Undecidability of bounded security protocols

Durgin
Lincoln
Mitchell
Scedrov

FMSP, Trento, Italy, July 1999

Outline
• Goals and motivations
• Syntax and semantics
• Bounded protocols
• Example
• LLF
• Conclusions
• Questions

Motivation
• Security protocols are difficult to design and analyze
  – Subtleties involving cryptographic primitives
  – Multiple concurrent runs
• Verification of a simple protocol may combine any number of data from valid or aborted runs
• To address these problems a variety of formal methods for security has been developed
• Dolev-Yao model: idealized assumptions about cryptographic primitives and nondeterministic adversary
Motivation

- Dolev-Yao Model:
  - Messages are composed of indivisible abstract values
  - Encryption is modeled in an idealized way (Black-box model of encryption/decryption)
  - Possible intruder actions are non-deterministic and applied throughout the protocol
    - Read and block any message
    - Decompose a message into parts
    - Generate fresh data when needed
    - Compose a new message from known and generated data and put it on the network

Syntax and Semantics

- Vocabulary or first-order signature: Contains symbols that can be used
- Term: Well-formed expression over the signature
- Fact: first-order atomic formula over the signature
- State: multi-set of facts
- Transition: rewrite rule containing pre- and post conditions
  - Rewrite Rule: (using existential quantification)
  - A theory $\tau$ is a finite set of facts and rewrite rules of form $I \rightarrow \tau$

Syntax and Semantics

- Existential elimination rule:
  - Existentially quantified axiom: $\exists x \phi$
  - Some formula: $\forall$
  - New symbol $y$ for the "$x$ that is presumed to exist"
  - Derive $\forall$ from $\{y/x\} \phi$ (y is not free in any other hypothesis)
Syntax and Semantics

- Example: state $S$, rule $R$
  
  $S = \{ P(f(a)), P(b) \}$
  
  $R = (P(x) \rightarrow \exists z.Q(f(x), z))$
  
- Result: application of the rule and a new value $c$ for $z$ result in successor state $S'$
  
  $S' = \{ Q(f(f(a)), c), P(b) \}$

Bounded Protocols

- Defined multi-set rewriting framework can be used to describe finite-state and infinite-state systems
- Using functions it is possible to describe computation over unbounded data types
  
  $A \rightarrow B: \{n\}^k$  $B \rightarrow A: \{f(n)\}^k$
- Easy to encode formulas as counter or Turing machines
- Principal authentication and secrecy protocols are mostly of bounded length (finite-state systems)
- Well-founded theories: syntactic form of a class of well-founded protocol theories
Bounded Protocols

• Participants:
  – Initialization Theory:
    • Distributes keys and establishes shared information
  – Role Generation Theory:
    • Defines the roles a principals can act in
  – Protocol Theories
    • Describes which steps the principals can perform

• Intruder:
  – Two-Phase Intruder Theory
    • Initializes the intruder and describes actions it can perform

Example: Needham-Schroeder

• Initialization Theory:

\[
\text{GOODGUY} : \rightarrow \exists_{k,T} \text{GOODGUY}(k_1, k_2), KP(k_1, k_2) \\
\text{BADKEY} : \rightarrow \exists_{k,T} \text{KP}(k_1, k_2), \text{BadKey}(k_1, k_2) \\
\text{ANNK} : \text{GoodGuy}(k_1, k_2) \rightarrow \text{AnnK}(k_1, \text{GoodGuy}(k_1, k_2) \\
\text{ANNKB} : \text{BadGuy} \rightarrow \text{AnnK}(k_1, \text{BadKey}(k_1, k_2)
\]

Example: Needham-Schroeder

• Role Generation Theory:

\[
\text{ROLA} : \text{GoodGuy}(k_1, k_2) \rightarrow \text{GoodGuy}(k_1, k_2), A_0(k_c) \\
\text{ROLB} : \text{GoodGuy}(k_1, k_2) \rightarrow \text{GoodGuy}(k_1, k_2), B_0(k_c)
\]
Example: Needham-Schroeder

• Protocol Theories A and B:

\[ A1: \text{Ann}(k^\prime), A_i(k) \rightarrow \exists x. A_i(k, k^\prime, x) \land N_{\text{SI}}(\text{enc}(k^\prime, \{x, k^\prime\})), \text{Ann}(k^\prime) \]

\[ A2: A_i(k, k^\prime, x), N_{\text{SI}}(\text{enc}(k^\prime, \{x, y\})) \rightarrow A_i(k, k^\prime, x, y), N_{\text{SI}}(\text{enc}(k^\prime, y)) \]

\[ B1: B_i(k), N_{\text{SI}}(\text{enc}(k^\prime, \{x, k^\prime\})), \text{Ann}(k^\prime) \rightarrow \exists y. B_i(k, k^\prime, x, y), N_{\text{SI}}(\text{enc}(k^\prime, \{x, y\})), \text{Ann}(k^\prime) \]

\[ B2: B_i(k, k^\prime, x, y), N_{\text{SI}}(\text{enc}(k^\prime, y)) \rightarrow B_i(k, k^\prime, x, y) \]

Example: Needham-Schroeder

• Two-Phase Intruder Theory:
  − Initialization Rules:
  
  \[ \text{LRNK} \rightarrow M(k), M(k), \text{BadKey}(k, k) \]

  − I/O Rules:

  \[ \text{REC} : N_{\text{SI}}(x) \rightarrow D(x) \]

  \[ \text{SND} : C(x) \rightarrow N_{\text{SI}}(x) \]

Example: Needham-Schroeder

• Two-Phase Intruder Theory:
  − Decomposition Rules
  
  \[ \text{DCMP} : D([x, y]) \rightarrow D(x), D(y) \]

  \[ \text{LRN} : D(x) \rightarrow M(k) \]

  \[ \text{DEC} : M(k), \text{KP}(k, k), D(\text{enc}(k, x)) \rightarrow M(k), D(x), M(\text{enc}(k, x)) \]

  \[ \text{LRN} : D(\text{enc}(k, x)) \rightarrow M(\text{enc}(k, x)), A(\text{enc}(k, x)) \]

  \[ \text{DECA} : M(k), \text{KP}(k, k), A(\text{enc}(k, x)) \rightarrow M(k), \text{KP}(k, k), D(x) \]
Example: Needham-Schroeder

- Two-Phase Intruder Theory:
  - Composition Rules:
    \[ COMP : C(x), C(y) \rightarrow C(\{x, y\}) \]
    \[ USE : M(x) \rightarrow C(x), M(x) \]
    \[ ENC : M(k_x), C(x) \rightarrow C(\text{enc}(k_x, x)), M(k_x) \]
    \[ GEN : \rightarrow \exists x. M(x) \]

Protocols w/o nonces

- A \rightarrow C: (0,0,0,0)_k
  - C \rightarrow A: if sent (1,1,1,1)_k then respond K
  - B_1 \rightarrow A: if sent (x_1, x_2, x_3, 0)_k then respond (x_1, x_2, x_3, 1)_k
  - B_2 \rightarrow A: if sent (x_1, x_2, 0, 1)_k then respond (x_1, x_2, 1, 0)_k
  - B_3 \rightarrow A: if sent (x_1, 0, 1, 1)_k then respond (x_1, 0, 1, 0)_k
  - B_4 \rightarrow A: if sent (0, 1, 1, 1)_k then respond (1, 0, 0, 0)_k

- If intruder routes message from A through 2^k principals C will reveal K
- Therefore, to compromise security exponential time is needed and secrecy cannot be compromised in polynomial time
- Even without new nonces decision problem is NP and DEXP-time hard
- But would not be considered secure under Dolev-Yao assumptions

D/NEXP time hardness

- Intruder attack tree:
  - DEXP-time: exponential time needed when attack follows one path at a time
  - NEXP-time: exponential time needed when attack starts all paths at once (no shortest path is exponential)
Undecidability

• **Theorem 4.** The implication problem for existential Horn clauses without function symbols is undecidable. In particular, there is no algorithm for deciding whether a set of existential Horn clauses without function symbols implies a single atomic formula $A(b_1, \ldots, b_k)$ without function symbols or variables.

This theorem has a straightforward direct proof based on axiomatizing a Cook’s theorem style Turing machine tableau.

Undecidability

• Protocol theory in restricted form:
  
  $A(\ldots), N(\ldots) \rightarrow \exists \ldots \exists A(\ldots), N(\ldots)$

  $A_i =$ states of agents

  $N_i =$ network messages

  Persistent facts removed

  Agent:

  $A_i, N_i, \forall x, (\alpha_1 \wedge \ldots \wedge \alpha_l) \rightarrow \exists y_1 \ldots \exists y_l, A_i, N_i, (\beta_1 \wedge \ldots \wedge \beta_m)$

  Can be written as a Horn clause:

  $\forall x_1 \ldots \forall x_l (\alpha_1 \wedge \ldots \wedge \alpha_l) \rightarrow \exists y_1 \ldots \exists y_m (\beta_1 \wedge \ldots \wedge \beta_m)$

Undecidability

• Intruder stores encryption of all atomic formulas derivable from a given Horn theory without function symbols

• By replaying these messages it is trying to carry out an arbitrary deduction

• But from theorem 4 follows that there is no algorithm to decide if this deduction is successful or not

• Therefore, secrecy is undecidable
LLF

- LF logical framework
  - Three-level calculus for objects, families, and kinds
  - Offers a deduction system to make proofs
- Extension of the LF logical framework
  - Linear implication (\(\rightarrow\))
  - Additive conjunction (\(&\))
  - Additive truth (\(T\))

LLF

- LLF (Linear Logic Framework), a linear logic tool is used to:
  - To search executions of a protocol and intruder for protocol flaws
  - Formally verify proofs of protocol optimizations

LLF

- Multiset rewriting and LLF:
  Each transition rule
  \[ A_1, \ldots, A_n \rightarrow \exists x.B_1, \ldots, B_m \]
  can be written as
  \[ A_1 \otimes \cdots \otimes A_n \rightarrow \exists x.B_1 \otimes \cdots \otimes B_m \]
  This allows to use linear logic tools for the protocol analysis
Example of Rule A1:

\[
A1: AnnK(k'_r), A_0(k'_r) \\
\rightarrow \exists k. A(k_r, k'_r, x). N_{A_0}(\text{enc}(k'_r, (x, k'_r))), AnnK(k'_r)
\]

\[\text{naA1: loop} \]

\[\text{a} \rightarrow \text{annKey B} \]

\[\text{a} \rightarrow \text{a}0 \text{ A} \]

\[\text{a} \rightarrow \text{a}1 \text{ A B (@ Na)} \]

\[\text{o} \rightarrow \text{toNet na l} \]

\[\text{crypt (((@ Na) * (@ (k2m A))) B)} \]

\[\text{o} \rightarrow \text{loop}. \]

Conclusions

- Secrecy is DEXP-time hard when no new data (not PSPACE hard!)
- Otherwise, secrecy is undecidable, even for restricted classes of protocols
- Useful notation for examining Dolev-Yao model with LLF tools
- It is possible to perform lower-bounds analysis

References

Questions

1. What is the role of nonces in the undecidability of security protocols? Explain.
   - For protocols w/o nonces already non-polynomial time needed to decide secrecy (DEXPTIME-hard)
   - With an unlimited number of nonces secrecy becomes undecidable
   - Intuitive reasoning:
     • Bounded number of nonces, arbitrary number of protocol runs
       - No new data
       - All possible combinations can be checked
       - Result: Decidability
     • Unbounded number of nonces and arbitrary number of protocol runs
       - Not all possible runs can be checked
       - Result: Undecidability
   - Theoretical reasoning:
     • Using Horn fragments

Questions

2. What are the advantages of using multiset rewriting, especially in regard to the intruder model used?
   • Defined multi-set rewriting framework can be used to describe finite-state and infinite-state systems
   • Using functions it is possible to describe computation over unbounded data types
   • Easy to encode as counter or Turing machines
   • Dolev-Yao model: Non-deterministic intruder
     - Rules can resemble intruder actions: Read, Send, Compose, Decompose, Generate
     - Arbitrary choice of a rule that can be applied gives non-determinism
     - Intruders can be reused for different protocols, only message format has to be adapted

Questions

3. What does the undecidability of secrecy mean for real protocol instances? What does it mean for models used for formal verification?
   • Real protocol instances
     - Using unbounded number of nonces makes it a mutual disadvantage:
       • Makes it impossible for designer to verify secrecy
       • Harder for intruder to undermine security of protocol
   • Formal verification models
     - Nonces are abstracted and/or limited in most formal verification methods
     - Therefore, secrecy proofs are usually made on this limited models, but do not determine the secrecy of real protocols