Exercise for Engineering Secure Software Systems

November 12, 2020: Warm-Up Examples, WhatsApp Task

Henning Schnoor

Institut für Informatik, Christian-Albrechts-Universität zu Kiel

Exercise Classes

class time

- help understanding material
- some background, examples
- **discussion** of tasks, review questions
- let me know what **you** want to discuss (directly/mail/vote)

tasks

- not many **run algorithm** exercises
- **examples**: apply formalisms, algorithms
- **modeling**: alternatives to lecture modeling, helps to understand modeling choices
- **proofs**: understand concepts and why algorithms work
- **practical**: use real tool to analyze protocols

note

You do not “need” to solve all the tasks, pick the ones you learn from. Solutions to most tasks will be published (remind me if I forget).

Exercise Group Organization

still looking for a partner?

- here and now?
- `https://wetalk.informatik.uni-kiel.de/`

re-organize until handing-in of second exercise

- adjust members in GitLab projects (no need to rename project, do not change URL)
- ensure at most two students in each project
- disinvite hs and “students”-group from unused repositories
- ensure “students”-group or hs has Maintainer-access to each used project

Running Example: Needham Schroeder Protocol

Attack on Needham-Schroeder

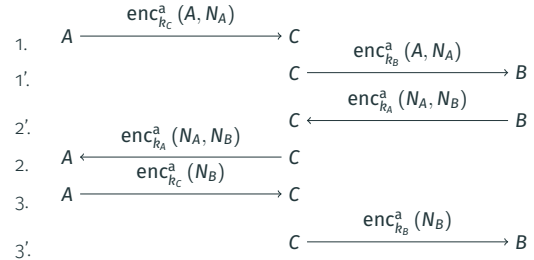
protocol

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

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 $\mathcal{A}$)



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?

Discussion: Tasks for this week

Exercise

Task (exercise git project)

Create a git project together with your exercise partner at `https://git.informatik.uni-kiel.de` using the naming scheme `LL-SEM-Lastname1-Lastname2` and add Henning Schnoor (username hs) as a **Maintainer** to your project. Usually, you should have an account from your Bachelor's studies. If you did not obtain your Bachelor in Kiel or do not have such an account for some other reason, see `http://www.inf.uni-kiel.de/de/service/technik-service/accounts` for details on how to obtain such an account. In the project name, `LL` is an abbrevation for the lecture, (e.g., **SEPVS** for Software Engineering für Parallele und Verteilte Systeme or **ESSS** for Engineering Secure Software Systems), **SEM** is an abbrevation for the semester, like **WS20** for Winter 2020/2021. Lastname1 and Lastname2 are the last names of the two students in the working group. Write an email to Henning Schnoor with the URL of the repository (it suffices for one student in each group to write this mail).

Handing in of exercises and feedback to your tasks will use this git account. For non-programming exercises, answers must be submitted in one of the formats pdf, markdown, or plain text.

Exercise

Task (WhatsApp Authentication)

The instant messenges service WhatsApp for mobile phones uses the following authorization schemes:

1. To activate an account, the user needs to register a phone number. The system then sends a text message (SMS) over the mobile phone network to the user. The message contains a random number, which the user enters into the app. This activates the account.
2. To mirror the mobile app in a web browser, the user visits a special web page, which displays a QR code. The user then scans this code using the app, and can then access her account from the web interface.

Use informal notation and arguments to specify and discuss the security of the protocols underlying these authentication mechanisms. Think about whether encryption and/or signatures are used in the protocols, which (cryptographic) infrastructure is required to run the protocol, and which assumptions the protocol designers made.

Preview: Tasks for next week

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?

Exercise

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.

Crypto and (Auth) Key Exchange

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

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

secure protocol?

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

Another Attempt: Authentication and Key “Exchange”

protocol

$A \to B$	$(\mathsf{enc}^{\mathsf{a}}_{k_B}(N_A), \mathsf{enc}^{\mathsf{a}}_{k_B}(A))$	“I’m Alice, if you are Bob then you can decrypt this!”
$B \to A$	$(\mathsf{enc}^{\mathsf{a}}_{k_A}(N_A, N_B), \mathsf{enc}^{\mathsf{a}}_{k_A}(B))$	“Sure can, and we can use N_B as a symmetric key!”

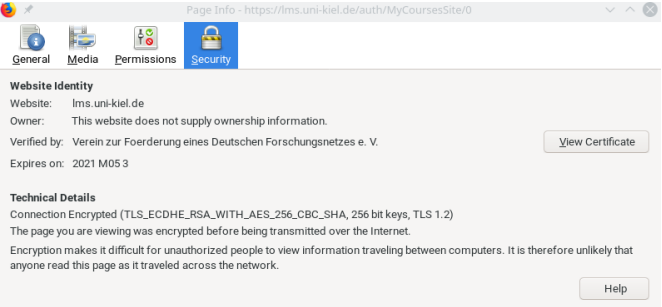
protocol intention

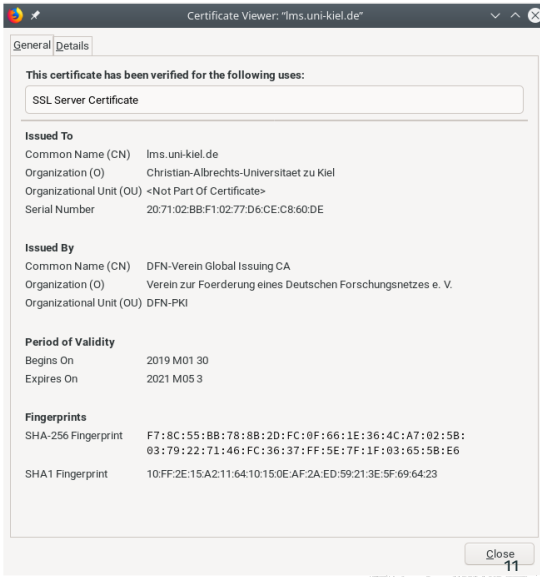
- intended security goal?
- measures to make protocol secure?

security

is the protocol secure?

TLS certificates





Example: Online-Banking

protocol

Alice, Bank

1 $A \to B \quad \mathsf{enc}^{\mathsf{a}}_{k_B}\left(\mathsf{sig}_{k_A}\left(\text{“init”}, N_A, A\right)\right)$

2 $B \to A \quad \mathsf{enc}^{\mathsf{a}}_{k_A}\left(\mathsf{sig}_{k_B}\left(\text{“recv”}, N_A, N_B, B\right)\right)$

3 $A \to B \quad \mathsf{enc}^{\mathsf{a}}_{k_B}\left(\mathsf{sig}_{k_A}\left(\text{“conf”}, N_A, N_B, A, B\right)\right)$

4 $B \to A \quad \mathsf{enc}^{\mathsf{a}}_{k_A}\left(\mathsf{sig}_{k_B}\left(\text{“ok”}\right)\right)$

5 $A \to B \quad \left(N_A, N_B, \mathsf{enc}^{\mathsf{a}}_{k_B}\left(\mathsf{sig}_{k_A}\left(\text{“transfer …”}\right)\right)\right)$

analysis

- idea of protocol?
- possible issues?
- formal analysis later (with tool)