Lecture 2

Subhashis Banerjee & Aalok Thakkar

Announcements:

1. There will be a test on Monday, Feb 2.

   • Lecture 1 (cryptographic primitives)

- Lecture 2 (DY model, attack traces, term derivation)

2. Content for this module is not easily available online.
   Take detailed notes.

3. There will be a programming assignment based on
   passive intruder detection.

4. Proofs You Can Lean On: Verified Proofs and
   Programs with Lean (tomorrow's colloqium)

Poly Network Hack

August 10, 2021

611 million USD or 4600 crores INR

Poly Network is an interoperability protocol that lets users trade one cryptocurrency for another, such as trading Bitcoin for Ethereum.

wner of...

ainData

Who can approve?

In 2017, Abbott implantable cardiac devices communicated wirelessly with home monitoring units.

But the devices did not properly authenticate or verify who they were talking to. An attacker within radio range could:

- Send unauthorized commands

- Drain the device battery
- Alter pacing behavior

The system assumed: “If the message decrypts correctly, it must be legitimate.”

The Four Failure Modes of Information Security

The underlying mathematical assumption is wrong or becomes wrong.

The lock design itself is broken.

The Four Failure Modes of Information Security

The underlying mathematical assumption is wrong or becomes wrong.

CS-3621

The lock design itself is broken.

SHA-1 protocol was *believed* to be collision-resistant. In 2017, Google and CWI Amsterdam demonstrated a practical, real collision (“SHAttered”).

The Four Failure Modes of Information Security

The underlying mathematical

assumption is wrong or becomes wrong. *The lock design itself is broken.*

SHA-1 protocol was *believed* to be collision-resistant. In 2017, Google and CWI Amsterdam demonstrated a practical, real collision (“SHAttered”).

Maybe one day a large fault-tolerant quantum computer will break RSA.

The Four Failure Modes of Information Security

CS-3810

Buffer overflows, side channels, incorrect randomness, timing leaks, nsafe defaults

The cryptography is correct, but the code or hardware is not.

Reusing keys is an engineering failure.

The lock is fine, but it was built poorly.

The Four Failure Modes of Information Security

The rules of interaction are flawed.

The lock works, but you gave the key to the wrong person.

The Four Failure Modes of Information Security

The rules of interaction are flawed.

The lock works, but you gave the key to the wrong person.

Poly Network hack and Abott pacemaker are examples of protocol failures.

The Four Failure Modes of Information Security

The system’s security assumptions about the world are wrong.

“A lot of our processes depended on the assumption that if you are holding your phone in your hand and you know the password, then we can be sure of your identity”

The Four Failure Modes of Information Security

The system’s security assumptions about the world are

wrong.

“A lot of our processes depended on the assumption that if you are holding your phone in your hand and you know the password, then we can be sure of your identity”

CS-2384: Digitalization and Privacy

Lecture 1, slide 12+.

Protocol Failures and How to Prevent Them

Read the Protocol Properly.

Protocol Failures and How to Prevent Them Read the Protocol

Properly.

Insecure!

How to Prevent Them

Testing?

Slightly

Better

Protocol Failures and How to Prevent Them

ated Testing?

Much

Better!

“Program testing can be used to show the presence of bugs, but never to show their absence!” — Edsger W. Dijkstra

How to *prove* the *absence* of bugs?

Correctness

Guarantees!

How to *prove* the *absence* of bugs? Formal Verification.

Make an abstract mathematical **model** of system (and ignore *irrelevant* details)

Cast any desirable property as a mathematical **formula**

Prove or disprove that said **formula**

holds of said **model**

Correctness Guarantees!

What do we want to verify?

Security Protocols!

Sequence of message exchanges to achieve some desirable goal

Built upon various **cryptographic schemes** used for manipulating information with some guarantees

These schemes can be assumed to be *perfect*.

A Simple Example

On a public network, two people share a number m, which they want kept secret.

$$A \rightarrow B : m \; B \rightarrow A : A \; B$$

m

Send the message back to confirm B has recieved.

Is this protocol *secure*? If and finish executing this I m protocol, can a malicious intruder get to know ?

A Simple Example

$A \rightarrow B : m \; B \rightarrow A : m$

The network is public; obviously should not send .

m

cryptographic mechanisms like encryption.

Need to ensure *secrecy* via

Assumptions

A B and are *honest principals*: assumed to not compromise the protocol.

I To honest principals, is just any other entity on the network. They cannot identify or distinguish malicious traffic from benign traffic. All decisions must be made solely from messages, not from beliefs about identity.

If a message of the wrong format is received, or none received at all?

Securing the Example Protocol

How to build a secure protocol?

$$A \rightarrow B : m$$

$$B \rightarrow A : m$$

Securing the Example Protocol

How to build a secure protocol?

$$A \rightarrow B : \text{enc}(m, k)$$

$$B \rightarrow A : m$$

What is encryption? What is k?

Symmetric Encryption

A, B

A pair of participants have a shared key:

$k_{\{A,B\}}$

A **shared key** ( )

A can encrypt a message with k and send it to . $_{\{A,B\}}$ B

B can decrypt the message with k . $_{\{A,B\}}$

$$\text{sdec}\left(\text{senc}(m, k), k'\right) = m \Leftrightarrow k' = k$$

Lecture 1, slide 6.

Securing the Example Protocol

How to build a secure protocol?

$$A \rightarrow B : \text{senc}_{(m,\, k}^{\{A,B\}})$$

$$B \rightarrow A : \text{senc}_{(m,\, k}^{\{A,B\}})$$

Securing the Example Protocol

How to build a secure protocol?

$$A \rightarrow B : \text{senc}_{(m,\, k^{\{A,B\}})}$$

$$B \rightarrow A : \text{senc}_{(m,\, k^{\{A,B\}})}$$

Two problems:

1. How does A know that the message was from B?

Securing the Example Protocol

How to build a secure protocol?

$$A \rightarrow B : \ (A, \text{senc}_{(m,\, k^{\{A,B\}})})$$

$$B \rightarrow A : \ (B, \text{senc}_{(m,\, k^{\{A,B\}})})$$

Two problems:

1. How does A know that the message was from B?

Securing the Example

protocol?

Protocol How to build a secure $A \rightarrow B : \ (A, \text{senc}_{(m,}$

$$k^{\{A,B\}}))$$

$$B \rightarrow A : \; B, \text{senc}(\; (m, k^{\{A,B\}}))$$

Two problems:

If they can share the keys, they can use the same protocol to share a message.

A B

1. How does know that the message was from ? 2. How do they share keys over a public channel?

Asymmetric Encryption

Example: RSA, Elgamal, ECC

A

Each participant has two keys:

pk_A

A **public key** ( ) : publicly known

pk^{-1}

A **private key** ( ) : kept secret

A

Perfect cryptography assumption:

Messages k encrypted with a public key can only be decrypted k^{-1} with the corresponding private key .

$$\mathrm{adec}\left(\mathrm{aenc}(m, k), k'\right) = m \Leftrightarrow$$

$$k' = k^{-1}$$

Asymmetric Encryption

Example: RSA, Elgamal, ECC

A

Each participant has two keys:

pk_A

A **public key** ( ) : publicly known

sk_A

A **secret key** ( ) : kept secret

Perfect cryptography assumption:

Messages k encrypted with a public key can only be decrypted k^{-1} with the corresponding private key .

$$\mathrm{adec}_{(\mathrm{aenc}(m,\ \mathrm{pk}_A),\mathrm{sk}_A)} = m$$

Lecture 1, slide 7.

Securing the Example Protocol

How to build a secure protocol?

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

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

Securing the Example Protocol

How to build a secure protocol?

$A \rightarrow B$: aenc(m, pk_B) $B \rightarrow A$: aenc(m, pk_A)

But how does B know pk_A ?

H

Additional assumption: Given pk_H, anyone in the channel can look up (but not vice versa).

Securing the Example Protocol

How to build a secure protocol?

$A \rightarrow B$: aenc(m, pk_B) $B \rightarrow A$: aenc(m, pk_A)

B
What if gets multiple messages?

Say B recieved $\mathrm{enc}(m, \mathrm{pk}$ and . Which one is from ? $_B)$ $\mathrm{enc}(m', \mathrm{pk}_B)$ A

Securing the Example Protocol

How to build a secure protocol?

$$A \rightarrow B : \; {}_{(}A, \mathrm{aenc}(m, \mathrm{pk}_B^{)})$$

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

Securing the Example Protocol

How to build a secure protocol?

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

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

A B

Is this protocol *secure*? If and finish executing this I m protocol, can a malicious intruder get to know ?

Securing the Example Protocol

Consider the following run of the protocol:

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

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

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

Securing the Example Protocol

Consider the following run of the protocol:

$$A \rightarrow B : \; (A, \text{aenc}(m, \text{pk}_B))$$

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

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

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

Securing the Example Protocol

Consider the following run of the protocol:

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

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

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

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

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

Securing the Example Protocol

Consider the following run of the protocol:

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

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

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

$I \rightarrow B : {}_{(}I, \text{aenc}(m, \text{pk}_B)_{)}$ $\rightarrow A : (B, \text{aenc}(m, \text{pk}_A))$

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

Securing the Example Protocol

Consider the following run of the protocol:

$A \rightarrow B : {}_{(}A, \text{aenc}(m, \text{pk}_B)_{)}$

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

$B \rightarrow A : {}_{(}B, \text{aenc}(m, \text{pk}_A)_{))}$ $I \rightarrow B : {}_{(}I, \text{aenc}(m, \text{pk}_B)_{)}$

Intruder (man) in the middle attack!

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

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

Securing the Example Protocol

Quick Fix?

$$A \rightarrow B : \text{aenc}_{((A, \text{aenc}(m, \text{pk}_B)),}$$

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

Securing the Example Protocol

Simplifying the notation

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

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

Securing the Example Protocol

Consider the following run of the protocol:

$$A \rightarrow : \underset{\{(A,\ \{m\}_B)\}}{B} \quad A \rightarrow B : \underset{\{(A,\ \{m\}_B)\}_B}{} \quad B \rightarrow A$$

$$: \underset{\{(B,\ \{m\}_A)\}_A}{}$$

Securing the Example Protocol

Consider the following run of the protocol:

$$A \rightarrow B : \underset{\{(A,\ \{m\}_B)\}_B}{} \quad B \rightarrow A : \underset{\{(B,\ \{m\}_A)\}_A}{} \quad A \rightarrow : \underset{\{(A,\ \{m\}_B)\}_B}{}$$

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

Securing the Example Protocol

Consider the following run of the protocol:

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

$\{(B, \{m\}_A)\}_A$

$B \rightarrow A : \ m' = (A, \{m\}_B)$

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

$I \rightarrow B : \{I, \{m'\}_B\}_B$

Securing the Example Protocol

Consider the following run of the protocol:

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

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

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

$I \rightarrow B : \{I, \{m'\}_B\}_B$

$B \rightarrow I : \{(B, \{m'\}_I)\}_I$

$m' = (A, \{m\}_B)$

Securing the Example Protocol

Consider the following run of the protocol:

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

$$A \rightarrow : \quad \{(A, \{m\}_B)\}_B$$

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

$$I \rightarrow B : \quad \{I, \{m'\}_B\}_B$$

$$B \rightarrow I : \quad \{(B, \{m'\}_I)\}_I$$

$$m' = \quad (A, \{m\}_B)$$

$$I \rightarrow B : \quad \{(I, \{m\}_B)\}_B$$

Securing the Example Protocol

Consider the following run of the protocol:

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

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

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

$m' = (A, \{m\}_B)$

$I \rightarrow B : \{I, \{m'\}_B\}_B$

$B \rightarrow I : \{(B, \{m'\}_I)\}_I$

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

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

Securing the Example Protocol

Consider the following run of the protocol: A

$$\rightarrow : \quad {}_{\{(A,\ \{m\}_B)\}_B}$$

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

$$B \rightarrow A : \quad {}_{\{(B,\ \{m\}_A)\}_A} \qquad I \rightarrow B : \quad {}_{\{I,\ \{m'\}_B\}_B} \qquad B \rightarrow I : \quad {}_{\{(B,\ \{m'\}_I)\}_I}$$

$$m' = \quad {}_{(A,\ \{m\}_B)} \qquad I \rightarrow B : \quad {}_{\{(I,\ \{m\}_B)\}_B} \qquad B \rightarrow I : \quad {}_{\{(B,\ \{m\}_I)\}_I}$$

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

"Security protocols are three-line programs that people still manage to get

wrong”

Roger Needham

Verification of Security Protocols

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

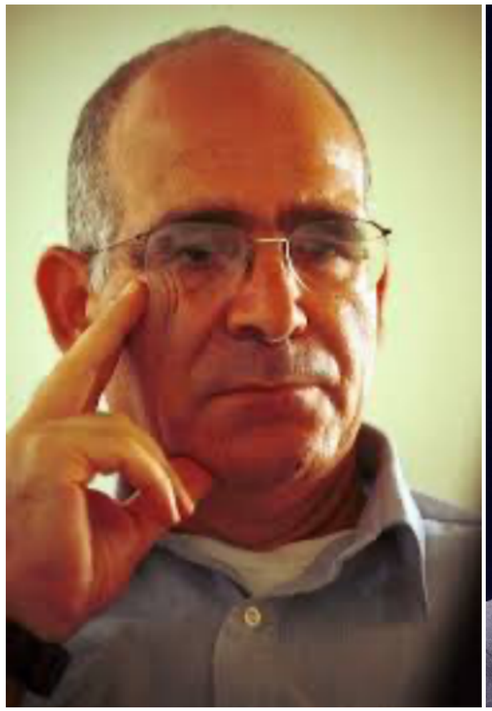
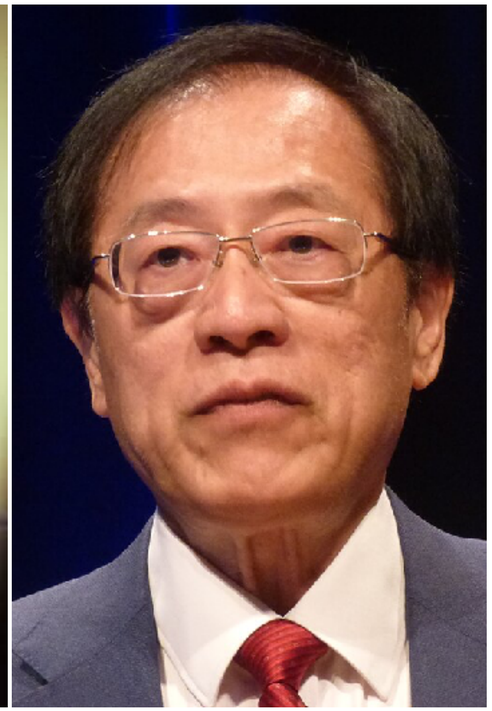
Assume perfect cryptography

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

Prove these properties hold, preferably by automated means

?

Our Formalism: Dolev-Yao Model

Daniel Dolev Andrew Yao

On the Security of Public Key Protocols (1983).

Our Formalism: Dolev-Yao Model

Split each communication into a send and a receive.

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

Our Formalism: Dolev-Yao Model

Split each communication into a send and a receive.

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

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

Our Formalism: Dolev-Yao Model

Split each communication into a send and a receive.

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

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

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

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

Our Formalism: Dolev-Yao Model

Split each communication into a send and a receive.

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

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

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

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

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

Our Formalism: Dolev-Yao Model

Split each communication into a send and a receive.

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

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

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

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

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

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

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

/

• 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

$t := m \mid f(t_1, \ldots, t_k)$ Term algebra.

Proof Rules for Generating Terms

When can agents/intruder generate a particular message term t?

Depends on what they know.

Proof Rules for Generating Terms

When can agents/intruder generate a particular message term t?

Depends on what they know.

Let X denote the set of all terms that an agent knows. Let

"$X \vdash t$" denote that the agent can generate a term t.

Proof Rules for Generating Terms

When can agents/intruder generate a particular message term t?

Depends on what they know.

If the agent knows term t, it can generate t.

$$\frac{}{X \vdash t}\ \text{ax}\ (t \in X)$$

Proof Rules for Generating Terms

When can agents/intruder generate a particular message term t?

Depends on what they know.

If the agent can generate the name of an agent A, it can generate its public key pk_A.

$$X \vdash A \quad \mathsf{pk}$$

$$X \vdash \mathsf{pk}_A$$

Proof Rules for Generating Terms

When can agents/intruder generate a particular message term t?

Depends on what they know.

If the agent can generate a pair, it can generate the individual elements.

$$X \vdash (t_0, t_1) \xrightarrow{\text{split}_i} X \vdash t_i$$

Proof Rules for Generating Terms

When can agents/intruder generate a particular message term t?

Depends on what they know.

If the agent can generate individual elements, it can generate the pair.

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

Proof Rules for Generating Terms

When can agents/intruder generate a particular message term t?

Depends on what they know.

If the agent can generate a public key k and a term m, it can generate $\mathrm{aenc}(m, k)$ encrypted with k.

$$\frac{X \vdash m \quad X \vdash k}{X \vdash \mathrm{aenc}(m, k)}\ \mathrm{aenc}$$

Proof Rules for Generating Terms

When can agents/intruder generate a particular message term t?

Depends on what they know.

k^{-1}

If the agent can generate a private key and

$\mathrm{aenc}(m, k)$ m

an encrypted term , it can generate .

$X \vdash \mathrm{aenc}(m, k)$ $\quad$ adec $X \vdash k^{-1}$

$X \vdash m$

Proof Rules for Generating Terms

$\mathrm{ax}\ (t \in X)$ $\quad$ pk $\quad$ $X \vdash t_i$

$X \vdash A$

$X \vdash \mathrm{pk}_A$ $\quad$ $X \vdash (t_0, t_1)$ $\quad$ split_i

$X \vdash t$

$X \vdash t \; X \vdash t' \;\; X \vdash (t, t')$ pair

$k) \; X \vdash k^{-1} \; X \vdash$ adec $\vdash \text{aenc}(m, k)$

$X \vdash \text{aenc}(m, \quad m \qquad X \vdash m \; X \vdash k \; X$ aenc

Proof Rules for Generating Terms

$m' = \quad (A, \{m\}_B)$

I

How do we know can generate

$A \rightarrow :$

$\{(A, \{m\}_B)\}_B$

aenc I, aenc

$((\quad\quad (m, \text{pk}_B)), \text{pk}_B)$

?

$I \rightarrow B : {}_{\{I, \{m'\}_B\}_B}\, B$

$\rightarrow I : {}_{\{(B, \{m'\}_I)\}_I}\, I$

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

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

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

Proof Rules for Generating Terms

$m' = {}_{(A, \{m\}_B)}$

I

How do we know can generate

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

$\text{aenc}_{((}I, \text{aenc}_{(}(m, \text{pk}_B)), \text{pk}_B)$?

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

$I \rightarrow B :$ $\{I, \{m'\}_B\}_B$ B

$\rightarrow I :$ $\{(B, \{m'\}_I)\}_I$ I

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

What does I know?

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

Proof Rules for Generating Terms

$m' = (A, \{m\}_B)$

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

I

How do we know I can generate

$\text{aenc}((I, \text{aenc}((m, \text{pk}_B)), \text{pk}_B))$?

$I \rightarrow B : {}_{\{I, \{m'\}_B\}_B}\, B$

$\rightarrow I : {}_{\{(B, \{m'\}_I)\}_I}\, I$

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

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

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

What does I know?

$\{(B, \{m'\}_I)\}_I$

Proof Rules for Generating Terms

$m' = {}_{(A, \{m\}_B)}$

I

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

How do we know can generate

$\text{aenc}(I, \text{aenc}((m, \text{pk}_B), \text{pk}_B))$?

$I \rightarrow B : \{I, \{m'\}_B\}_B$ B

$\rightarrow I : \{(B, \{m'\}_I)\}_I$ I

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

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

What does I know? $\{(B, \{(A, \{m\}_B)\}_I)\}_I$

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

Proof Rules for Generating Terms

$m' = (A, \{m\}_B)$

I

How do we know I can generate

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

$\text{aenc}((I, \text{aenc}((m, \text{pk}_B)), \text{pk}_B))$?

$I \rightarrow B : \{I, \{m'\}_B\}_B$ B

$\rightarrow I : \{(B, \{m'\}_I)\}_I$ $I \rightarrow$

B : $\{(I, \{m\}_B)\}_B$ $B \rightarrow$ B, $(\{(A, \{m\}_B)\}_I)$

I : $\{(B, \{m\}_I)\}_I$

What does I know?

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

Proof Rules for Generating Terms

$m' = (A, \{m\}_B)$

I

How do we know can generate

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

$$\text{aenc}((I, \text{aenc}((m, \text{pk}_B)), \text{pk}_B))?$$

$$I \rightarrow B : \{I, \{m'\}_B\}_B$$

$$\rightarrow I : \{(B, \{m'\}_I)\}_I$$

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

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

What does I know?

$$\{(A, \{m\}_B)\}_I$$

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

Proof Rules for Generating Terms

$m' = (A, \{m\}_B)$

I

How do we know I can generate $\text{aenc}((\text{aenc}((m, \text{pk}_B)), \text{pk}_B))$?

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

$I \rightarrow B : \{I, \{m'\}_B\}_B$ B

$\rightarrow I : \{(B, \{m'\}_I)\}_I$ I

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

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

What does I know?

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