A Theory of Assertions for Dolev-Yao Models

Vaishnavi Sundararajan
Univ. Rennes, CNRS, IRISA

19th March 2019

Introduction

- Security protocol: a pattern of communications to achieve a security goal in an insecure environment.
- Each communication is of the form $A \longrightarrow B\colon m$.
- Malicious intruder can mix-and-match messages (even without breaking cryptography).
- Need formal analysis of protocols to guarantee security goals!

Logical Flaws: Example

$$A \rightarrow B : \{m\}_{pk(B)}$$

$$B \rightarrow A : \{m\}_{pk(A)}$$

Logical Flaws: Example

$$A \rightarrow B : \{m\}_{pk(B)}$$

$$B \rightarrow A : \{m\}_{pk(A)}$$

$$A \rightarrow \quad : \{m\}_{pk(B)}$$

$$I \rightarrow B : \{m\}_{pk(B)}$$

$$B \rightarrow I : \{m\}_{pk(I)}$$

$$\rightarrow A : \{m\}_{pk(A)}$$

Dolev-Yao Model

- Framework for analysis of security protocols.
- Messages are abstract terms rather than bit strings.
- Encryption, hashing etc. abstract functions on terms.
- Cryptography assumed to be perfect, no cryptanalysis!
- Formalize properties, verify.

Dolev-Yao Model: Intruder

Intruder *I* cannot break encryption, but can

- see any message
- block any message
- redirect any message
- generate messages — according to set rules!
- send messages in someone else's name
- initiate new communication according to the protocol

Certification in Dolev-Yao

- Dolev-Yao treats all messages as "terms".
- What if protocol involves certificates? For authorization, delegation etc.
- Encoded as terms in Dolev-Yao — bit commitment, mathematical operations, protocol-specific tagging etc.
- Not always concise/readable!

Example

- *A* sends to *B* m encrypted in some key k unknown to *B*, along with a certificate which says m is either a or b.
- Encode this certificate as a term in Dolev-Yao algebra.
- Uses 1-out-of-2 encryption: For a given $\{m\}_k$, show that it is of the form $\{m_i\}_k$ where $m_i \in \{m_0, m_1\}$, without revealing i.
- Needs multiplication, exponentiation, and hashing!

ZKP Terms [BHM08]

- Extend Dolev-Yao model with "ZKP terms".
- $\mathrm{ZK}_{p,q}(\alpha_1,\ldots,\alpha_p ; \beta_1,\ldots,\beta_q ; F)$
- αs: private; βs: public; F defines link between αs and βs.
- More readable certificate than encoding into terms.

$$\mathrm{ZK}_{2,3}\left(m, k ; \{m\}_k, a, b ; \beta_1 = enc(\alpha_1, \alpha_2) \wedge (\alpha_1 = \beta_2 \vee \alpha_1 = \beta_3)\right)$$

BHM08: Backes, M.; Hritcu, C.; Matteo, M. (2008) "Type-checking zero-knowledge." In *Proc. CCS '08*, 357–370.

ZKP Terms (Contd.)

- Sounds great! So why reinvent the wheel?
- Consider $\{m = a \text{ or } m = b\}$ and $\{m = a \text{ or } m = c\}$ with $b \neq c$.
- Would like to be able to derive $m = a$ from these two.
- ZKP terms don't allow derivations. Cannot infer $m = a$ from these certificates in this system.

Overall Idea

- Extend Dolev-Yao model with a class of abstract objects called 'assertions' which capture certification.

- Assertions are distinct from terms, and clearly specify the statements of the certificates they model.

- Inference on assertions is possible, independent of underlying implementation.

Assertions

- Assertions have the following syntax.

  $\alpha := t_1 = t_2 \mid P(t) \mid \alpha_1 \wedge \alpha_2 \mid \alpha_1 \vee \alpha_2 \mid \exists x.\ \alpha \mid A \textit{ says } \alpha$

- P is any application-specific predicate.

- *says* allows agents to "sign" an assertion as coming from them.

- Existential quantification lets agents hide witnesses.

- Earlier example now looks as follows:

  $$A \to B : \{m\}_k, \exists xy.[\{m\}_k = \{x\}_y \wedge (x = a \vee x = b)]$$

Existential Quantification

- When exactly can one existentially quantify out a term from an assertion?
- m from $m = t$? m from $\{m\}_k = t$?
- Quantification becomes complicated in the presence of encryption!

Abstractability

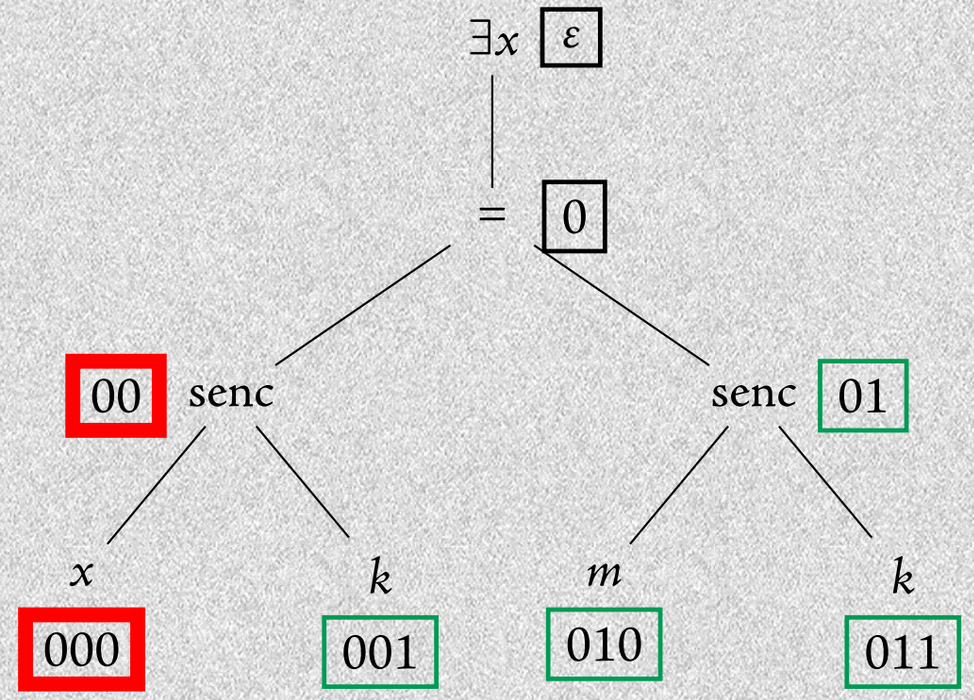
- Position p inside term t is 'abstractable' if we can replace the subterm at p with something else and build rest of t back up.

- We consider a notion of abstractability w.r.t. a set S, if we can use (some) terms in S to build relevant parts of t.

- Lift to assertions, but handle carefully in the presence of existential quantification.

Abstractability: Assertions

- $S = \{\text{senc}(m, k), k\}$
- $\alpha = \exists x.[\text{senc}(x, k) = \text{senc}(m, k)]$
- $\text{abs}(S, \alpha) = \{001, 01, 010, 011\}$



Inference system for Assertions

- Sequents of the form $S; A \vdash \alpha$.

- Simple equality rule: if t derivable from S, can state $t = t$.

- Some rules for manipulating equality make use of abstractability.

Inference system for Assertions

- Abstractability used by projection, substitution, existential introduction etc.

- Can go from $\alpha(t)$ to $\alpha(u)$ if all occurrences of t abstractable from α w.r.t. the set of terms S.

- Restricted contradiction rule: two terms t and u such that the structure of t and u can be determined to be different, but $S; A \vdash t = u$.

$$\frac{}{S;\, A \cup \{\alpha\} \vdash \alpha}\; ax$$

$$\frac{S \vdash_{dy} t}{S;\, A \vdash t = t}\; eq$$

$$\frac{S;\, A \vdash \mathrm{f}(t_1, \ldots, t_r) = \mathrm{f}(u_1, \ldots, u_r)}{S;\, A \vdash t_i = u_i}\; proj_i \quad [t_i, u_i \text{ abstractable w.r.t. } S]$$

$$\frac{S;\, A \vdash t = u}{S;\, A \vdash \alpha}\; \bot \quad [S \Vdash t \bot u]$$

$$\frac{S;\, A \vdash \alpha[t]_p \quad S;\, A \vdash t = u}{S;\, A \vdash \alpha[u]_p}\; subst \quad [t \text{ abstractable w.r.t. } S,\ S \vdash_{dy} u]$$

Inference system for Assertions

- *A says* is essentially a signature with *A*'s secret key, can be removed by an *unsay* rule.
- Rules for logical operators $\wedge$, $\vee$ and $\exists$ are as in standard intuitionistic logic (caveat of abstractability for $\exists$i).

Assertions: Actions

- As with terms, agents can send and receive assertions.
- Can now branch based on the derivability of assertions: confirm and deny actions.
- An *A*-action is a send, receive, confirm or deny by *A*.
- Actions specified with as much pattern as possible for terms, with variables for terms unknown to recipeint.

Runtime Model

- Each agent accumulates terms and assertions generated and received, in a knowledge state $(X; \Phi)$.
- Represent by $(X_A; \Phi_A)$ the knowledge state of agent A.
- Represent by $(X_I; \Phi_I)$ the knowledge state of the intruder I.
- Knowledge states used to enable actions, and possibly updated after performing actions.

Runtime Model (Contd.)

- A protocol is just a set of roles.
- Can consider various instantiations of roles — sessions.
- A run is an admissible (according to enabling conditions!) interleaving of such sessions.
- One can think of a transition system with states that keep track of agents' knowledge and all the sessions in progress, where enabled actions induce transitions.

Enabling & Updates

Action	Enabling conditions	Updates
A sends t, α with new nonces $\vec{m}$	$X_A \cup \{\vec{m}\} \vdash_{dy} t$ $X_A \cup \{\vec{m}\}; \Phi_A \vdash \alpha$	$X'_A = X_A \cup \{\vec{m}\}$ $X'_I = X_I \cup \{t\}$ $\Phi'_I = \Phi_I \cup \{\alpha\}$
A receives t, α	$X_I \vdash_{dy} t$ $X_I; \Phi_I \vdash \alpha$	$X'_A = X_A \cup \{t\}$ $\Phi'_A = \Phi_A \cup \{\alpha\}$
A : *confirm* α	$X_A; \Phi_A \vdash \alpha$	No update
A : *deny* α	$X_A; \Phi_A \nvdash \alpha$	No update

Case Study: FOO e-Voting Protocol

- Proposed by Fujioka, Okamoto and Ohta in 1992. [FOO92]
- Voter contacts admin, who checks voter's id and authenticates.
- Authenticated voter then sends vote anonymously to collector.
- Admin should not know vote, collector should not know id.
- Terms-only model ensures this via blind signatures.

FOO92: Fujioka, A.; Okamoto, T.; Ohta, K. (1992), "A Practical Secret Voting Scheme for Large Scale Elections", *Advances in Cryptology — AUSCRYPT '92*, 244–251.

FOO Protocol: Terms-only

$V \rightarrow A \quad : \quad V, \{\text{blind}(\{v\}_r, b)\}_{sg(V)}$

$A \rightarrow V \quad : \quad \{\text{blind}(\{v\}_r, b)\}_{sg(A)}$

$V \not\rightarrow C \quad : \quad \{\{v\}_r\}_{sg(A)}$

$C \rightarrow \quad : \quad list, \{\{v\}_r\}_{sg(A)}$

$V \not\rightarrow C \quad : \quad r$

$$\text{unblind}(\{\text{blind}(t, b)\}_{sg(A)}, b) = \{t\}_{sg(A)}$$

FOO Protocol: What we want

$V \to A$: $\{v\}_k$, *"V wants to vote with this encryption of a valid vote"*

$A \to V$: *"V is eligible and wants to vote with the term sent earlier"*

$V \not\to C$: $\{v\}_{k'}$, *"Some eligible agent was authorized by A to vote with a valid vote, this term is a re-encryption of that same vote."*

A does not have to modify V's term (which contains the vote) in order to certify it!

FOO Protocol: Assertions

$$
\begin{aligned}
V \to A \quad &: \quad \{v\}_{r_A}, V \textit{ says } \{\exists x, r : \{x\}_r = \{v\}_{r_A} \wedge \text{valid}(x)\} \\
A \quad &: \quad \textit{deny} \;\; \exists x : \text{voted}(V, x) \\
&\qquad \textit{insert} \;\; \text{voted}(V, \{v\}_{r_A}) \\
A \to V \quad &: \quad A \textit{ says } \Big[\text{elg}(V) \wedge \text{voted}(V, \{v\}_{r_A}) \\
&\qquad\qquad \wedge V \textit{ says } \{\exists x, r : \{x\}_r = \{v\}_{r_A} \wedge \text{valid}(x)\}\Big] \\
V \not\to C \quad &: \quad \{v\}_{r_C}, r_C, \\
&\qquad \exists X, y, s : \Big\{A \textit{ says } \Big[\text{elg}(X) \wedge \text{voted}(X, \{y\}_s) \\
&\qquad\qquad\qquad \wedge X \textit{ says } \{\exists x, r : \{x\}_r = \{y\}_s \\
&\qquad\qquad\qquad\qquad \wedge \text{valid}(x)\}\Big] \\
&\qquad\qquad \wedge y = v\Big\}
\end{aligned}
$$

Verification

- Derivability problem: Given a finite set of terms X, a finite set of assertions Φ, and an assertion α, is it the case whether $X; \Phi \vdash \alpha$?

- Insecurity problem: Given a protocol Pr and a designated secret assertion α, is there a run of Pr at the end of which $X_I, \Phi_I \vdash \alpha$?

Derivability Problem

- Proof search: Start from the desired conclusion, try to build a proof tree using inference system.
- For assertions, slightly problematic because of two reasons:
  - ∨e: Need to check that the conclusion of the rule is derivable from each disjunct separately; two proofs to search for!
  - ∃i: Need to pick appropriate term as witness; unbounded search!

Derivability Problem

- Consider down-closures. $(S;A)$ said to be down-closed if:
  - S contains all bound variables of A
  - If $\beta \wedge \gamma \in A$, then $\{\beta, \gamma\} \subseteq A$
  - If $\beta \vee \gamma \in A$, then $\beta \in A$ or $\gamma \in A$
  - If $\exists x.\beta \in A$, then $\beta \in A$
  - If a *says* $\beta \in A$, then $\beta \in A$
- $(T;B)$ dc of $(S;A)$ if it is minimal, dc with $S \subseteq T$ & $A \subseteq B$.

Derivability Problem

- Helpful because various "left" properties hold about this system.
  - Conjunction: $S; A \cup \{\beta \wedge \gamma\} \vdash \alpha$ iff $S; A \cup \{\beta, \gamma\} \vdash \alpha$.
  - Disjunction: $S; A \cup \{\beta \vee \gamma\} \vdash \alpha$ iff $S; A \cup \{\beta\} \vdash \alpha$ and $S; A \cup \{\gamma\} \vdash \alpha$.
  - Exists: $S; A \cup \{\exists x.\beta\} \vdash \alpha$ iff $S \cup \{x\}; A \cup \{\beta\} \vdash \alpha$. *
  - Says: $S; A \cup \{\text{a } \textit{says } \beta\} \vdash \alpha$ iff $S; A \cup \{\beta, \text{a } \textit{says } \beta\} \vdash \alpha$.
- Enough to consider $\text{trim}(B) = \{t = u \mid t = u \in B\}$ for a dc $(T; B)$.

* Conditions apply

Derivability Problem

- $S; A \vdash \alpha$ iff all dc $T; B \vdash \alpha$.
- $T; B \vdash \alpha$ iff $T; \mathrm{trim}(B) \vdash \alpha$ using core $= \{ax, eq, \bot, subst, proj, \wedge\mathrm{i}, \vee\mathrm{i}, \exists\mathrm{i}\}$.
  - Proofs in core have a normal form — can be decomposed into two parts:
    - Proofs of $T; \mathrm{trim}(B) \vdash_{eq} \mu(t)=\mu(u)$ for each $t = u \in E$, and
    - A proof of $T; E \vdash \alpha$ using only $\wedge\mathrm{i}, \vee\mathrm{i}, \exists\mathrm{i}$, says

μ: assigns witnesses for the quantifiers

E: set of equalities that are subformulas of α

Derivability Problem

- Problem of μ assigning unboundedly large terms for witnesses for ∃i remains.
- Adapt idea of 'small substitutions', as presented by [RT03] for the terms-only system.
- Key notion there: If the intruder can achieve the same 'view' with a smaller term, no need to use a larger term.
- Have μ, want small ν s.t. for t, u subterms of S, A, α
  if $S; A \vdash_{eq} \mu(t) = \mu(u)$ then $S; A \vdash_{eq} \nu(t) = \nu(u)$.

RT03: Rusinowitch, M.; Turuani, M.(2003),"Protocol insecurity with a finite number of sessions and composed keys is np-complete", *Theoretical Computer Science*, 299(1- 3):451–475.

Derivability Problem

- For every down-closure $(T; B)$, need to guess a set of equalities E and a small substitution μ s.t. $(T; B)$ derives $\mu(E)$, and $T; E \vdash \alpha$.
  - $(T; B)$ is linear in the size of $(S; A)$
  - E polynomial in the size of α (since subformulas)
  - μ polynomial in the size of S; A and α (since small)
  - A proof of $T; E \vdash \alpha$ linear in the size of α.
- Obtain a Π_2, i.e. a $\mathrm{coNP}^{\mathrm{NP}}$ procedure.

Derivability Problem

- This bound is tight — the problem is Π_2-complete.
- Reduction from the validity problem for QBF formulas of the form $\forall p_1 \ldots p_m \exists q_1 \ldots q_n \psi$.
- Can define for each such QBF formula S, A and α s.t. $S; A \vdash \alpha$ iff $\forall p_1 \ldots p_m \exists q_1 \ldots q_n \psi$ is valid.

Insecurity Problem

- For the derivability problem, just one substitution μ for the witnesses for ∃i. Here, the intruder can inject terms, so a σ for the input variables in $(S; A)$ as well as μ.
- Can get small ν instead of μ as earlier. But not yet clear how to do that for σ in the presence of μ.
- Solve the insecurity problem for finitely many sessions and bounded σ. Guess a σ and then use the derivability algorithm.
- Reduction from QBF validity gives us Π_3-completeness.

Summary

- Extended the Dolev-Yao model with assertions.
- Case study via the FOO e-voting protocol.
- Studied derivability and insecurity problems.
- Derivability Π_2-complete, insecurity (bounded σ) Π_3-complete.

Future Work

- Effect of adding other operators into assertion syntax
- Derivability in the presence of equational theories
- Implementation for assertions
- Tool support

References

- Existential assertions for voting protocols
  R Ramanujam, Vaishnavi Sundararajan and S P Suresh
  Proc. FC 2017 Workshops (Voting '17), Springer LNCS vol. 10323, 337–352.

- The complexity of disjunction in intuitionistic logic
  R Ramanujam, Vaishnavi Sundararajan and S P Suresh
  Proc. LFCS 2016, Springer LNCS vol. 9537, 349–363.

- Extending Dolev-Yao with assertions
  R Ramanujam, Vaishnavi Sundararajan and S P Suresh
  Proc. ICISS 2014, Springer LNCS vol. 8880, 50–68.

Thank you!