Engineering Secure Software Systems

December 15, 2020: Distance Learning Review, Beyond Rusinowitch-Turuani Analysis: Limitations and Practice

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

Automatic Analysis: Theoretical Foundations

Automatic Analysis: Undecidability

- Arbitrarily Many Sessions
- Incomplete Algorithms

Overview

Part I: Crypto Protocols

Foundations

Cryptography

An Example and an Attack

More Examples

Formal Protocol Model

Automatic Analysis: Theoretical Foundations

Automatic Analysis: Undecidability

Arbitrarily Many Sessions

Incomplete Algorithms

Decidability

Rusinowitch-Turuani Theorem [RT03]

INSECURE is NP-complete

model

INSECURE: instances given

- theorem only covers fixed number of instances
- instance $\triangleq$ protocol session

reality

unbounded number of sessions

- many users for single server
- different (or same) users at different servers

number of concurrent TLS sessions?

Towards Automatic Analysis

seen in lecture

- formalization of NS protocol must contain sessions to find attack
  - sender instance of $A \to C$
  - receiver instance of $A \to B$
- unsatisfying: this "tells the algorithm where to look"

possible way out: over-approximate

- observation: more instances only make the situation worse (more insecure)
- therefore: let algorithm analyze the following:
  - sender instance of $A \to B$, $A \to C$, $B \to C$
  - receiver instance of $A \to B$, $A \to C$, $B \to C$
- issues?

“parallel” attacks

Rusinowitch-Turuani analysis

- instances fixed
- hence, protocol sessions fixed

problem

there are issues in protocols that need an “arbitrary” number of sessions

reference

Jonathan K. Millen. “A Necessarily Parallel Attack”. In: In Workshop on Formal Methods and Security Protocols. 1999

protocol

1	$A \rightarrow B$	A	
2	$B \rightarrow A$	$[N_1, N_2]$	
3	$A \rightarrow B$	$\mathrm{enc}^{\mathrm{a}}_{k_B}([N_1, \underbrace{N_2}_{=:x}, \underbrace{FAIL}_{=:y}])$	
4	$B \rightarrow A$	$[N_1, x, \mathrm{enc}^{\mathrm{a}}_{k_B}([x, y, N_1])]$	

more precisely

step 3:

- B verifies N_1
- B does not verify correctness of N_2
- matches N_2 with variable x, FAIL with variable y

attack

1.	$A \xrightarrow{A} B$
1'.	$(A) \xrightarrow{A} B$
2a.	$B \xrightarrow{[N_1, N_2]} (A)$
2'.	$B \xrightarrow{[N_1', N_2']} (A)$
2b.	$(B) \xrightarrow{[N_1, N_1']} A$
3.	$A \xrightarrow{\mathrm{enc}^{a}_{k_B}([N_1, N_1', \mathrm{FAIL}])} B$
4.	$B \xrightarrow{[N_1, N_1', \mathrm{enc}^{a}_{k_B}([N_1', \mathrm{FAIL}, N_1])]} (A)$
3'.	$(A) \xrightarrow{\mathrm{enc}^{a}_{k_B}([N_1', \mathrm{FAIL}, N_1])} B$
4'.	$B \xrightarrow{[N_1', \mathrm{FAIL}, \mathrm{enc}^{a}_{k_B}([\mathrm{FAIL}, N_1, N_1'])]} (A)$

security

- there is an attack
- attack requires 2 responder instances
- fact: protocol is secure if there is only one instance

consequence

- analysing a single instance is not enough
- generalization: arbitrarily many instances
- analysis of unbounded number of instances required
- not covered by Rusinowitch Turuani

Exercise

Task (the FFGG prototocol: too complicated?)

Can you come up with a simpler protocol that is secure when only one session is running, but becomes insecure if the adversary can start as many instances as she wishes? Is there an "advantage" of the ffgg protocol (as an example illustrating the need for the analysis of parallel sessions) over your example?

Unbounded Version of INSECURE

required

analysis of extension of INSECURE to (arbitrarily many) parallel sessions

formalization

- input to algorithm may not contain explicit sessions anymore
- alternative: "template" for instances
  - instance $\mathcal{I}_{A \to B}$ may be started arbitrarily often
    - "between A and B"
    - "between A and C"
    - $\dots$

issue

FAIL-rule may only be contained in "relevant" instance

Unbounded Version of INSECURE

approach

- specify instances, initial attacker knowledge as usual
- mark one instance as goal (usually contains FAIL constant)

definition

protocol P_{unb} based on P, if P_{unb} obtained from P by

- replicating instances (with fresh variables)
- changing identities in non-goal instances

issues

- changing identities must "respect" knowledge of keys
- straight-forward for asymmetric keys, more technical for symmetric keys
- see discussion in exercise class

this lecture

- no formal definition
- follow these ideas in practical security spefications
- case-study (as reading exercise) later: modeling of Needham-Schroeder in ProVerif

Exercise

Task (unbounded instances formalization)

Specify the Needham-Schroeder protocol as an instance of the decision problem UNBOUNDED-INSECURE, and show that it is insecure in this formalization. Discuss the differences between expressing the protocol using this formalism compared to the earlier formalization using the decision problem INSECURE.

Note: You do not need to make your constructions formal. The goal of this exercise is to get a good understanding on how a formal definition of INSECURE (which we did not fully state in the lecture) would look like.

Theorem

the following problem is undecidable:

Problem: UNBOUNDED-INSECURE
Input: protocol $P = (\{\mathcal{I}_0, \ldots, \mathcal{I}_{n-1}\}, I)$
Question: is there an insecure protocol P_{unb} based on P?

?

missing something? we didn't even really define UNBOUNDED-INSECURE!

simplification

result true for very simple modeling of UNBOUNDED-INSECURE

Undecidability Result

formalization for undecidability

“simplest” formalization of unbounded sessions: result covers more expressive models as well

“minimal requirements”

- protocol consists of instances $\{\mathcal{I}_0, \ldots, \mathcal{I}_{n-1}\}$, each instance has a single receive/send rule
- adversary may activate each instance as often as she wishes
- there is only a single symmetric key k shared by all protocol instances (no PKI, no identities)

undecidability proof

works for this model

Undecidability Proof I

TGI refresher

L_1, L_2 languages, L_1 undecidable and $L_1 \leq L_2$, then L_2 undecidable.

reduction

$L_1 \leq L_2$ means:

L_1-questions can be translated into L_2-questions.

formally:

there is a total, computable function $f : \Sigma^* \to \Sigma^*$ such that for all x:

$$x \in L_1 \textit{ iff } f(x) \in L_2.$$

Post's Correspondence Problem

seen in TGI

halting problem, Rice's theorem

drawback

talk about encodings of Turing machines

classical problem

Problem:	PCP (Post's correspondence problem)
Input:	$(x_1, y_1), \ldots, (x_n, y_n)$ with $x_i, y_i \in \{0, 1\}^*$
Question:	Is there a sequence $i_1, \ldots, i_\ell$ with $x_{i_1} x_{i_2} \ldots x_{i_\ell} = y_{i_1} y_{i_2} \ldots y_{i_\ell}$?

theorem

PCP is undecidable.

Emil L. Post. "A variant of a recursively unsolvable problem". In: Bull. Amer. Math. Soc. 52.4 (Apr. 1946), pp. 264–268. URL: `https://projecteuclid.org:443/euclid.bams/1183507843`

Undecidability Proof

want to show

UNBOUNDED-INSECURE is undecidable

proof (sketch)

- show PCP $\leq$ UNBOUNDED-INSECURE
- describe **computable** function $f\colon \{0,1\}^* \to \{0,1\}^*$ such that

  $x \in$ PCP *iff* $f(x) \in$ UNBOUNDED-INSECURE

Undecidability Proof II

PCP

infinite search space: find $i_1 \dots i_\ell$ with $x_{i_1} \dots x_{i_\ell} = y_{i_1} \dots y_{i_\ell}$

UNBOUNDED-INSECURE

infinite search space: choice of instances in

- protocol P_{unb} based on P
- execution order of attack

let instances perform “concatenation” of PCP strings

note

$x_1, \dots, x_n, y_1, \dots, y_n$ can be hard-coded into UNBOUNDED-INSECURE instance

issues

- adversary can use “fake PCP substrings”
- use cryptography to authenticate substrings and concatenation from PCP instance

Undecidability Proof III

input

$(x_1, y_1), \ldots, (x_n, y_n)$ PCP instance,
$x_i = x^i_1 \ldots x^i_{|x_i|}$, $y_i = y^i_1 \ldots y^i_{|y_i|}$

idea

adversary can use protocol instances to initialize domino sequence or add new tile

instances

- for each $i \in \{1, \ldots, n\}$: A^i_{init} $\quad \epsilon \rightarrow \mathsf{enc}^s_k \left([x^i_{|x_i|}, [x^i_{|x_i|-1}, [\ldots, x^i_1] \ldots]], [y^i_{|y_i|}, [y^i_{|y_i|-1}, [\ldots, y^i_1] \ldots]] \right)$
- f.e. i: A^i_{step} $\quad \mathsf{enc}^s_k([x, y]) \rightarrow \mathsf{enc}^s_k \left([x^i_{|x_i|}, [x^i_{|x_i-1|}, [\ldots, [x^i_1, x]]]], [y^i_{|y_i|}, [y^i_{|y_i-1|}, [\ldots, [y^i_1, y]]]] \right)$
- verification B_{check}: $\quad \mathsf{enc}^s_k([x, x]) \rightarrow \mathsf{FAIL}$

correctness

- A^i_{init}, A^i_{step}: Adversary gets exactly $\mathsf{enc}^s_k([t_1, t_2])$, where t_1, t_2 constructed by "domino rules"
- B_{check}: if adversary solves domino puzzle, release FAIL constant
- so: domino solvable iff protocol insecure in unbounded setting

Exercise

Task (Rusinowitch-Turuani with specified maximal number of sessions)

We saw in the lecture that the "unbounded session" version of INSECURE is undecidable. A weaker version of that problem can be obtained by allowing instances to INSECURE to be accompanied by a maximal number of copies in which the adversary may start the corresponding protocol instance (we assume a mechanism that automatically renames variables to ensure that they are "local" to the copy in which they are used). Does the "positive" part of the Rusinowitch-Turuani theorem still hold for this generalization?

Hint: You are not expected to give a formal proof of your conjectures, an informal justification suffices. Also, be explicit about how the "maximal number of copies" is specified in the input to your generalized problem.

The Edge of Decidability

lecture results

- Rusinowitch Turuani: **bounded** sessions decidable
- PCP reduction: **unbounded** sessions undecidable

middle ground?

- “restricted” unbounded sessions?
- simple loops in protocol?
- data structure processing?
- more complex protocol goals?

results

there is a lot!

The Edge of Decidability: References I

- Ralf Küsters and Tomasz Truderung. "On the Automatic Analysis of Recursive Security Protocols with XOR". In: STACS. Ed. by Wolfgang Thomas and Pascal Weil. Vol. 4393. Lecture Notes in Computer Science. Springer, 2007, pp. 646–657. ISBN: 978-3-540-70917-6
- Detlef Kähler, Ralf Küsters, and Tomasz Truderung. "Infinite State AMC-Model Checking for Cryptographic Protocols". In: LICS. IEEE Computer Society, 2007, pp. 181–192
- Henning Schnoor. "Deciding Epistemic and Strategic Properties of Cryptographic Protocols". In: ESORICS. Ed. by Sara Foresti, Moti Yung, and Fabio Martinelli. Vol. 7459. Lecture Notes in Computer Science. Springer, 2012, pp. 91–108. ISBN: 978-3-642-33166-4

The Edge of Decidability: References II

- Steve Kremer and Robert Künnemann. "Automated analysis of security protocols with global state". In: Journal of Computer Security 24.5 (2016), pp. 583–616. DOI: 10.3233/JCS-160556. URL: https://doi.org/10.3233/JCS-160556
- Jannik Dreier, Charles Duménil, Steve Kremer, and Ralf Sasse. "Beyond Subterm-Convergent Equational Theories in Automated Verification of Stateful Protocols". In: Principles of Security and Trust - 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings. Ed. by Matteo Maffei and Mark Ryan. Vol. 10204. Lecture Notes in Computer Science. Springer, 2017, pp. 117–140. ISBN: 978-3-662-54454-9. DOI: 10.1007/978-3-662-54455-6

The Edge of Decidability: References III

- Jannik Dreier, Lucca Hirschi, Sasa Radomirovic, and Ralf Sasse. "Automated Unbounded Verification of Stateful Cryptographic Protocols with Exclusive OR". In: 31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018. IEEE Computer Society, 2018, pp. 359–373. ISBN: 978-1-5386-6680-7. URL: https://ieeexplore.ieee.org/xpl/conhome/8428826/proceeding
- Robert Künnemann, Ilkan Esiyok, and Michael Backes. "Automated Verification of Accountability in Security Protocols". In: CoRR abs/1805.10891 (2018). arXiv: 1805.10891. URL: http://arxiv.org/abs/1805.10891
- ...

Undecidability: Consequences

result

(in)security with arbitrary many sessions is undecidable

consequences

- no complete “push-button” analysis of security
  - hardly unexpected
- justification for “user-unfriendly” input for Rusinowitch Turuani algorithm
  - some automatic “preprocessing” possible, but does not solve problem

analysis still required

what are options for practice?

Rusinowitch Turuani Analysis

approach

- fixed choice of instances
  - fixes identities, roles (e.g., “Alice as initiator in session with Bob”)
  - fixes number of sessions
  - fixes max. number of messages
- attack found: protocol insecure
- no attack found: secure **in this scenario**

justification

- most attacks found by checking small systems
- unusual for an attack to require “many” sessions

usual security approach

- worst-case assumptions
- “unusual” attacks are exactly what we do automatic analysis for
- situation not satisfying

manual approach

- proof using protocol structure
- for every message: *if* accepted, *then* earlier …
- then "protocol run as intended"

expensive and error-prone

automatic analysis

- problem is undecidable
- cannot have both
  - **soundness** result "protocol secure" is correct
  - **completeness** if protocol secure, this is recognized
- need to look at "incomplete" algorithms

Overview

Part I: Crypto Protocols

- Foundations
- Cryptography
- An Example and an Attack
- More Examples

Formal Protocol Model

Automatic Analysis: Theoretical Foundations

Automatic Analysis: Undecidability

- Arbitrarily Many Sessions
- Incomplete Algorithms

construct security proof

- algorithm searches for security proof
- on failure: abort or endless loop
- algorithm is correct (sound)

consequence

secure protocols are recursively enumerable (semi-decidable)

construct attack

- algorithm searches for attack
- on failure: abort or endless loop
- algorithm is correct (sound)

consequence

insecure protocols are recursively enumerable (semi-decidable)

what's wrong?

something does not add up! (*aka don't cite this slide!*)

Incomplete Algorithms

seen

- searching for security proofs and attacks can never cover everything
- way out: **heuristics** (cp. NP-complete problems)

heuristics

- there is always a price!
- what do we give up?

over-approximate attacker

- simplified attacker model
- gives “too much power” to attacker
- constructs “over-approximated” attack
- user must check attack
- algorithm sound, not complete (for security)

Incomplete Algorithms in Lecture

abstractions

- over-approximation of attacker
- leads to finite model
- apply model checking

lecture: skipped due to time constraints

logic-based modeling

- models protocol properties in (FO Horn) logic
- leads to Horn theory
- apply satisfiability testing

lecture: cover this in practice (ProVerif), brief look at theory

Computationally Nice Logics

propositional logic

- $\varphi = \exists x_1 \forall y_1 \exists x_2 \dots \forall y_n$
  $(x_1 \vee \overline{x_9} \vee y_4) \wedge \dots \wedge (y_6 \vee \overline{x_3} \vee \overline{y_{44}})$
- relevant algorithmic problems: decidable, NP-complete

first-order logic

- $\varphi = \exists x_1 \forall y_1 \exists x_2 \dots \forall y_n$
  $R_1(x_1, x_9, y_4) \vee (R_2(x_3, y_1, x_{13}) \wedge \dots)$
- relevant algorithmic problems: undecidable

complexity reduction

syntactically defined sub-logic with “nicer” complexity? Horn clauses

- allows unit resolution
- “largest” sublogic for which propositional satisfiability is PTIME-solvable [Sch78]
- still undecidable first-order theory, but “better behaved”