Lecture 3

Subhashis Banerjee & Aalok Thakkar

Announcements:

Four quizzes: Best three. No make up quizzes. 30%

Two assignments: Both 10% (total 20%) One

midterm: 25%

One final exam: 25%

Verification of Security Protocols

Abstract the protocol into a **formal model** (automata, logic)

Assume perfect cryptography

Specify required guarantees as **mathematical properties** over these abstract models

Prove these properties hold, preferably by automated means

?

Our Formalism: Dolev-Yao Model

On the Security of Public Key Protocols (1983).

$$A \to B : \{(A, \{m\}_B)\}_B$$

$$B \to A : \{(B, \{m\}_A)\}_A$$

$$A!B : \mathrm{aenc}((A, \mathrm{aenc}((m, \mathrm{pk}_B)), \mathrm{pk}_B)$$

$$B? : \mathrm{aenc}((X, \mathrm{aenc}((m', \mathrm{pk}_B)), \mathrm{pk}_B)$$

$$A? : \mathrm{aenc}((B, \mathrm{aenc}((m, \mathrm{pk}_A)), \mathrm{pk}_A)$$

$$B!X : \mathrm{aenc}((B, \mathrm{aenc}((m', \mathrm{pk}_X)), \ldots$$

$pk_X)$

Our Formalism: Dolev-Yao Model

Split

each communication into a send and a receive.

I

The intruder is essentially the network.

I

- Each send captured by

I

- Each receive assumed to come from

A send action need not have a corresponding receive action.

Our Formalism: Dolev-Yao Model

/

Intruder cannot break encryption. It can:

- **See** any message sent on the public channel
- **Block** any message from reaching the intended recipient •

Re-route any message to any principal

• **Masquerade** as any principal and send messages in their name •

Initiate new communication according to the protocol • **Generate**

messages according to some rules

Messages as Term Algebra

Messages are **not** structured documents.

Ignore extraneous details (headers, metadata, formatting)

Formally modelled as symbolic terms

$t := m \mid a \mid \mathrm{pk}(a) \mid \mathrm{sk}(a) \mid \mathrm{pair}(t_1, t_2) \mid \mathrm{aenc}(t, \mathrm{pk}(a))$ Atomic terms m

(messages) and a (agent names).

Proof Rules for Generating Terms

ax ($t \in X$)

$X \vdash a$

pk

t_1) $X \vdash t_i$

split$_i$

$X \vdash \mathrm{pk}(a)$

$X \vdash t$

$X \vdash \mathrm{pair}(t_0,$ $X \vdash t$ $X \vdash t'$

$X \vdash \mathrm{pair}(t, \quad t')$

pair

$\mathrm{pk}(a))$ $X \vdash \mathrm{sk}(a)$ $X \vdash m$ $X \vdash \mathrm{pk}(a)$

aenc

$X \vdash \mathrm{aenc}(m,$ $X \vdash m$ $X \vdash \mathrm{aenc}(m,$

$\mathrm{pk}(a))$

adec

Proof Rules for Generating Terms

$\mathrm{pk}(B)$ $))$, $\mathrm{pk}(I)$ $))$, $\mathrm{pk}(I)$ $)$

$X_I \vdash \mathrm{sk}(I)$ ax $(\mathrm{sk}_I \in X_I)$ adec

$X_I \vdash \mathrm{aenc}\ (\mathrm{pair}\ (B, \mathrm{aenc}\ (\mathrm{pair}\ (A, \mathrm{aenc}\ (m,$

$X_I \vdash \mathrm{pair}\ (B, \mathrm{aenc}\ (\mathrm{pair}\ (A,\ \mathrm{aenc}\ (m, \mathrm{pk}(B)\ )), \mathrm{pk}(I)))$ $X_I \vdash \mathrm{aenc}\ (\mathrm{pair}\ (A, \mathrm{aenc}\ (m,$

pk(B)), pk(I))

$X_I \vdash$ sk(I) ax (sk$_I \in X_I$) adec

split$_1$

pk(B))) split$_1$ $X_I \vdash$ aenc

$X_I \vdash$ pk(B)

$X_I \vdash I$ ax ($I \in X_I$)

(m, pk(B)) pair

$X_I \vdash B$

ax ($B \in X_I$) pk

$X_I \vdash$ pair (A, aenc (m,

$X_I \vdash$ pair (I, aenc (m, pk(B))) aenc $X_I \vdash$ aenc (pair (I, aenc (m, pk(B)), pk(B))

Proof Rules for Generating Terms

Given I's knowledge, can it derive a given term t?

$$\Leftrightarrow$$

Given a deductive proof system for generating terms and a set of known terms X, *does there exist a derivation for term* t?

That is, $X \vdash t$? And can we automate this?

Two Intruder Problems

Passive intruder problem: Can the intruder violate some property just by listening to the network?

Active intruder problem: Can the intruder violate some property by listening to network + allowable behaviours?

Two Intruder Problems

Passive intruder problem: Can the intruder violate some property just by listening to the network?

Fixed X, fixed t. Check if $X \vdash t$.

Active intruder problem: Can the intruder violate some property by listening to network + allowable behaviours?

Two Intruder Problems

Passive intruder problem: Can the intruder violate some

property just by listening to the network?

Fixed X, fixed t. Check if $X \vdash t$.

Active intruder problem: Can the intruder violate some property by listening to network + allowable behaviours?

X

Come up with an and a suitable mapping for X t

$X \vdash t$

variables in and such that we have .

Passive Intruder Problem

Given X and t, check if $X \vdash t$.

check if $X \vdash t$.

Consider

Passive Intruder

Problem Given X and t,

$X = \{\mathrm{aenc}(m, \mathrm{pk}(A)),$
$\mathrm{pair}(\mathrm{sk}(B),$
$\mathrm{aenc}(\mathrm{pair}(n, \mathrm{sk}(A)), \mathrm{pk}(C)))$
$\mathrm{pair}(n, \mathrm{aenc}(\mathrm{sk}(C),$
$\mathrm{pk}(B)))\}$

Consider

Passive Intruder

Problem Given X and t,

check if $X \vdash t$.

$X \vdash m$?

X = {aenc(m, pk(A)),
pair(sk(A),
aenc(pair(n,sk(A)), pk(C)))
pair(n, aenc(sk(C),
pk(B)))}

$X \vdash m$?

Automated Proof Discovery

Consider
Passive Intruder

Problem Given X and t, check if $X \vdash t$.

$X = \{\mathrm{aenc}(m, \mathrm{pk}(A)),$
$\mathrm{pair}(\mathrm{pk}(B), \mathrm{aenc}(\mathrm{pair}(m, \mathrm{pk}(B)), \mathrm{pk}(C)))$
$\mathrm{aenc}(\mathrm{sk}(C), \mathrm{pk}(B))\}$

$X \vdash \mathrm{aenc}(m, \mathrm{pk}(C))$?

Given X and t, check if $X \vdash t$.

X

Rules do not change .

Some rules *construct*, that is give rise to bigger terms: encryption, pairing

Some rules *destruct*, that is give rise to smaller terms: decryption, split

Automated Proof Discovery

What is the size of a term?

X

Rules do not change .

Some rules *construct*, that is give rise to **bigger terms**: encryption, pairing

Some rules *destruct*, that is give rise to **smaller terms**: decryption, split

Size of Terms

$$\text{aenc}\big(\text{pair}\big(B, \text{aenc}\big(\text{pair}\big(A, \text{aenc}\big(m, \text{pk}(B)\big)\big), \text{pk}(I)\big)\big), \text{pk}(I)\big)$$

Treat a term like a tree. Count the number of nodes!

Formally modelled as symbolic terms

$$t := m \mid a \mid \mathrm{pk}(a) \mid \mathrm{sk}(a) \mid \mathrm{pair}(t_1, t_2) \mid \mathrm{aenc}(t, \mathrm{pk}(a))$$

Size of Terms

$$\mathrm{aenc}\left(\mathrm{pair}\left(B, \mathrm{aenc}\left(\mathrm{pair}\left(A, \mathrm{aenc}\left(m, \mathrm{pk}(B)\right)\right), \mathrm{pk}(I)\right)\right), \mathrm{pk}(I)\right)$$

$$\mathrm{size}(t) = \begin{cases} 1 \text{ if } t \text{ is atomic} \\ 1 + \sum \mathrm{size}(t_i) \text{ if } t = f(t_1, t_2, \cdots, t_k) \end{cases}$$

Formally modelled as symbolic terms

$t := m \mid a \mid \mathrm{pk}(a) \mid \mathrm{sk}(a) \mid \mathrm{pair}(t_1, t_2) \mid \mathrm{aenc}(t, \mathrm{pk}(a))$

Abnormal Proofs

$X \vdash m$

$X \vdash \mathrm{pk}(a)$ aenc

$X \vdash t_0 \quad X \vdash t_1$ pair

$X \vdash \mathrm{pair}(t_0, t_1)$ split$_0$ $X \vdash t_0$

$X \vdash \mathrm{aenc}(m, \mathrm{pk}(a)) \quad X \vdash m$ adec

$X \vdash \mathrm{sk}(a)$

One where a *construct rule* is immediately followed by *destruct rule*.

Normal Proofs

One where no *construct rule* is immediately followed by *destruct rule*.

Normal Proofs

A normal proof is one where the major premise of a *destructor rule* is not obtained by the application of a *constructor rule*.

One where no *construct rule* is immediately followed by *destruct rule*.

Normal Proofs

destructor rule constructor rule

$t')$

pk

split$_i$

$X \vdash t$

$X \vdash t \; X \vdash t'$

pair

ax $(t \in X)$

$X \vdash a$

$X \vdash \text{pair}(t_0,$ $X \vdash \text{pair}(t,$

$t_1)$ $X \vdash t_i$

$X \vdash \text{pk}(a)$

$\text{pk}(a))$ $X \vdash \text{sk}(a)$ $X \vdash m$ $X \vdash \text{pk}(a)$

aenc

$X \vdash m$

$X \vdash \text{aenc}(m,$

$X \vdash \text{aenc}(m,$

$\text{pk}(a))$

adec

Normal Proofs

If there exists a proof of $X \vdash t$, then there exists a minimal proof.

The shortest proof π must be normal.

Normal Proofs

If there exists a proof of $X \vdash t$, then there exists a minimal proof.

The shortest proof π must be normal.

Suppose not!

$\xi\ X \vdash u\ \xi$

Then there is a subproof of such that ends in a destructor rule, and ξ

the major premise of is yielded by some constructor rule. We will show how $\xi\ X$

$\vdash u\ \pi$

to replace by a smaller proof of , thus contradicting the minimality of .

Normal Proofs

If there exists a proof of $X \vdash t$, then there exists a minimal proof.

The shortest proof π must be normal.

Case I:

$\text{ax } (t \in X)$ major premise is empty.

$X \vdash t$

Normal Proofs

If there exists a proof of $X \vdash t$, then there exists a minimal proof.

The shortest proof π must be normal.

Case II:

$\pi_0 \; \pi_1$

$$\dfrac{\dfrac{\begin{matrix}\vdots & \vdots\\ X \vdash t_0 & X \vdash t_1\end{matrix}}{X \vdash \mathrm{pair}(t_0, t_1)}\ \mathrm{pair}}{X \vdash t_i}\ \mathrm{split}_i$$

Normal Proofs

If there exists a proof of $X \vdash t$, then there exists a minimal proof.

The shortest proof π must be normal.

Case II:

π_0 π_1

$\vdots$ $\vdots$

$X \vdash t_0$ $X \vdash t$ 1 pair $X \vdash$

pair(t_0, t_1)split$_i$

$X \vdash t_i$

π_i

$\vdots$

$X \vdash t_i$

Normal Proofs

If there exists a proof of $X \vdash t$, then there exists a minimal proof.

The shortest proof π must be normal.

Case III:

π' π''

$\vdots$ $\vdots$

$X \vdash m$

$X \vdash \mathrm{pk}(a)$ aenc

π''' $\vdots$

π' $\vdots$

$\vdots$

$X \vdash \mathrm{aenc}(m, \mathrm{pk}(a))$

$X \vdash m$

adec

$X \vdash \mathrm{sk}(a)$

$X \vdash m$

There exists a proof of if and only if $X \vdash t$ there exists a normal proof of .

normalisation theorem

Normal Proofs

$X \vdash t$

normalisation theorem

Normal Proofs

$X \vdash t$

There exists a proof of if and only if $X \vdash t$

there exists a normal proof of . What is the size of the shortest normal proof?

Subterms

Treat a term like a tree. Subterm is like a subtree.

Formally modelled as symbolic terms

$$t := m \mid a \mid \mathrm{pk}(a) \mid \mathrm{sk}(a) \mid \mathrm{pair}(t_1, t_2) \mid \mathrm{aenc}(t, \mathrm{pk}(a))$$

Subterms

Treat a term like a tree. Subterm is like a subtree.

$$X \subseteq \mathrm{st}(X)$$

$$\mathrm{pair}(t_0, t_1) \in \mathrm{st}(X) \Rightarrow \{t_0, t_1\} \subset \mathrm{st}(X)$$

$$\mathrm{aenc}(m, k) \in \mathrm{st}(X) \Rightarrow \{m, k\} \subset \mathrm{st}(X)$$

Formally modelled as symbolic terms

$$t := m \mid a \mid \mathrm{pk}(a) \mid \mathrm{sk}(a) \mid \mathrm{pair}(t_1, t_2) \mid \mathrm{aenc}(t, \mathrm{pk}(a))$$

Subterms

Treat a term like a tree. Subterm is like a subtree. $\text{aenc}(\text{pair}(B,$

$$\text{aenc}(\text{pair}(A, \text{aenc}(m, \text{pk}(B))), \text{pk}(I))), \text{pk}(I))$$

Subterms

Treat a term like a tree. Subterm is like a subtree.

aenc(pair(*B*, aenc(pair(*A*, aenc(*m*, pk(*B*))), pk(*I*))),

pk(*I*)) pair(*B*, aenc(pair(*A*, aenc(*m*, pk(*B*))), pk(*I*))) pk(*I*)

Subterms

Treat a term like a tree. Subterm is like a subtree.

$\mathrm{aenc}(\mathrm{pair}(B, \mathrm{aenc}(\mathrm{pair}(A, \mathrm{aenc}(m, \mathrm{pk}(B))), \mathrm{pk}(I))),$

$\mathrm{pk}(I))\ \mathrm{pair}(B, \mathrm{aenc}(\mathrm{pair}(A, \mathrm{aenc}(m, \mathrm{pk}(B))), \mathrm{pk}(I)))\ \mathrm{pk}(I)$

$\mathrm{aenc}(\mathrm{pair}(A, \mathrm{aenc}(m, \mathrm{pk}(B))), \mathrm{pk}(I))\ B$

Subterms

Treat a term like a tree. Subterm is like a subtree.

$\mathrm{aenc}(\mathrm{pair}(B, \mathrm{aenc}(\mathrm{pair}(A, \mathrm{aenc}(m, \mathrm{pk}(B))), \mathrm{pk}(I))),$

$\mathrm{pk}(I))$ $\mathrm{pair}(B, \mathrm{aenc}(\mathrm{pair}(A, \mathrm{aenc}(m, \mathrm{pk}(B))), \mathrm{pk}(I)))$ $\mathrm{pk}(I)$

$\mathrm{aenc}(\mathrm{pair}(A, \mathrm{aenc}(m, \mathrm{pk}(B))), \mathrm{pk}(I))$ B A

$\text{pair}(A, \text{aenc}(m, \text{pk}(B)))$ $\text{aenc}(m, \text{pk}(B))$ m $\text{pk}(B)$

Subterms

Treat a term like a tree. Subterm is like a subtree. Is the

number of subterms equal to the size of the term?

Subterms

Treat a term like a tree. Subterm is like a subtree. Is the

number of subterms equal to the size of the term?

$$|\mathrm{st}(X)| \leq \sum_{t \in X} |t|$$

Normal Proofs

$X \vdash t$

There exists a proof of if and only if

$X \vdash t$

there exists a normal proof of .

π $X \vdash t$ $X \vdash u$

Claim: Let be a normal derivation of . If is an $X \vdash t$ $u \in \mathrm{st}(X \cup \{t\})$

intermediate step in , then .

If the last rule is a destruction rule, then $u \in \mathrm{st}(X)$.

$\pi \; X \vdash t \; X \vdash u$

Let be a normal derivation of . If is an $X \vdash t$ $u \in \mathrm{st}(X \cup \{t\})$ intermediate step in , then . If the last rule is a destruction rule, then $u \in \mathrm{st}(X)$.

By Induction:

Subterm Property

then …

If the last term is generated by ax,

Subterm Property

π $X \vdash t$ $X \vdash u$

Let be a normal derivation of . If is an $X \vdash t$ $u \in$

st($X \cup \{t\}$)

intermediate step in , then .

If the last rule is a destruction rule, then $u \in$

$\mathrm{st}(X)$. By Induction: If the last term is generated by pair, then …

Subterm Property

π $X \vdash t$ $X \vdash u$

Let be a normal derivation of . If is an $X \vdash t$ $u \in$

$\mathrm{st}(X \cup \{t\})$

intermediate step in , then .

If the last rule is a destruction rule, then $u \in \mathrm{st}(X)$.

By Induction:

If the last term is generated by pair, then …

$$\frac{\begin{array}{cc}\pi_0 & \pi_1 \\ \vdots & \vdots \\ X \vdash t_0 & X \vdash t_1\end{array}}{X \vdash \mathrm{pair}(t_0, t_1)}\ \mathrm{pair}$$

Subterm Property

If $X \vdash u$ is an intermediate step…

Either it is in π or or . $_0$ π_1 u = pair(t_0, t_1)

Therefore $u \in \mathrm{st}(X \cup \{t_0\}) \cup \mathrm{st}(X \cup \{t_0\}) \cup \{\mathrm{pair}(t_0, t_1)\}$

π_0 π_1

⋮ ⋮

$X \vdash t_0$ $X \vdash t$ 1 pair

$X \vdash \mathrm{pair}(t_0, t_1)$

Subterm Property

If $X \vdash u$ is an intermediate step…

Either it is in π or or . $_0$ π_1 u = pair(t_0,

t_1)

Therefore $u \in$ st($X \cup$ pair(t_0, t_1)}

π_0 π_1

⋮ ⋮

$$\frac{X \vdash t_0 \quad X \vdash t_1}{X \vdash \text{pair}(t_0, t_1)} \text{ pair}$$

Subterm Property

Let π be a normal derivation of $X \vdash t$. If $X \vdash u$ is an intermediate step in π, then $u \in \text{st}(X \cup \{t\})$.

If the last rule is a destruction rule, then $u \in \text{st}(X)$.

By Induction: If the last term is generated by encryption, then …

$\pi' \; \pi''$

$\vdots \; \vdots$

$X \vdash m$ pk(a) aenc

$X \vdash$

$X \vdash$ aenc(m, pk(a))

Subterm Property

$\pi \; X \vdash t \; X \vdash u$

Let be a normal derivation of . If is an $X \vdash t \; u \in$

$\mathrm{st}(X \cup \{t\})$

intermediate step in , then .

If the last rule is a destruction rule, then $u \in \mathrm{st}(X)$.

By Induction:

If the last term is generated by split, then … π ⋮

$X \vdash \mathrm{pair}(t_0, t_1)$ split_i

$X \vdash t_i$

Subterm Property

π $X \vdash t$ $X \vdash u$

Let be a normal derivation of . If is an $X \vdash t$ $u \in$

$\mathrm{st}(X \cup \{t\})$

intermediate step in , then .

If the last rule is a destruction rule, then $u \in \mathrm{st}(X)$.

By Induction: If the last term is generated by decryption, then …

$$\pi' \; \vdots \quad \pi'' \; \vdots$$

$$X \vdash \mathrm{aenc}(m, \mathrm{pk}(a)) \quad \text{adec}$$

$$X \vdash \mathrm{sk}(a)$$

$X \vdash m$

Normalisation + Subterm Property

$X \vdash t$

If there exists a proof of , if and only if there $X \vdash t$

exists a normal proof of .

π $X \vdash t$ $X \vdash u$

Let be a normal derivation of . If is an $X \vdash t$ u $\in$

$\mathrm{st}(X \cup \{t\})$

intermediate step in , then . If the last rule is a destruction rule, then $u \in \mathrm{st}(X)$.

Normalisation + Subterm Property

$X \vdash t$

There exists a proof of if and only if there $X \vdash t$ exists a normal proof of , with each branch $|\mathrm{st}(X \cup \{t\})|$

bounded by .

Normalisation + Subterm Property

$X \vdash t$

There exists a proof of if and only if there $X \vdash t$

exists a normal proof of , with each branch $X \cup$

$\{t\}$

bounded by the size of , that is

$|t| + \sum_{t' \in X} |t'|$.

Naive Algorithm:

Let $N = |t| + \sum_{t' \in X} |t'|$.

If you have X, you can build

$X' = \{\text{terms generated from } X \text{ in one step}\}$ How?

Naive Algorithm:

Let $N = |t| + \sum$.

$|t'|$

$t' \in X$

If you have X, you can build

$X' = \{$terms generated from X in one step$\}$ How? $X' = \{t :$

$\exists\, \text{pair}(t, \_) \in X \vee \text{pair}(\_, t,) \in X\}$

Naive Algorithm:

Let $N = |t|$ + $\sum$.

$|t'|$

$t' \in X$

If you have X, you can build

$X' = \{$terms generated from X in one step$\}$ How? $X' = \{t :$

$\exists\, \mathrm{pair}(t, \_) \in X \vee \mathrm{pair}(\_, t,) \in X\} \cup \{\mathrm{pair}(t, t') : t, t' \in X\}$

Naive Algorithm:

Let $N = |t| + \sum$.

Naive Algorithm:

If you have X, you can build

X' = {terms generated from X in one step} How?

$$X' = \{t : \exists\, \text{pair}(t, \_) \in X \lor$$

pair(_, t,) $\in X$}

$\cup$ {pair(t, t') : $t, t' \in X$}

$$\cup\ \{m : \text{sk}(A) \in X \land \text{enc}(m, \text{pk}(A)) \in X\}$$

$$|t'|$$

$$t' \in X$$

Let $N = |t| + \sum_{t' \in X} |t'|$.

If you have X, you can build

$X' = \{$terms generated from X in one step$\}$ How?

$X' = \{t : \exists \, \text{pair}(t, \_) \in X \vee \text{pair}(\_, t,) \in X\}$

$\cup \, \{\text{pair}(t, t') : t, t' \in X\}$

$\cup \, \{m : \text{sk}(A) \in X \wedge \text{enc}(m, \text{pk}(A)) \in X\}$

$\cup \{\mathrm{enc}(m, \mathrm{pk}(A)) : m \in X, \mathrm{pk}(A) \in X\} \cup \{\mathrm{pk}(A) : A \in X\}$

Naive Algorithm:

Let $N = |t| + \sum_{t' \in X} |t'|$.

If you have X, you can build

$X' = \{$terms generated from X in one step$\}$

In time $O(|X|^2)$.

Naive Algorithm:

Let $N = |t| + \sum_{t' \in X} |t'|$.

If you have X, you can build

$X' = \{$terms generated from X in one step$\}$

In time $O(N^2)$.

Naive Algorithm:

Let $N = |t| + \sum_{t' \in X} |t'|$.

If you have X, you can build

$X' = \{$terms generated from X in one step$\}$

In time $O(N^2)$.

Repeat this N times.

Naive Algorithm:

Let $N = |t|$ $+$ $\sum$.

$|t'|$

$t' \in X$

$O(N^3)$

In time , construct the set of all N

terms derivable in at most steps. t

derivable in N steps $\Leftrightarrow X \vdash t$.

Let $N = |t| + \sum_{t' \in X} |t'|$.

Inputs: X, t

Can you do it in linear time?

Naive Algorithm:

For i in range $(0, N)$,

Construct X' in $O(N^2)$

Check $t \in X$!

Set $X \leftarrow X'$

Passive Intruder Problem:

Passive intruder problem: Can the intruder violate some property just by listening to the network?

Fixed X, fixed t. Check if $X \vdash t$.

Decidable in polynomial time given certain operators and term algebra.

Passive Intruder Problem:

Passive intruder problem: Can the intruder violate some property just by listening to the network?

Fixed X, fixed t. Check if $X \vdash t$.

Decidable in polynomial time given certain operators and term algebra.

t := m | a | pk(a) | sk(a) | pair(t_1, t_2) | aenc(t, pk(a)) k | senc(t, k) | sign(t,sk(a)) | hash(t)

Active Intruder Problem:

Active intruder problem: Can the intruder violate some property by listening to network + allowable behaviours?

$$A \rightarrow : (A, \mathrm{aenc}(m, \mathrm{pk}_B))$$

$$I \rightarrow B : (I, \mathrm{aenc}(m, \mathrm{pk}_B))$$

$$B \rightarrow I : (B, \mathrm{aenc}(m, \mathrm{pk}_I))$$

$$\rightarrow A : (B, \mathrm{aenc}(m, \mathrm{pk}_A))$$

Active Intruder Problem:

M

- Take an arbitrary Turing machine
- Encode its configurations (state, tape, head position) as symbolic messages
- Design a protocol such that:

M

  - Each valid protocol step corresponds to one transition of . M
  - The intruder can drive the protocol forward iff makes a valid transition

- Define an insecure state iff M reaches a halting configuration.

Formal Verification Tools

solver.

DY-based bounded Active
Intruder detection.

Relational modelling with a SAT

ProVerif Tamarin Alloy 6

What all can you verify?

As models:

Dolev–Yao adversary

• Full network control (intercept, replay, modify, inject) • Perfect cryptography (no guessing / no breaking primitives) Multiple concurrent protocol sessions

Compromised principals (key reveal, corruption models)

What all can you verify?

As specifications:

Secrecy/Confidentiality: Message secrecy, key secrecy

Authentication: Aliveness, weak agreement

Protocol Correctness: Message origin authenticity, session binding, freshness guarantees, replay resistance

Equivalence / Privacy Properties: anonymity, unlinkability, observational equivalence,

What all can you NOT verify?

Cryptographic Strength & Computation

Implementation & Deployments

Real-time guarantees

Assumptions

What all can you NOT verify?

Cryptographic Strength & Computation

Implementation & Deployments

Real-time guarantees

Assumptions

Polynomial-time/probablistic adversaries

Quantitative Properties

User Intent or Semantic Meaning

Correctness of modelling or specifications.