Engineering Secure Software Systems

November 10, 2020: Crypto Protocols: Example and Formal Model

Henning Schnoor

Institut für Informatik, Christian-Albrechts-Universität zu Kiel

Part I: Crypto Protocols

Overview

Part I: Crypto Protocols

- Foundations
- Cryptography
- An Example and an Attack

More Examples

Formal Protocol Model

- Motivation and Requirements
- Messages: Formal Terms
- Message Construction: Dolev-Yao Closure

Summary: Public and Secret Keys

Priv.-Doz. Dr. Henning Schnoor

wiss. Mitarbeiter

Christian-Albrechts-Platz 4, R.1215 (CAP 4)
Phone: +49 431 880-4467
Telefax: +49 431 880-7617
henning.schnoor@email.uni-kiel.de

PGP key: henning-schnoor-pgp-key.asc

with public key k_{HS}, you can …

- send encrypted emails to me

  $$x \rightarrow \mathrm{enc}^{a}_{k_{HS}}(x)$$

- verify whether I signed a message

  $$\text{test } \mathrm{sig}_{\hat{k}_{HS}}(x)$$

with secret key $\hat{k}_{HS}$, I can …

- decrypt mails encrypted with my public key

  $$\mathrm{enc}^{a}_{k_{HS}}(x) \rightarrow x$$

- sign messages that will successfully verify against my public key

  $$x \rightarrow \mathrm{sig}_{\hat{k}_{HS}}(x)$$

Assumptions too strong?

impossible with adversary-controlled network

- reply after $\leq t$ seconds
- reliable emergency call system
- delivery guarantee for messages
- …

in general

"liveness" properties cannot be guaranteed when network is completely unreliable

protocol design futile

If all others "maximally dishonest:" communication not reasonable

consequence: assumptions (depend on scenario), examples:

- at least Alice and Bob are honest
- existence of a trusted third party (TTP)
- Alice and Bob share secret key
- availability of PKI (public-key infrastructure)
- …

Hopeless Situations?

cryptography can only help you so far …

political electronic elections in Germany

Der Zweite Senat hat entschieden, dass der Einsatz elektronischer Wahlgeräte voraussetzt, dass die ***wesentlichen Schritte der Wahlhandlung*** *und der Ergebnisermittlung vom Bürger* ***zuverlässig und ohne besondere Sachkenntnis überprüft werden können****. Dies ergibt sich aus dem Grundsatz der Öffentlichkeit der Wahl (Art. 38 in Verbindung mit Art. 20 Abs. 1 und Abs. 2 GG), der gebietet, dass alle wesentlichen Schritte der Wahl öffentlicher Überprüfbarkeit unterliegen, soweit nicht andere verfassungsrechtliche Belange eine Ausnahme rechtfertigen.*

Example: Authentication

goal

Bob expects “authenticated” message from Alice

problems

- attacker can always send message in Alice’s name!
- Bob needs way to check authenticity of message

cannot require that message arrives (liveness)

require only: **if** Bob “accepts,” **then** message is from Alice.

need infrastructure: Alice “can do” something the attacker “can’t”

- Alice has private key, Bob knows public key
- Alice and Bob share private secret
- Alice can authenticate herself using a certificate
- ...

A Secure Protocol: Simple Authentication

protocol

$$A \xrightarrow{[M, A]} B$$
$$A \xleftarrow{N_b} B$$
$$A \xrightarrow{sig_{k_A}([M, B, N_b])} B$$

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?

Exercise

Task (simple example protocol)

We consider the following simple authentication protocol:

- Alice sends a message M to Bob, together with her name A,
- Bob answers with a Nonce N_b,
- Alice answers with the term $\mathsf{sig}_{k_A}([M, B, N_B])$.

Please answer the following questions:

1. What are the security properties guaranteed by the protocol?
2. What is the purpose of the nonce N_B? What happens if we omit it?
3. What happens if the B is removed from Alice's last message?

Overview

An Example: Authentication

goal

authentication: Bob wants to be sure that he is talking to Alice

infrastructure

PKI: Alice and Bob have public keys k_A and k_B

authentication protocol

$A \rightarrow B$	A	Hi, I'm Alice!
$B \rightarrow A$	$\mathsf{enc}^{\mathsf{a}}_{k_A}(N_B)$	Prove this!
$A \rightarrow B$	$\mathsf{enc}^{\mathsf{a}}_{k_B}(N_B)$	I know Alice's secret key $\hat{k}_A$!

secure protocol?

- can Bob be sure he is talking to Alice when he receives $\mathbf{enc}^{\mathsf{a}}_{k_B}(N_B)$?
- obvious "bug" in protocol?

goal

authentication and key exchange

protocol

$A \to B \quad \mathrm{enc}^{a}_{k_B}(A, N_A)$

$B \to A \quad \mathrm{enc}^{a}_{k_A}(N_A, N_B)$

$A \to B \quad \mathrm{enc}^{a}_{k_B}(N_B)$

then $N_A \oplus N_B$ secure session key for A and B

really?

- protocol: 1978 [NS78]
- attack found: 1995 [Low96]

recall

Needham: three-line programs!

Attack on Needham-Schroeder

protocol

1. $A \to \cancel{B}\ C$ $\quad \mathrm{enc}^{a}_{\cancel{k_B}\ k_C}(A, N_A)$
2. $\cancel{B}\ C \to A$ $\quad \mathrm{enc}^{a}_{k_A}(N_A, \cancel{N_B}\ N_C)$
3. $A \to \cancel{B}\ C$ $\quad \mathrm{enc}^{a}_{\cancel{k_B}\ k_C}(\cancel{N_B}\ N_C)$

situation

- Alice starts protocol as initiator with C (attacker)
- Bob starts protocol as responder with Alice
- adjust protocol for this situation

attack (Charlie controlled by $\mathcal{A}$)

1. $A \xrightarrow{\mathrm{enc}^{a}_{k_C}(A, N_A)} C$

1'. $C \xrightarrow{\mathrm{enc}^{a}_{k_B}(A, N_A)} B$

2'. $C \xleftarrow{\mathrm{enc}^{a}_{k_A}(N_A, N_B)} B$

2. $A \xleftarrow{\mathrm{enc}^{a}_{k_A}(N_A, N_B)} C$

3. $A \xrightarrow{\mathrm{enc}^{a}_{k_C}(N_B)} C$

3'. $C \xrightarrow{\mathrm{enc}^{a}_{k_B}(N_B)} B$

consequence

- who is attacked?
- Bob “thinks” only Alice knows N_A and N_B
- C knows N_A and N_B
- what about Alice’s point of view?
- suggestions to fix protocol?

The Needham-Schroeder-Lowe Protocol

protocol

$A \to B \quad \mathrm{enc}^{\mathrm{a}}_{k_B}(A, N_A)$

$B \to A \quad \mathrm{enc}^{\mathrm{a}}_{k_A}(N_A, N_B, B)$

$A \to B \quad \mathrm{enc}^{\mathrm{a}}_{k_B}(N_B)$

then $N_A \oplus N_B$ secure session key for A and B

intuition

- attack "mixes" messages from different protocol sessions
- consequence: B "talks to" C instead of A
- change: A realizes that message does not come from C

Attack on Needham-Schroeder-Lowe?

protocol

1. $A \to B \quad \mathrm{enc}^{a}_{k_B}(A, N_A)$
2. $B \to A \quad \mathrm{enc}^{a}_{k_A}(N_A, N_b, B)$
3. $A \to B \quad \mathrm{enc}^{a}_{k_B}(N_b)$

consequence

- Alice "talks to C," receives message with "sender" B
- Alice aborts
- **good practice**: sender and receiver in messages

attack attempt

1. $A \xrightarrow{\mathrm{enc}^{a}_{k_C}(A, N_A)} C$

1'. $C \xrightarrow{\mathrm{enc}^{a}_{k_B}(A, N_A)} B$

2'. $C \xleftarrow{\mathrm{enc}^{a}_{k_A}(N_A, N_B, B)} B$

2. $A \xleftarrow{\mathrm{enc}^{a}_{k_A}(N_A, N_B, B)} C$

3. abort

Task (Fixing Broken Authentication Protocols)

Consider the two authentication protocols presented in the exercise class:

a)

1. $A \to B \quad (A, \mathrm{enc}^{\mathrm{a}}_{k_B}(N_A))$
2. $B \to A \quad (B, \mathrm{enc}^{\mathrm{a}}_{k_A}(N_A))$

b)

1. $A \to B \quad (\mathrm{enc}^{\mathrm{a}}_{k_B}(N_A), \mathrm{enc}^{\mathrm{a}}_{k_B}(A))$
2. $B \to A \quad (\mathrm{enc}^{\mathrm{a}}_{k_A}(N_A, N_B), \mathrm{enc}^{\mathrm{a}}_{k_A}(B))$

Both of these protocols can be attacked with a similar attack as the Needham-Schroeder protocol or the example protocol we covered in the first exercise class. Suggest changes to the protocols that address these problems, and argue why you think your revised versions of the protocols are secure. Be as specific as possible in what “secure” means in this case.

Overview

Example: Woo-Lam Authentication Protocol

prerequisites

k_{AS}, k_{BS}: symmetric keys shared between Alice (Bob) and Server

protocol

1.	$A \to B$	A
2.	$B \to A$	N_B
3.	$A \to B$	$\mathrm{enc}^s_{k_{AS}}(N_B)$
4.	$B \to S$	$\mathrm{enc}^s_{k_{BS}}\left([A, \mathrm{enc}^s_{k_{AS}}(N_B)]\right)$
5.	$S \to B$	$\mathrm{enc}^s_{k_{BS}}(N_B)$

idea

- only Alice can encrypt N_B with k_{AS}
- server can check correctness

issues?

- server is "decryption oracle"
- Alice does not "know" that she "talks to Bob"

reference

Thomas Y. C. Woo and Simon S. Lam. "Authentication for Distributed Systems". In: Computer 25.1 (Jan. 1992), pp. 39–52. ISSN: 0018-9162. DOI: `10.1109/2.108052`. URL: `http://dx.doi.org/10.1109/2.108052`

Woo-Lam Protocol is insecure

protocol

1. $A \to B \quad A$
2. $B \to A \quad N_B$
3. $A \to B \quad \mathrm{enc}^s_{k_{AS}}(N_B)$
4. $B \to S \quad \mathrm{enc}^s_{k_{BS}}\left([A, \mathrm{enc}^s_{k_{AS}}(N_B)]\right)$
5. $S \to B \quad \mathrm{enc}^s_{k_{BS}}(N_B)$

analysis

- **trash**: result of decrypting $\mathbf{enc}^s_{k_{CS}}(N_B)$ with K_{AS}
- B believes A participated in protocol run
- assumptions?

attack: C controlled by $\mathcal{A}$, B honest

1	$C \xrightarrow{A} B$
1'	$C \xrightarrow{C} B$
2	$C \xleftarrow{N_B} B$
2'	$C \xleftarrow{N'_B} B$
3	$C \xrightarrow{\mathrm{enc}^s_{k_{CS}}(N_B)} B$
3'	$C \xrightarrow{\mathrm{enc}^s_{k_{CS}}(N_B)} B$
4	$B \xrightarrow{\mathrm{enc}^s_{k_{BS}}\left([A, \mathrm{enc}^s_{k_{CS}}(N_B)]\right)} S$
4'	$B \xrightarrow{\mathrm{enc}^s_{k_{BS}}\left([C, \mathrm{enc}^s_{k_{CS}}(N_B)]\right)} S$
5	$B \xleftarrow{\mathrm{enc}^s_{k_{BS}}(\mathrm{trash})} S$
5'	$B \xleftarrow{\mathrm{enc}^s_{k_{BS}}(N_B)} S$

Overview

Overview

Finding Attacks

seen up to now: manual analysis

issues found by

- construction of message sequence: reachability of “bad state”
- argument that the attacker “knows” each message she sends: attacker gains sufficient knowledge to trigger required events

goal: automatic analysis

want algorithm that comes up with attack, or “proves” that there is no attack

required

formal model in which we can express protocols, security, and attacks

Model Requirement: Express Needham Schroeder

minimal requirement for formal model

must be able to formalize Needham-Schroeder(-Lowe) protocol, attack, security.

attacker actions

1. C “talks to Bob in Alice’s name” steps $1', 2', 3'$
2. C makes Alice accept N_B instead of N_C step 2
3. C exchanges data between sessions all steps
4. C lets Alice and Bob wait steps $1', 2', 2, 3$

consequences for model

1. untrusted message delivery
2. Alice’s protocol specification must not mention N_C
3. attacker can send arbitrary terms, limited only by cryptography
4. attacker controls scheduling

Model Requirements: Generalizing from Needham-Schroeder Example

untrusted message delivery

messages delivered by network without meta-information

Alice's protocol specification must not mention N_C

expected terms cannot be hard-coded, model steps as receive/send-rules with variables instead

attacker can send arbitrary terms, limited only by cryptography

adversary controls network, uses "message construction" rules precisely defined using so-called Dolev-Yao closure

attacker controls scheduling

scheduling (execution order) explicitly done by adversary

Roadmap: Formal Model

features

1. untrusted message delivery
2. Alice's protocol specification must not mention N_C
3. attacker can send arbitrary terms, limited only by cryptography
4. attacker controls scheduling

components of formal model

messages	formal terms
message construction, delivery, parsing	Dolev-Yao closure, receive/send actions, substitutions, matching
protocol specifications	protocol instance, protocol
sessions, scheduling	execution order
protocol security: no combination of adversary actions breaks protocol goal	protocol runs, (successful) attacks

Overview

Messages: Abstracting from Bitstrings

implemented protocol

- messages are bitstrings
- constructed by crypto algorithms
- attacker: arbitrary probabilistic polynomial-time algorithm

formal model

- messages are terms
- algorithms represented by function symbols
- attacker: nondeterministic choice of messages

why?

advantages of term model?

definition: terms

$\mathcal{T}$: smallest set with

- $\{\epsilon\} \cup \mathcal{C} \cup \mathcal{V} \cup \textbf{IDs} \subseteq \mathcal{T}$, — empty message, constants, variables, names
- for all $i \in \mathbb{N}$, all $a \in \textbf{IDs}$: $N_i, k_a, \hat{k}_a \in \mathcal{T}$, — random values, keys
- if $t_1, t_2 \in \mathcal{T}$, then $[t_1, t_2] \in \mathcal{T}$, — pairs/sequences
- if $t, t_k \in \mathcal{T}$, then $\textbf{enc}^{s}_{t_k}(t) \in \mathcal{T}$, — symm. encryption
- if $t \in \mathcal{T}$, $a \in \textbf{IDs}$, then $\textbf{enc}^{a}_{k_a}(t) \in \mathcal{T}$, — asymm. encryption
- if $t, t_k \in \mathcal{T}$, then $\textbf{MAC}_{t_k}(t) \in \mathcal{T}$, — symm. signature (MAC)
- if $t \in \mathcal{T}$, $a \in \textbf{IDs}$, then $sig_{k_a}(t) \in \mathcal{T}$, — asymm. signature
- if $t \in \mathcal{T}$, then $\textbf{hash}(t) \in \mathcal{T}$. — hash function

messages

term without variable: **ground term**, **message**.

$$
\begin{tikzcd}
 & & & & N_A & & \\
 & & \mathrm{sig}_{k_A}(.) \arrow[r] & {[[,],]} \arrow[ur] \arrow[r] \arrow[dr] & N_B & & \\
 & & & & A & & \\
 & & & k_{AS} & & & \\
 & & & & N_B & & N_A \\
\mathrm{enc}^{a}_{k_B}(.) \arrow[r] & {[[,],]} \arrow[uuuur] \arrow[r] \arrow[dddddr] & \mathrm{enc}^{s}_{.}(.) \arrow[uur] \arrow[r] & {[.]} \arrow[ur] \arrow[dr] & & & \\
 & & & & \mathrm{sig}_{k_A}(.) \arrow[r] & {[[,],]} \arrow[uur] \arrow[r] \arrow[dr] & N_B \\
 & & & & & & A \\
 & & & k_{BS} & & & \\
 & & & & N_B & & N_A \\
 & & \mathrm{enc}^{s}_{.}(.) \arrow[uur] \arrow[r] & {[.]} \arrow[ur] \arrow[dr] & & & \\
 & & & & \mathrm{sig}_{k_A}(.) \arrow[r] & {[[,],]} \arrow[uur] \arrow[r] \arrow[dr] & N_B \\
 & & & & & & A
\end{tikzcd}
$$

remarks

- converting to term representation is straight-forward
- optimization possibilities?

definition: subterms

term $t \in \mathcal{T}$, then $\mathsf{Sub}\,(\mathsf{t})$ defined inductively:

- $\mathsf{Sub}\,(\mathsf{t}) = \{t\}$ if t atomic, i.e., $t \in \{\epsilon\} \cup \mathcal{C} \cup \mathsf{IDs} \cup \left\{N_i, k_a, \hat{k}_a \mid i \in \mathbb{N}, a \in \mathsf{IDs}\right\}$
- $\mathsf{Sub}\,([\mathsf{t}_1, \mathsf{t}_2]) = \{[t_1, t_2], t_1, t_2\} \cup \mathsf{Sub}\,(\mathsf{t}_1) \cup \mathsf{Sub}\,(\mathsf{t}_2)$,
- $\mathsf{Sub}\,\left(\mathsf{enc}^{\mathsf{s}}_{\mathsf{t_k}}\,(\mathsf{t})\right) = \left\{\mathsf{enc}^{\mathsf{s}}_{t_k}\,(t)\,, t_k, t\right\} \cup \mathsf{Sub}\,(\mathsf{t}) \cup \mathsf{Sub}\,(\mathsf{t_k})$,
- $\mathsf{Sub}\,\left(\mathsf{enc}^{\mathsf{a}}_{\mathsf{k_a}}\,(\mathsf{t})\right) = \left\{\mathsf{enc}^{\mathsf{s}}_{k_a}\,(t)\,, k_a, t\right\} \cup \mathsf{Sub}\,(\mathsf{t})$,
- $\mathsf{Sub}\,\left(\mathsf{sig}_{\mathsf{k_a}}\,(\mathsf{t})\right) = \left\{\mathsf{sig}_{k_a}\,(t)\,, k_a, t\right\} \cup \mathsf{Sub}\,(\mathsf{t})$,
- $\mathsf{Sub}\,(\mathsf{MAC}_{\mathsf{t_k}}(\mathsf{t})) = \{\mathsf{MAC}_{t_k}(t), t_k, t\} \cup \mathsf{Sub}\,(\mathsf{t})$,
- $\mathsf{Sub}\,(\mathsf{hash}\,(\mathsf{t})) = \{\mathsf{hash}\,(t)\} \cup \mathsf{Sub}\,(\mathsf{t})$.

for $S \subseteq \mathcal{T}$: $\mathsf{Sub}\,(\mathsf{S}) = \cup_{t \in S} \mathsf{Sub}\,(\mathsf{t})$.

Model Requirement: Express Cryptographic Limitations

situation in protocol run: $\mathcal{A}$ knows messages in set S

- own keys
- keys of dishonest parties
- "common knowledge" terms
  $\mathtt{init}, \mathtt{request}, \ldots$
- messages sent by participants so far
- ...

cryptographic operations to cover

- asymmetric encryption — always possible
- symmetric encryption — only with key
- decryption (both cases) — only with key
- signature / MAC — only with key
- apply hash function — always possible
- ...

question

which messages can $\mathcal{A}$ send?

formally

define set **DY** (S) of messages that $\mathcal{A}$ can **derive** from S

Key Concept: Dolev-Yao Closure

reference

Danny Dolev and Andrew Chi-Chih Yao. "On the security of public key protocols". In: IEEE Transactions on Information Theory 29.2 (1983), pp. 198–207

simple attacker modeling

- standard model, many extensions
- consider primitives in isolation
- only derivations, no indistinguishability (see later)
- actual cryptography abstracted away

too simple?

- assume "perfect cryptography"
- practice: do RSA, AES, ElGamal satisfy this?
- abstraction step must be justified!

abstraction soundness

nontrivial topic, subtle issues — ~~(possibly) later in the lecture!~~

Definition: Dolev-Yao Closure

intuition

- DY closure contains everything we cannot stop the adversary from knowing
- and nothing else!
- represents *optimistic* view of cryptography

$S \subseteq \mathcal{T}$, then $\mathrm{DY}(S)$ is the smallest set $D \subseteq \mathcal{T}$ with

- $S \cup \{\epsilon\} \cup \mathbf{IDs} \subseteq D$,
- $t_1, t_2 \in D$ iff $[t_1, t_2] \in D$,
- if $t \in D$ and $a \in \mathbf{IDs}$, then $\mathbf{enc}^{\mathrm{a}}_{k_a}(t) \in D$,
- if $t, t_k \in D$, then $\mathbf{enc}^{\mathrm{s}}_{t_k}(t), \mathbf{MAC}_{t_k}(t) \in D$,
- if $t \in D$ and $\hat{k}_a \in D$, then $sig_{k_a}(t) \in D$,
- if $\mathbf{enc}^{\mathrm{s}}_{t_k}(t) \in D$ and $t_k \in D$, then $t \in D$,
- if $\mathbf{enc}^{\mathrm{a}}_{k_a}(t) \in D$ and $\hat{k}_a \in D$, then $t \in D$,
- if $\mathbf{sig}_{k_a}(t) \in D$, then $t \in D$,
- if $\mathbf{MAC}_{k_i}(t) \in D$, then $t \in D$,
- if $t \in D$, then $\mathbf{hash}(t) \in D$.

note

model allows composed keys for symmetric cryptosystems

Next Session: Review Questions

Next Session

review questions

- we will start the session with discussing review questions
- 5-15 minutes, depending on
  - time (I will roughly follow last year's schedule)
  - participation

your preparation

- review lecture notes up to today
- try to answer review questions marked "during semester"

your participation

- to have a nice discussion: activate cameras!
- come with follow-up questions or ideas for answers!
- present in class orally or via screen-sharing

before we go

any questions?

Thanks!

"See you" next time!

References

References i

Danny Dolev and Andrew Chi-Chih Yao. "On the security of public key protocols". In: IEEE Transactions on Information Theory 29.2 (1983), pp. 198–207.

Gavin Lowe. "Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR". In: TACAS. Ed. by Tiziana Margaria and Bernhard Steffen. Vol. 1055. Lecture Notes in Computer Science. Springer, 1996, pp. 147–166. ISBN: 3-540-61042-1.

Roger M. Needham and Michael D. Schroeder. "Using Encryption for Authentication in Large Networks of Computers". In: Communications of the ACM 21.12 (1978), pp. 993–999.

Thomas Y. C. Woo and Simon S. Lam. "Authentication for Distributed Systems". In: Computer 25.1 (Jan. 1992), pp. 39–52. ISSN: 0018-9162. DOI: 10.1109/2.108052. URL: http://dx.doi.org/10.1109/2.108052.