Extensions of Dolev-Yao theory and the secrecy problem

A Baskar (CMI) R Ramanujam (IMSc) S P Suresh (CMI)

Automata, Concurrency, and Timed Systems
CMI
February 2, 2010

Outline

1. Security protocols
2. Dolev-Yao model
3. Extensions of the basic model
4. An automaton construction

Outline

1. Security protocols
2. Dolev-Yao model
3. Extensions of the basic model
4. An automaton construction

Security protocols

Security protocols are three line programs that people still manage to get wrong.

Roger Needham

An example protocol

$$A \rightarrow B : \{n\}_B$$
$$B \rightarrow A : \{n\}_A$$

Attacks

- Another look at the same

Attacks

- Another look at the same

  $A!B : \{n\}_B$

  $A?B : \{n\}_A$

Attacks

- Another look at the same

$A!B : \{n\}_B \qquad B?A : \{n\}_B$

$A?B : \{n\}_A \qquad B!A : \{n\}_A$

Attacks

- Another look at the same

$$A!B:\{n\}_B \qquad B?A:\{n\}_B$$
$$A?B:\{n\}_A \qquad B!A:\{n\}_A$$

- …and an attack!

Attacks

- Another look at the same

  $A!B:\{n\}_B \qquad B?A:\{n\}_B$

  $A?B:\{n\}_A \qquad B!A:\{n\}_A$

- …and an attack!

  $A!B:\{p\}_B$

Attacks

- Another look at the same

  $A!B:\{n\}_B$ $\quad$ $B?A:\{n\}_B$

  $A?B:\{n\}_A$ $\quad$ $B!A:\{n\}_A$

- …and an attack!

  $A!B:\{p\}_B$

  $B?I:\{p\}_B$

Attacks

- Another look at the same

  $A!B:\{n\}_B \qquad B?A:\{n\}_B$

  $A?B:\{n\}_A \qquad B!A:\{n\}_A$

- …and an attack!

  $A!B:\{p\}_B$

  $\qquad\qquad B?I:\{p\}_B$

  $\qquad\qquad B!I:\{p\}_I$

- Another look at the same

  $A!B:\{n\}_B \qquad B?A:\{n\}_B$

  $A?B:\{n\}_A \qquad B!A:\{n\}_A$

- ...and an attack!

  $A!B:\{p\}_B$

  $\qquad\qquad B?I:\{p\}_B$

  $\qquad\qquad B!I:\{p\}_I$

  $A?B:\{p\}_A$

Outline

1 Security protocols

2 Dolev-Yao model

3 Extensions of the basic model

4 An automaton construction

The framework

- Protocol specifications mention abstract names and roles
- Runs are got by

The framework

- Protocol specifications mention abstract names and roles
- Runs are got by
  - instantiating the roles to many sessions
  - and interleaving them arbitrarily
  - in the presence of an all powerful intruder
  - respecting some admissibility conditions.

The framework

- Protocol specifications mention abstract names and roles
- Runs are got by
  - instantiating the roles to many sessions
  - and interleaving them arbitrarily
  - in the presence of an all powerful intruder
  - respecting some admissibility conditions.
- Intruder can

The framework

- Protocol specifications mention abstract names and roles
- Runs are got by
  - instantiating the roles to many sessions
  - and interleaving them arbitrarily
  - in the presence of an all powerful intruder
  - respecting some admissibility conditions.
- Intruder can
  - learn messages travelling over the network
  - construct new messages and play them back (under a possibly assumed identity).

The framework

- Protocol specifications mention abstract names and roles
- Runs are got by
  - instantiating the roles to many sessions
  - and interleaving them arbitrarily
  - in the presence of an all powerful intruder
  - respecting some admissibility conditions.
- Intruder can
  - learn messages travelling over the network
  - construct new messages and play them back (under a possibly assumed identity).
- Admissibility Are the messages sent by the intruder constructible given her current knowledge?

The framework

- Protocol specifications mention abstract names and roles
- Runs are got by
  - instantiating the roles to many sessions
  - and interleaving them arbitrarily
  - in the presence of an all powerful intruder
  - respecting some admissibility conditions.
- Intruder can
  - learn messages travelling over the network
  - construct new messages and play them back (under a possibly assumed identity).
- Admissibility Are the messages sent by the intruder constructible given her current knowledge?
- Secrecy problem Is a secret leaked to the intruder by some run of the protocol?

Message construction rules

$\dfrac{}{t}\,Ax\ (t \in X)$	
$\dfrac{(t_0, t_1)}{t_i}\,split_i\ (i = 0, 1)$	$\dfrac{t_0 \quad t_1}{(t_0, t_1)}\,pair$
$\dfrac{\{t\}_k \quad inv(k)}{t}\,decrypt$	$\dfrac{t \quad k}{\{t\}_k}\,encrypt$
destruction rules	construction rules

Figure: Derivation rules (from X)

Decidability

- The passive intruder deduction problem: given X and t, check if there is proof of $X \vdash t$
- This problem is decidable.
  - A notion of normal proofs.
  - If $X \vdash t$ is provable, there is a normal proof of $X \vdash t$.
  - Every term r occurring in a normal proof of $X \vdash t$ is a subterm of $X \cup \{t\}$.
  - Derive bounds on the size of normal proofs from this.

Non-normal proofs

- An example:

$$\dfrac{\dfrac{\dfrac{}{t}\,Ax \quad \dfrac{}{t}\,Ax}{(t,t)}\,pair}{t}\,split_0$$

Non-normal proofs

- An example:

$$\dfrac{\dfrac{}{t}\,Ax \quad \dfrac{}{t}\,Ax}{\dfrac{(t,t)}{t}\,split_0}\,pair$$

- Another one:

$$\dfrac{\dfrac{\dfrac{}{t}\,Ax \quad \dfrac{}{k}\,Ax}{\{t\}_k}\,encrypt \quad \dfrac{}{k}\,Ax}{t}\,decrypt$$

Normalization rules

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\pi_1 \\ t\end{matrix} \qquad \begin{matrix}\vdots\,\pi_2 \\ t'\end{matrix}}{(t, t')}\,\mathit{pair}}{t}\,\mathit{split}_0 \quad \rightsquigarrow \quad \begin{matrix}\vdots\,\pi_1 \\ t\end{matrix}$$

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\pi_1 \\ t\end{matrix} \qquad \begin{matrix}\vdots\,\pi_2 \\ k\end{matrix}}{\{t\}_k}\,\mathit{pair} \qquad \begin{matrix}\vdots\,\pi_3 \\ \mathit{inv}(k)\end{matrix}}{t}\,\mathit{decrypt} \quad \rightsquigarrow \quad \begin{matrix}\vdots\,\pi_1 \\ t\end{matrix}$$

Subterm property

Lemma

If π is a normal proof of $X \vdash t$ and r occurs in π:

- $r \in st(X \cup \{t\})$
- *if π ends in a destruction rule, then $r \in st(X)$.*

Subterm property

Lemma

If π is a normal proof of $X \vdash t$ and r occurs in π:

- $r \in st(X \cup \{t\})$
- *if π ends in a destruction rule, then $r \in st(X)$.*

$$\dfrac{\begin{array}{c}\vdots\,\pi_1\\ t\end{array} \qquad \begin{array}{c}\vdots\,\pi_2\\ k\end{array}}{\{t\}_k}\ \textit{encrypt}$$

- if r occurs in π_1,
  $r \in st(X \cup \{t\})$
- if r occurs in π_2,
  $r \in st(X \cup \{k\})$
- therefore, if r occurs in π,
  $r \in st(X \cup \{\{t\}_k\})$

Subterm property

Lemma

If π is a normal proof of $X \vdash t$ and r occurs in π:

- $r \in st(X \cup \{t\})$
- *if π ends in a destruction rule, then $r \in st(X)$.*

$$\dfrac{\begin{matrix} \vdots\, \pi_1 \\ \{t\}_k \end{matrix} \qquad \begin{matrix} \vdots\, \pi_2 \\ inv(k) \end{matrix}}{t}\ decrypt$$

- if r occurs in π_1 or π_2, $r \in st(X \cup \{\{t\}_k\})$
- since π is normal, π_1 does not end with the *encrypt* rule
- so it ends with a destruction rule, and $\{t\}_k \in st(X)$
- so any r occurring in π is in $st(X)$.

A polynomial-time algorithm

- The height of a normal proof of $X \vdash t$ is bounded by $n = |st(X \cup \{t\})|$.
- Let $X_0 = X$
- Compute $X_i = \text{one-step-derivable}(X_{i-1}) \cap st(X \cup \{t\})$, for $i \leq n$
- Check if $t \in X_n$!

Outline

1 Security protocols

2 Dolev-Yao model

3 Extensions of the basic model

4 An automaton construction

Extensions

- What about other cryptographic primitives?
- Diffie-Hellman encryption, exclusive or, homomorphic encryption, blind signatures, ...
- A large body of results: Rusinowitch & Turuani 2003, Millen & Shmatikov 2001, Comon & Shmatikov 2003, Chevalier, Küsters, Rusinowitch & Turuani 2005, Delaune & Jacquemard 2006, Bursuc, Comon & Delaune 2007, Lafourcade, Lugiez & Treinen 2007

Cancellations: the xor case

- One new construction rule:

$$\frac{t_1 \quad \cdots \quad t_n}{(t_1 \oplus \cdots \oplus t_n)\downarrow}$$

- Normalization rules: no more than one occurrence of any term as a premise of an *xor* rule
- Simplify

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\pi'_1\\ t'_1\end{array} \quad \cdots \quad \begin{array}{c}\vdots\,\pi'_m\\ t'_m\end{array}}{t_1}\,\mathit{xor} \qquad \begin{array}{c}\vdots\,\pi_2\\ t_2\end{array} \quad \cdots \quad \begin{array}{c}\vdots\,\pi_n\\ t_n\end{array}}{t}\,\mathit{xor}$$

Cancellations: the xor case

- One new construction rule:

$$\frac{t_1 \quad \cdots \quad t_n}{(t_1 \oplus \cdots \oplus t_n)\downarrow}$$

- Normalization rules: no more than one occurrence of any term as a premise of an *xor* rule
- Simplify

$$\dfrac{\dfrac{\dfrac{\vdots\ \pi'_1}{t'_1} \quad \cdots \quad \dfrac{\vdots\ \pi'_m}{t'_m}}{t_1}\,\mathit{xor} \qquad \dfrac{\vdots\ \pi_2}{t_2} \quad \cdots \quad \dfrac{\vdots\ \pi_n}{t_n}}{t}\,\mathit{xor}$$

to

$$\dfrac{\dfrac{\vdots\ \pi'_1}{t'_1} \quad \cdots \quad \dfrac{\vdots\ \pi'_m}{t'_m} \quad \dfrac{\vdots\ \pi_2}{t_2} \quad \cdots \quad \dfrac{\vdots\ \pi_n}{t_n}}{t}\,\mathit{xor}$$

Subterm property

- The cases other than *xor* go through smoothly
- *xor* brings cancellations to the party!

$$\dfrac{\begin{array}{c}\vdots\ \pi_1 \\ t_1 \oplus t_2\end{array} \qquad \begin{array}{c}\vdots\ \pi_2 \\ t_2 \oplus t_3\end{array}}{t_1 \oplus t_3}$$

- t_2 is not a subterm of the conclusion. Is it a subterm of the premises?

Subterm property

- The cases other than *xor* go through smoothly
- *xor* brings cancellations to the party!

$$\dfrac{\begin{array}{c}\vdots\,\pi_1\\ t_1 \oplus t_2\end{array} \qquad \begin{array}{c}\vdots\,\pi_2\\ t_2 \oplus t_3\end{array}}{t_1 \oplus t_3}$$

- t_2 is not a subterm of the conclusion. Is it a subterm of the premises? One can argue that it is!

Subterm property

- The cases other than *xor* go through smoothly
- *xor* brings cancellations to the party!

$$\dfrac{\begin{matrix}\vdots\,\pi_1 \\ t_1 \oplus t_2\end{matrix} \qquad \begin{matrix}\vdots\,\pi_2 \\ t_2 \oplus t_3\end{matrix}}{t_1 \oplus t_3}$$

- t_2 is not a subterm of the conclusion. Is it a subterm of the premises? One can argue that it is!
- Moral: We cannot work with syntactic subterms any more, but there is still some way of bounding the set terms occurring in proofs.

Term syntax

$$\mathscr{T} ::= m \mid (t_1, t_2) \mid [t_1, t_2] \mid \{t\}_k$$

Normal terms: Terms that do not contain a subterm of the form $\{[t_1, t_2]\}_k$. For a term t, get its normal form $t\downarrow$ by pushing encryptions over blind pairs, all the way inside.

Term syntax

$$\mathscr{T} ::= m \mid (t_1, t_2) \mid [t_1, t_2] \mid \{t\}_k$$

Normal terms: Terms that do not contain a subterm of the form $\{[t_1, t_2]\}_k$. For a term t, get its normal form $t\downarrow$ by pushing encryptions over blind pairs, all the way inside.

$\dfrac{[t, t'] \quad k}{[\{t\}_k\downarrow, \{t'\}_k\downarrow]}\ \mathit{encrypt}$	$\dfrac{\{t\}_k\downarrow \quad \mathit{inv}(k)}{t}\ \mathit{decrypt}$	$\dfrac{(t_0, t_1)}{t_i}\ \mathit{split}_i$	$\dfrac{[t_0, t_1]\downarrow \quad t_i\downarrow}{t_{1-i}}\ \mathit{blindsplit}_i$
$\dfrac{}{t}\ \mathit{Ax}\ (t \in X)$	$\dfrac{t \quad k}{\{t\}_k\downarrow}\ \mathit{encrypt}\ (t \text{ not a blind pair})$	$\dfrac{t_1 \quad t_2}{(t_1, t_2)}\ \mathit{pair}$	$\dfrac{t_1 \quad t_2}{[t_1, t_2]}\ \mathit{bpair}$

Figure: analz and synth rules for normal terms (with assumptions from $X \subseteq \mathscr{T}$)

Modelling blind signatures

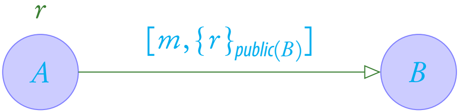
A (a voter) wants to get B (a registration authority) to sign a message m for her, without revealing m to him.

Modelling blind signatures

A (a voter) wants to get B (a registration authority) to sign a message m for her, without revealing m to him. In other words, A wants the message $\{m\}_{private(B)}$.

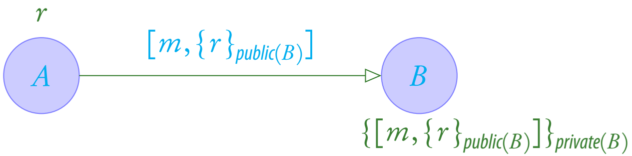
Modelling blind signatures

A (a voter) wants to get B (a registration authority) to sign a message m for her, without revealing m to him. In other words, A wants the message $\{m\}_{private(B)}$.



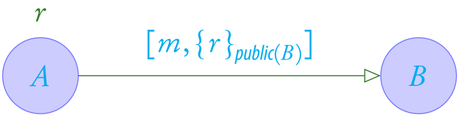
Modelling blind signatures

A (a voter) wants to get B (a registration authority) to sign a message m for her, without revealing m to him. In other words, A wants the message $\{m\}_{private(B)}$.



Modelling blind signatures

A (a voter) wants to get B (a registration authority) to sign a message m for her, without revealing m to him. In other words, A wants the message $\{m\}_{private(B)}$.

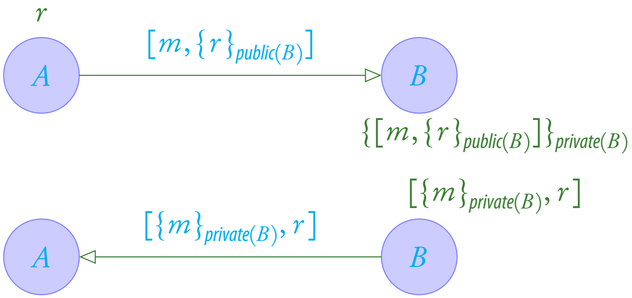


$$\{[m, \{r\}_{public(B)}]\}_{private(B)}$$

$$[\{m\}_{private(B)}, r]$$

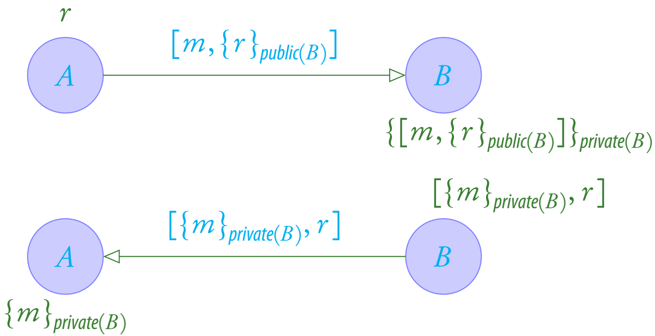
Modelling blind signatures

A (a voter) wants to get B (a registration authority) to sign a message m for her, without revealing m to him. In other words, A wants the message $\{m\}_{private(B)}$.



Modelling blind signatures

A (a voter) wants to get B (a registration authority) to sign a message m for her, without revealing m to him. In other words, A wants the message $\{m\}_{private(B)}$.



Alternative theories

- A simpler system. Delaune, Kremer, Ryan 2009, Baskar, Ramanujam, Suresh 2007.

$$\frac{[t, \{m\}_k] \quad inv(k)}{[\{t\}_{inv(k)}, m]}$$

  Passive intruder deduction is PTIME decidable.

- A much harder system. Lafourcade, Lugiez, Treinen 2007.

$$\frac{t_1 + \cdots + t_\ell \quad k}{\{t_1\}_k + \cdots + \{t_\ell\}_k}$$

$$\frac{t_1 + \cdots + t_\ell + \cdots + t_m \quad t_\ell + \cdots + t_m + \cdots + t_n}{t_1 + \cdots + t_{\ell-1} - t_{m+1} - \cdots - t_n}$$

  Decidable but non-elementary upper bound.

- Our system: Decidable with a DEXPTIME upper bound.

Some difficult proofs

$$\dfrac{\dfrac{}{[a,\{b\}_k]}\,\textit{Ax} \qquad \dfrac{\dfrac{}{b}\,\textit{Ax} \quad \dfrac{}{k}\,\textit{Ax}}{\{b\}_k}\,\textit{encrypt}}{a}\,\textit{blindsplit}_1$$

Some difficult proofs . . .

$$
\dfrac{\dfrac{\dfrac{}{[a, b]}\,\mathit{Ax} \qquad \dfrac{}{k}\,\mathit{Ax}}{[\{a\}_k, \{b\}_k]}\,\mathit{encrypt} \qquad \dfrac{}{\{b\}_k}\,\mathit{Ax}}{\{a\}_k}\,\mathit{blindsplit}_1
$$

Some difficult proofs ...

$$
\dfrac{\dfrac{}{[\{a\}_{k_1k_2}, b]}\,\mathit{Ax} \qquad \dfrac{\dfrac{}{[b, \{a\}_{k_1k_2k_1}]}\,\mathit{Ax} \qquad \dfrac{\dfrac{\dfrac{}{[\{a\}_{k_1k_2}, b]}\,\mathit{Ax} \quad \dfrac{}{k_1}\,\mathit{Ax}}{[\{a\}_{k_1k_2k_1}, \{b\}_{k_1}]}\,\mathit{encrypt} \qquad \dfrac{}{\{b\}_{k_1}}\,\mathit{Ax}}{\{a\}_{k_1k_2k_1}}\,\mathit{blindsplit}_1}{b}\,\mathit{blindsplit}_1}{\{a\}_{k_1k_2}}\,\mathit{blindsplit}_1
$$

Some difficult proofs ...

$$
\dfrac{\dfrac{}{[a,\{a\}_{k_1}]}\,Ax \qquad \dfrac{\dfrac{\dfrac{}{[a,\{a\}_{k_1}]}\,Ax \quad \dfrac{}{k_1}\,Ax}{[\{a\}_{k_1},\{a\}_{k_1 k_1}]}\,encrypt \qquad \dfrac{}{\{a\}_{k_1 k_1}}\,Ax}{\{a\}_{k_1}}\,blindsplit_1}{a}\,blindsplit_1
$$

Decidability: the proof idea

- The examples suggest that it is not easy to come up with a bound on the terms occurring in the proof.
- Instead of trying to prove that it is finite, we prove that it is regular.
  - Show that every term in a normal proof of $X \vdash t$ is of the form $\{p\}_x$ where $p \in st(X \cup \{t\})$ and x is a sequence of keys from $st(X \cup \{t\})$.
  - Show that for each $p \in st(X \cup \{t\})$, $\mathscr{L}_p = \{x \in \mathscr{K}^* | X \vdash \{p\}_x\}$ is a regular set.
  - To check whether $X \vdash t$, check whether $\varepsilon \in \mathscr{L}_t$.

Decidability: the proof idea

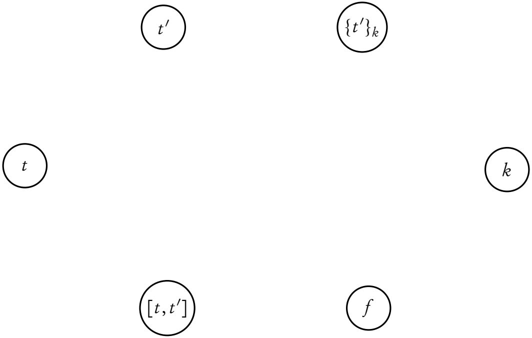
- The examples suggest that it is not easy to come up with a bound on the terms occurring in the proof.
- Instead of trying to prove that it is finite, we prove that it is regular.
  - Show that every term in a normal proof of $X \vdash t$ is of the form $\{p\}_x$ where $p \in st(X \cup \{t\})$ and x is a sequence of keys from $st(X \cup \{t\})$.
  - Show that for each $p \in st(X \cup \{t\})$, $\mathscr{L}_p = \{x \in \mathscr{K}^* | X \vdash \{p\}_x\}$ is a regular set.
  - To check whether $X \vdash t$, check whether $\varepsilon \in \mathscr{L}_t$.
  - Properties of the $\mathscr{L}_p$:
    - $kx \in \mathscr{L}_p$ iff $x \in \mathscr{L}_{\{p\}_k}$
    - if $x \in \mathscr{L}_p$ and $x \in \mathscr{L}_{[p,p']}$, then $x \in \mathscr{L}_{p'}$
    - if $x \in \mathscr{L}_p$ and $\varepsilon \in \mathscr{L}_k$, then $xk \in \mathscr{L}_p$
    - if $\varepsilon \in \{t\}_k$ and $\varepsilon \in inv(k)$ then $\varepsilon \in t$.

Outline

1. Security protocols
2. Dolev-Yao model
3. Extensions of the basic model
4. An automaton construction

An example

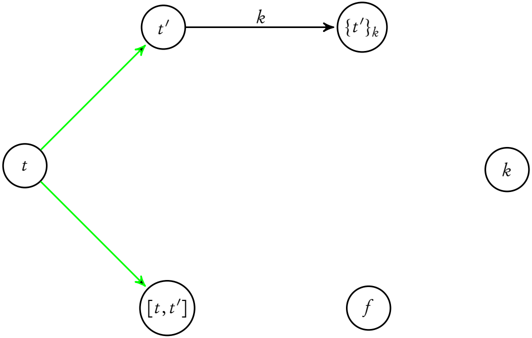
$$\{[t,t'],\{t'\}_k,k\} \vdash \{t\}_k$$



the set of subterms

An example

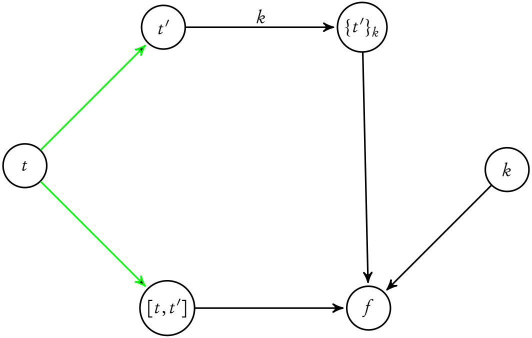
$\{[t,t'],\{t'\}_k,k\} \vdash \{t\}_k$



$t',[t,t'] \vdash t$ and t' encrypted with k is $\{t'\}_k$

An example

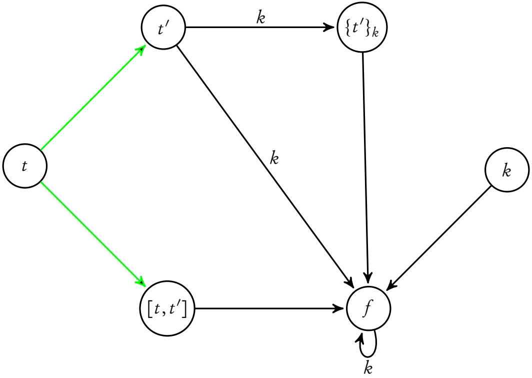
$$\{[t,t'],\{t'\}_k,k\} \vdash \{t\}_k$$



the initial set of terms X

An example

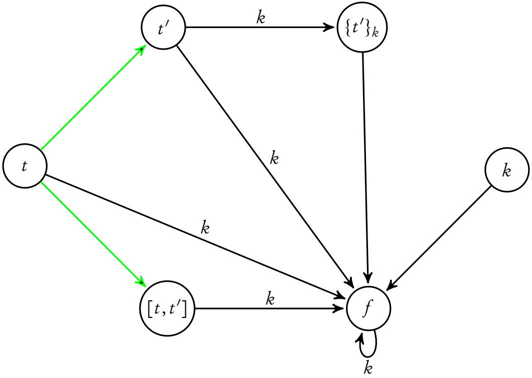
$$\{[t,t'],\{t'\}_k,k\}\vdash\{t\}_k$$



$k \in X$ and $t' \overset{k}{\Rightarrow} f$

An example

$$\{[t,t'],\{t'\}_k,k\} \vdash \{t\}_k$$



$[t,t'] \overset{k}{\Rightarrow} f$ and $t \overset{k}{\Rightarrow} f$

Another example

$$\{[t, \{t'\}_k], t', k\} \vdash t$$

$\{t'\}_k$

t'

t

k

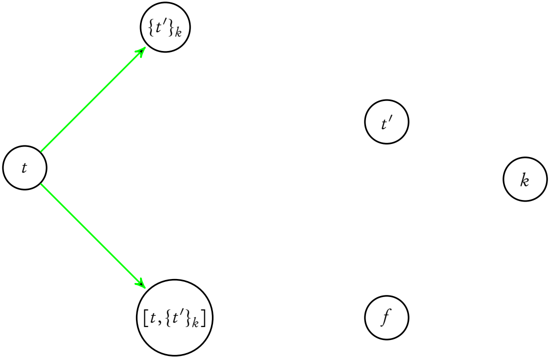
$[t, \{t'\}_k]$

f

the set of subterms

Another example

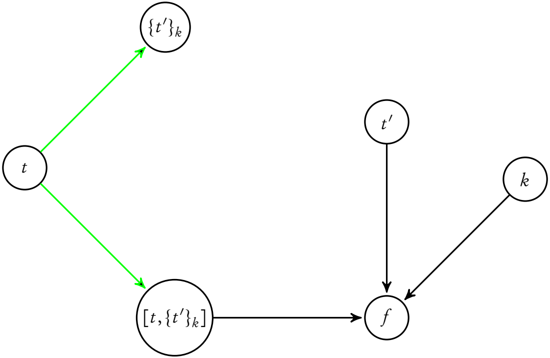
$$\{[t,\{t'\}_k],t',k\} \vdash t$$



$$\{t'\}_k,[t,\{t'\}_k] \vdash t$$

Another example

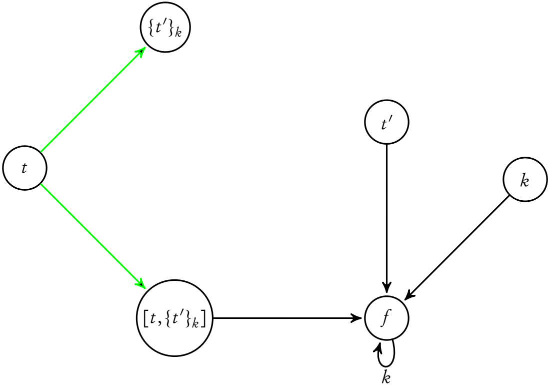
$$\{[t, \{t'\}_k], t', k\} \vdash t$$



the initial set of terms X

Another example

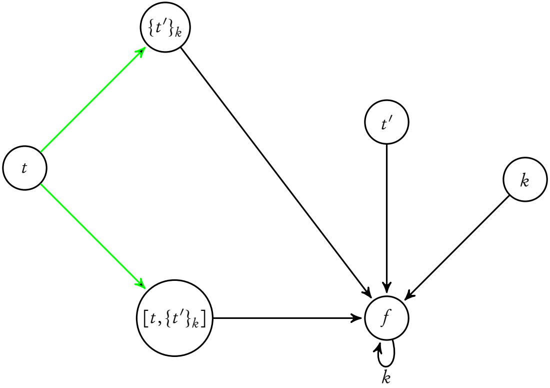
$$\{[t, \{t'\}_k], t', k\} \vdash t$$



$k \in X$

Another example

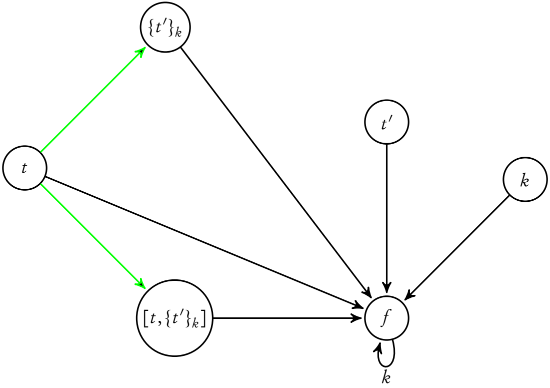
$$\{[t,\{t'\}_k],t',k\} \vdash t$$



$t' \overset{k}{\Rightarrow} f$

Another example

$$\{[t,\{t'\}_k],t',k\} \vdash t$$



$t \Rightarrow f$

Proof normalization

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ t'\end{matrix} \qquad \begin{matrix}\vdots\,\pi'' \\ t''\end{matrix}}{[t', t'']}\,\textit{bpair} \qquad \begin{matrix}\vdots\,\delta \\ k\end{matrix}}{[\{t'\}_k\!\downarrow, \{t''\}_k\!\downarrow]}\,\textit{encrypt}$$

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ t'\end{matrix} \qquad \begin{matrix}\vdots\,\delta \\ k\end{matrix}}{\{t'\}_k\!\downarrow}\,\textit{encrypt} \qquad \dfrac{\begin{matrix}\vdots\,\pi'' \\ t''\end{matrix} \qquad \begin{matrix}\vdots\,\delta \\ k\end{matrix}}{\{t''\}_k\!\downarrow}\,\textit{encrypt}}{[\{t'\}_k\!\downarrow, \{t''\}_k\!\downarrow]}\,\textit{bpair}$$

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ \{t'\}_k\!\downarrow\end{matrix} \qquad \begin{matrix}\vdots\,\pi'' \\ \{t''\}_k\!\downarrow\end{matrix}}{[\{t'\}_k\!\downarrow, \{t''\}_k\!\downarrow]}\,\textit{bpair} \qquad \begin{matrix}\vdots\,\delta \\ \textit{inv}(k)\end{matrix}}{[t', t'']}\,\textit{decrypt}$$

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\pi' \\ \{t'\}_k\!\downarrow\end{matrix} \qquad \begin{matrix}\vdots\,\delta \\ \textit{inv}(k)\end{matrix}}{t'}\,\textit{decrypt} \qquad \dfrac{\begin{matrix}\vdots\,\pi'' \\ \{t''\}_k\!\downarrow\end{matrix} \qquad \begin{matrix}\vdots\,\delta \\ \textit{inv}(k)\end{matrix}}{t''}\,\textit{decrypt}}{[t', t'']}\,\textit{bpair}$$

Figure: The normalization rules I

Proof normalization ...

$$\cfrac{\cfrac{\overset{\vdots\,\pi'}{[t,t']} \quad \overset{\vdots\,\pi''}{t'}}{t}\,\textit{blindsplit} \qquad \overset{\vdots\,\delta}{k}}{\{t\}_k\downarrow}\,\textit{encrypt}$$

$$\cfrac{\cfrac{\overset{\vdots\,\pi'}{[t,t']} \quad \overset{\vdots\,\delta}{k}}{[\{t'\}_k\downarrow,\{t'\}_k\downarrow]}\,\textit{encrypt} \qquad \cfrac{\overset{\vdots\,\pi''}{t'} \quad \overset{\vdots\,\delta}{k}}{\{t'\}_k\downarrow}\,\textit{encrypt}}{\{t\}_k\downarrow}\,\textit{blindsplit}$$

$$\cfrac{\cfrac{\overset{\vdots\,\pi'}{[\{t'\}_k\downarrow,\{t'\}_k\downarrow]} \quad \overset{\vdots\,\pi''}{\{t'\}_k\downarrow}}{\{t\}_k\downarrow}\,\textit{blindsplit} \qquad \overset{\vdots\,\delta}{\mathit{inv}(k)}}{t}\,\textit{decrypt}$$

$$\cfrac{\cfrac{\overset{\vdots\,\pi'}{[\{t'\}_k\downarrow,\{t'\}_k\downarrow]} \quad \overset{\vdots\,\delta}{\mathit{inv}(k)}}{[t,t']}\,\textit{decrypt} \qquad \cfrac{\overset{\vdots\,\pi''}{\{t'\}_k\downarrow} \quad \overset{\vdots\,\delta}{\mathit{inv}(k)}}{t'}\,\textit{decrypt}}{t}\,\textit{blindsplit}$$

Figure: The normalization rules II

Proof normalization …

Lemma

Whenever $X \vdash t$*, there is a normal proof of* t *from* X*.*

Lemma

Whenever $X \vdash t$, there is a normal proof of t from X.

Lemma

Let π be a normal proof of t from X, and let δ be a sub-proof of π with root labelled r. Then the following hold:

1. *If δ ends with an analz rule, then for every u occurring in δ there is $p \in st(X)$ and keyword x such that $u = \{p\}_x\downarrow$.*
2. *If δ ends with a synth rule, then for every u occurring in δ, either $u \in st(X \cup \{r\})$ or there is $p \in st(X)$ and keyword x such that $u = \{p\}_x\downarrow$.*
3. *If the last rule of δ is decrypt or split with major premise r_1, then $r_1 \in st(X)$.*

The automaton construction

Similar to the construction in [Bouajjani, Esparza, Maler 1997]

$$\mathscr{A}_i = (Q, \Sigma, \hookrightarrow_i, F), Q = Y_0 \cup \{f\}, \Sigma = K_0, \text{ and } F = \{f\}.$$

The automaton construction

Similar to the construction in [Bouajjani, Esparza, Maler 1997]

$$\mathscr{A}_i = (Q, \Sigma, \hookrightarrow_i, F), Q = Y_0 \cup \{f\}, \Sigma = K_0, \text{ and } F = \{f\}.$$

- 1. if $t \in Y_0, k \in K_0$ such that $\{t\}_k\!\downarrow \in Y_0$, then $t \overset{k}{\hookrightarrow}_0 \{\{t\}_k\!\downarrow\}$.
  2. if $t, t', t'' \in Y_0$ such that t is the conclusion of an instance of the *bpair* or *blindsplit*$_i$ rules with premises t' and t'', then $t \overset{\varepsilon}{\hookrightarrow}_0 \{t', t''\}$.

The automaton construction

Similar to the construction in [Bouajjani, Esparza, Maler 1997]

$$\mathscr{A}_i = (Q, \Sigma, \hookrightarrow_i, F), Q = Y_0 \cup \{f\}, \Sigma = K_0, \text{ and } F = \{f\}.$$

- 1. if $t \in Y_0, k \in K_0$ such that $\{t\}_k\!\downarrow \in Y_0$, then $t \overset{k}{\hookrightarrow}_0 \{\{t\}_k\!\downarrow\}$.
  2. if $t, t', t'' \in Y_0$ such that t is the conclusion of an instance of the *bpair* or *blindsplit$_i$* rules with premises t' and t'', then $t \overset{\varepsilon}{\hookrightarrow}_0 \{t', t''\}$.

- 1. if $q \overset{a}{\Rightarrow}_i C$, then $q \overset{a}{\hookrightarrow}_{i+1} C$.
  2. if $\{t\}_k\!\downarrow \in Y_0$ and $t \overset{k}{\Rightarrow}_i C$, then $\{t\}_k\!\downarrow \overset{\varepsilon}{\hookrightarrow}_{i+1} C$.
  3. if $k \in K_0$ and $k \overset{\varepsilon}{\Rightarrow}_i \{f\}$, then $f \overset{k}{\hookrightarrow}_{i+1} \{f\}$.
  4. if $\Gamma \subseteq Y_0$, $t \in Y_0$, and if there is an instance r of one of the rules whose set of premises is (exactly) Γ and conclusion is t the following holds:

     if $u \overset{\varepsilon}{\Rightarrow}_i \{f\}$ for every $u \in \Gamma$, then $t \overset{\varepsilon}{\hookrightarrow}_{i+1} \{f\}$.

Correctness of the construction

Theorem

*(**Completeness**) For any* $t \in Y_0$ *and any keyword* x*, if* $X_0 \vdash \{t\}_x\downarrow$*, then there exists* $i \geq 0$ *such that* $t \stackrel{x}{\Rightarrow}_i \{f\}$.

Correctness of the construction

Theorem

(**Completeness**) *For any* $t \in Y_0$ *and any keyword* x*, if* $X_0 \vdash \{t\}_x\downarrow$*, then there exists* $i \geq 0$ *such that* $t \overset{x}{\Rightarrow}_i \{f\}$.

Lemma

Suppose $i, d \geq 0$*,* $t \in Y_0, x, y \in K_0^*$*, and* $C \subseteq Q$ *(with* $D = C \cap Y_0$*). Suppose the following also hold: 1)* $t \overset{x}{\Rightarrow}_{i,d} C$*, and 2)* $C \subseteq Y_0$ *or* $X_0 \vdash y$*. Then* $X_0 \cup \{D\}_y \vdash \{t\}_{xy}$.

Correctness of the construction

Theorem

(**Completeness**) *For any* $t \in Y_0$ *and any keyword* x*, if* $X_0 \vdash \{t\}_x\downarrow$*, then there exists* $i \geq 0$ *such that* $t \overset{x}{\Rightarrow}_i \{f\}$.

Lemma

Suppose $i, d \geq 0$, $t \in Y_0, x, y \in K_0^*$*, and* $C \subseteq Q$ *(with* $D = C \cap Y_0$*). Suppose the following also hold: 1)* $t \overset{x}{\Rightarrow}_{i,d} C$*, and 2)* $C \subseteq Y_0$ *or* $X_0 \vdash y$*. Then* $X_0 \cup \{D\}_y \vdash \{t\}_{xy}$.

Theorem

(**Soundness**) *For any* i*, any* $t \in Y_0$*, and any keyword* x*, if* $t \overset{x}{\Rightarrow}_i \{f\}$*, then* $X_0 \vdash \{t\}_x\downarrow$.

Summary

- Interesting extension of the Dolev-Yao theory

Summary

- Interesting extension of the Dolev-Yao theory
- Related work Tree automata used extensively in security protocol literature. Typically the accepted language is an over-approximation of the set of derivable terms.

Summary

- Interesting extension of the Dolev-Yao theory
- Related work Tree automata used extensively in security protocol literature. Typically the accepted language is an over-approximation of the set of derivable terms. But there is a lot of potential for automata to be used in the analysis of derivations.

Summary

- Interesting extension of the Dolev-Yao theory
- Related work Tree automata used extensively in security protocol literature. Typically the accepted language is an over-approximation of the set of derivable terms. But there is a lot of potential for automata to be used in the analysis of derivations.
- Future work Lots of unresolved questions: Lower bounds or tighter upper bounds, complexity of the active intruder theory, better upper bounds for a general abelian group operator with encryption (the [LLT2007] result) etc.

Thank you!