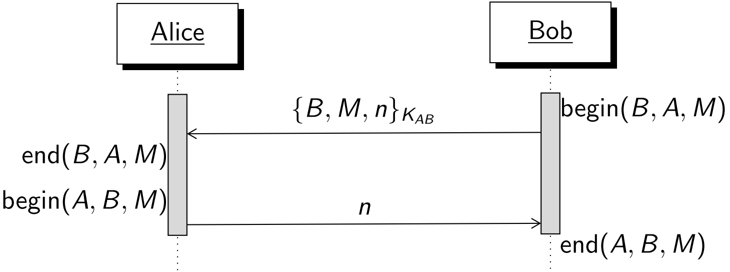
CP Handshake - Symmetric Key

Which authentication properties are satisfied by the protocol?



CP Handshake - Symmetric Key

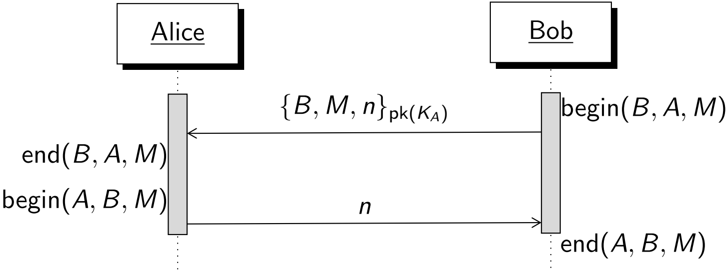
Which authentication properties are satisfied by the protocol?



Answer: non-injective agreement begin$(B, A, M), \ldots,$ end(B, A, M) and injective agreement begin$(A, B, M), \ldots,$ end(A, B, M)

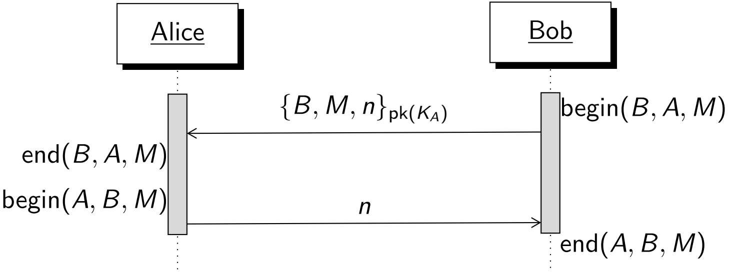
CP Handshake - Asymmetric Key

Which authentication properties are satisfied by the protocol?



CP Handshake - Asymmetric Key

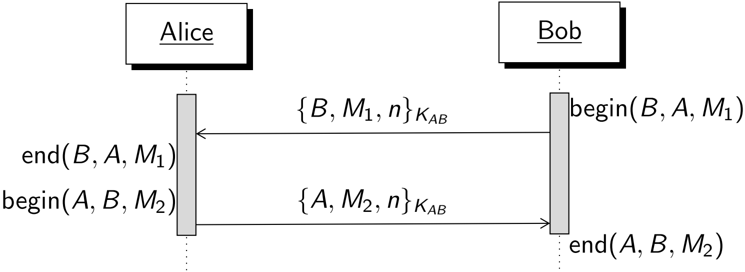
Which authentication properties are satisfied by the protocol?



Answer: just injective agreement begin$(A, B, M), \ldots,$ end(A, B, M), since the challenge might come from Oliver

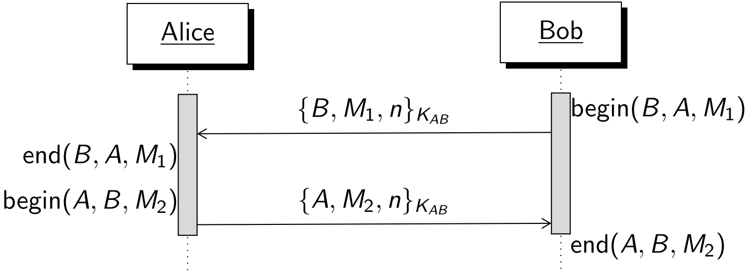
CC Handshake - Symmetric Key

Which authentication properties are satisfied by the protocol?



CC Handshake - Symmetric Key

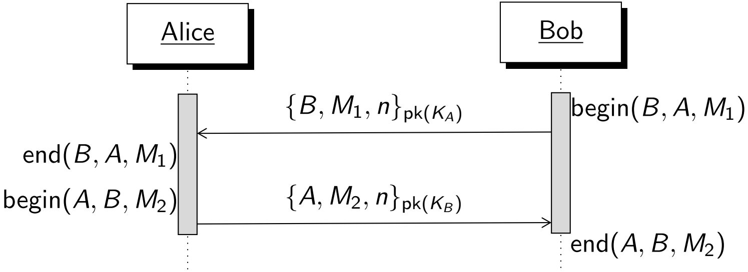
Which authentication properties are satisfied by the protocol?



Answer: non-injective agreement begin$(B, A, M_1), \ldots,$ end(B, A, M_1) and injective agreement begin$(A, B, M_2), \ldots,$ end(A, B, M_2)

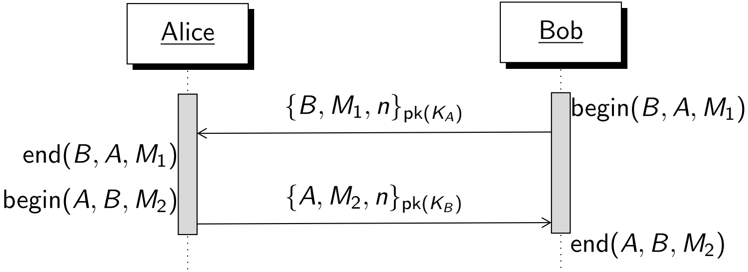
CC Handshake - Asymmetric Key

Which authentication properties are satisfied by the protocol?



CC Handshake - Asymmetric Key

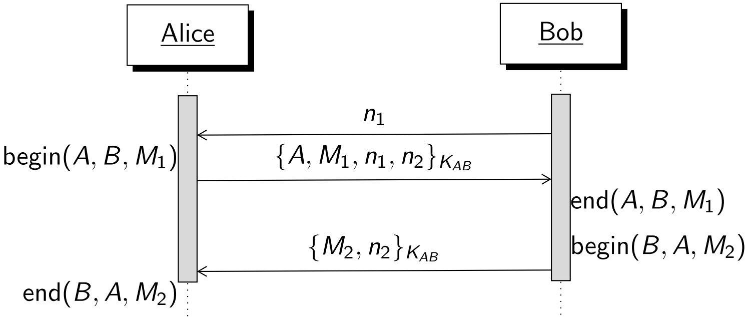
Which authentication properties are satisfied by the protocol?



Answer: just injective agreement begin$(A, B, M_2), \ldots,$ end(A, B, M_2), since the challenge might come from Oliver

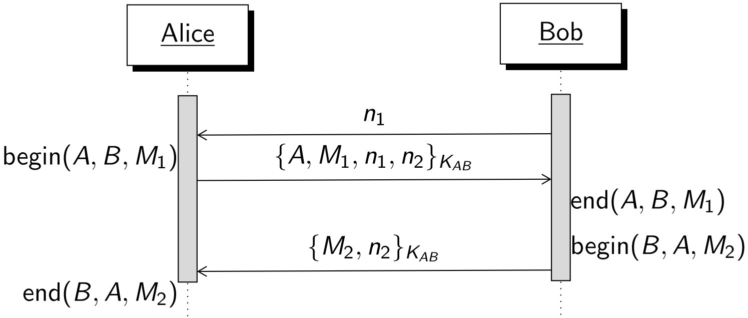
Mutual Authentication - Symmetric Key

Which authentication properties are satisfied by the protocol?



Mutual Authentication - Symmetric Key

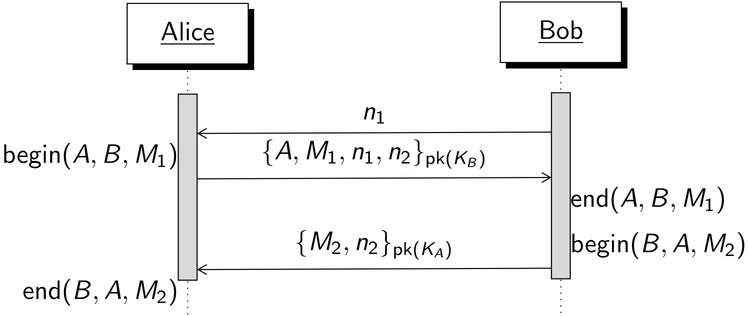
Which authentication properties are satisfied by the protocol?



Answer: injective agreement begin$(B, A, M_1), \ldots,$ end(B, A, M_1) and injective agreement begin$(A, B, M_2), \ldots,$ end(A, B, M_2)

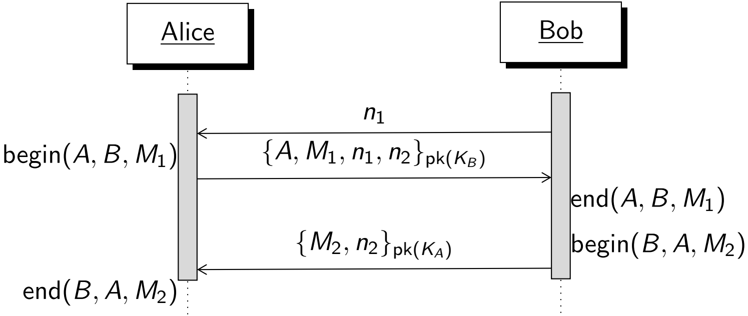
Mutual Authentication - Asymmetric Key

Which authentication properties are satisfied by the protocol?



Mutual Authentication - Asymmetric Key

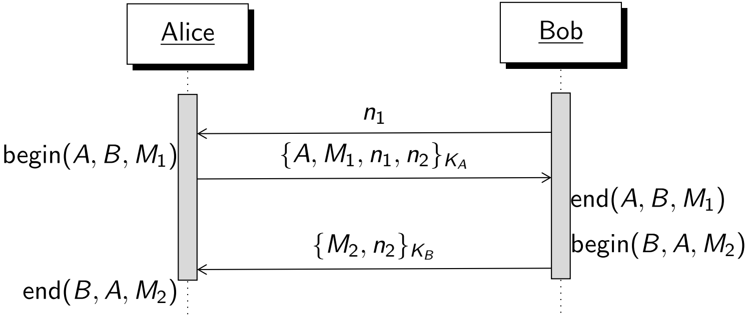
Which authentication properties are satisfied by the protocol?



Answer: just injective agreement begin$(B, A, M_2), \ldots,$ end(B, A, M_2), since the first response might come from Oliver

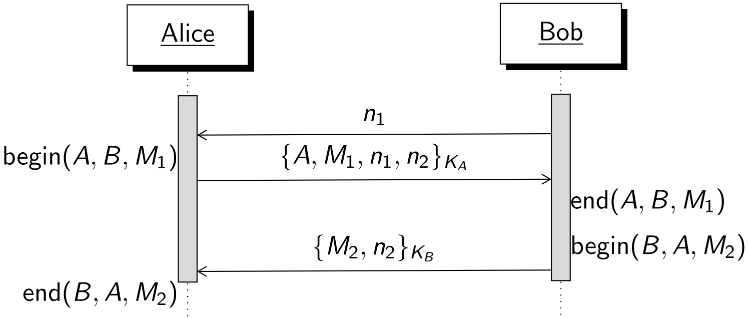
Mutual Authentication - Asymmetric Key (Revised)

Which authentication properties are satisfied by the protocol?



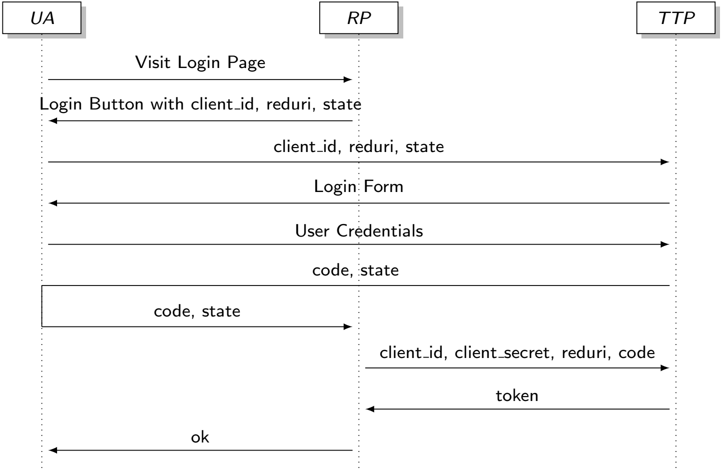
Mutual Authentication - Asymmetric Key (Revised)

Which authentication properties are satisfied by the protocol?



Answer: injective agreement begin$(B, A, M_1), \ldots,$ end(B, A, M_1) and injective agreement begin$(A, B, M_2), \ldots,$ end(A, B, M_2)

OAuth 2.0 (Explicit Mode)



What Now?

We have shown how to formalize security properties of protocols

- showing that a property is false is "easy": counter-example
- showing that a property is true is more complicated, since most useful security properties are undecidable
- very easy for humans to make mistakes, think about previous cases!
- luckily, we have verification tools for secrecy and authentication properties of cryptographic protocols
- next lecture: ProVerif, a state-of-the-art verification tool