

SPi Calculus: Outline

- What is it?
- Basic SPi Calculus Notation
- Basic Example
- Example with Channel Establishment
- Example using Cryptography

SPi Calculus: What is it?

- SPi Calculus is an executable model for the description and analysis of cryptographic protocols
- Spi Calculus is an extension to Pi Calculus

SPi Calculus: Processes

Spi Calculus is made up of **Processes**. When all the processes are combined we have a program or protocol.

SPi Calculus: Processes

Processes are defined as follows:

$P \triangleq$ (some action our process does)

Where P is our process

SPi Calculus: Processes

Processes can do many things

- They can create other processes
- They can send messages
- They can receive messages
- They can run other processes

You can think of a process as the set of actions a principal takes (Alice, Bob, Malory etc.)

SPi Calculus: Processes

When processes send or receive messages
they do this over channels

SPi Calculus: Basic Definitions

Channels

- A channel is a named communications medium
- Channels can be restricted so that only certain processes can communicate on them

SPi Calculus: Basic Definitions

Channel Example:



Process A communicates to Process B
through Channel_{AB}

SPi Calculus: Basic Definitions

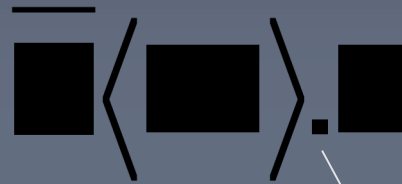
Unfortunately we can't just say:

Process $A \triangleq$ Listen on Channel AB for a
Message M

We have to use SPi Calculus Notation

Pi Calc: Basic Notation (1)

Process Grammar-Output



Sequential
Operator

The above is how we state “Output the message M on Channel C and then run process P”

Pi Calc: Basic Notation (2)

Process Grammar-Input

$$C(x).P$$

Input the message x on the channel C and then run process P (P will have access to x)

Pi Calc: Basic Notation (3)

Process Grammar-Composition

$P | Q$

- A composition $P | Q$ behaves as processes P and Q running in parallel. Each may interact with the other on channels known to both, or with the outside world, independently of the other.

Pi Calc: Basic Notation (4)

Process Grammar-Restriction

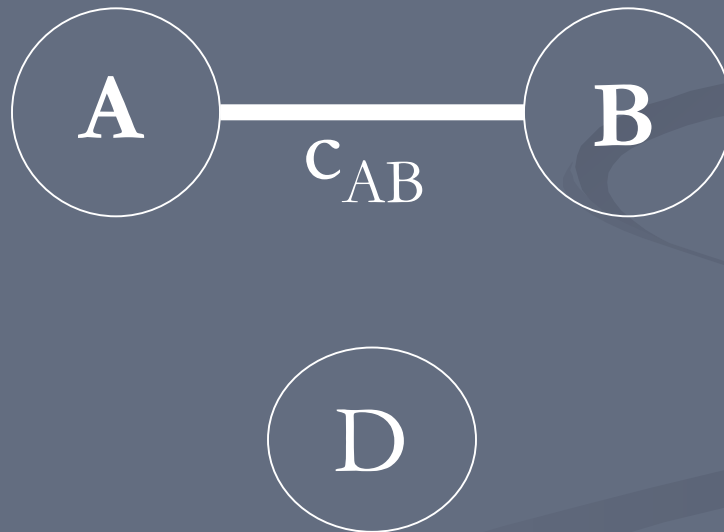
$$(\nu n)P$$

- A restriction $(\nu n)P$ is a process that makes a new, private name n , and then behaves as P . (Note that n is restricted to P)

Pi Calc: Restriction Example

c_{AB} is restricted to process A and B

$$(\nu c_{AB})(A \mid B)$$



Process D cannot use c_{AB}

Pi Calc: Basic Example

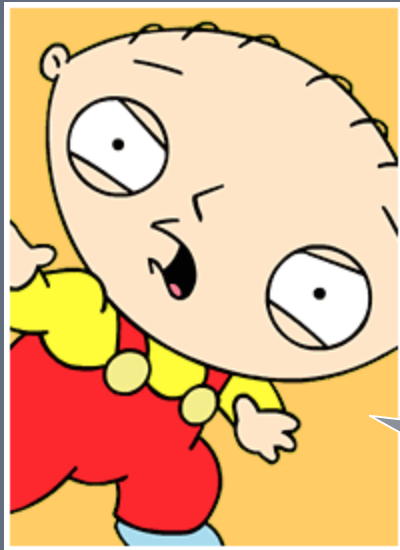
A basic example of a Protocol
using the notation we just learned

*

Pi Calc: Basic Example

Principal A uses the channel AB to send a single message M to Principal B

Principal A



Channel AB



Mother, I come bearing a gift. I'll give you a hint: it's in my diaper and it's not a toaster.

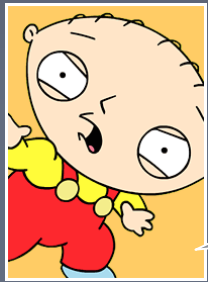
Principal B



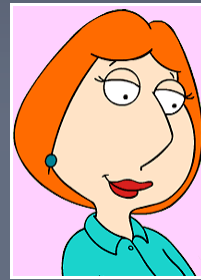
Pi Calc: Basic Example (2)

Principal A

Principal B



Channel AB

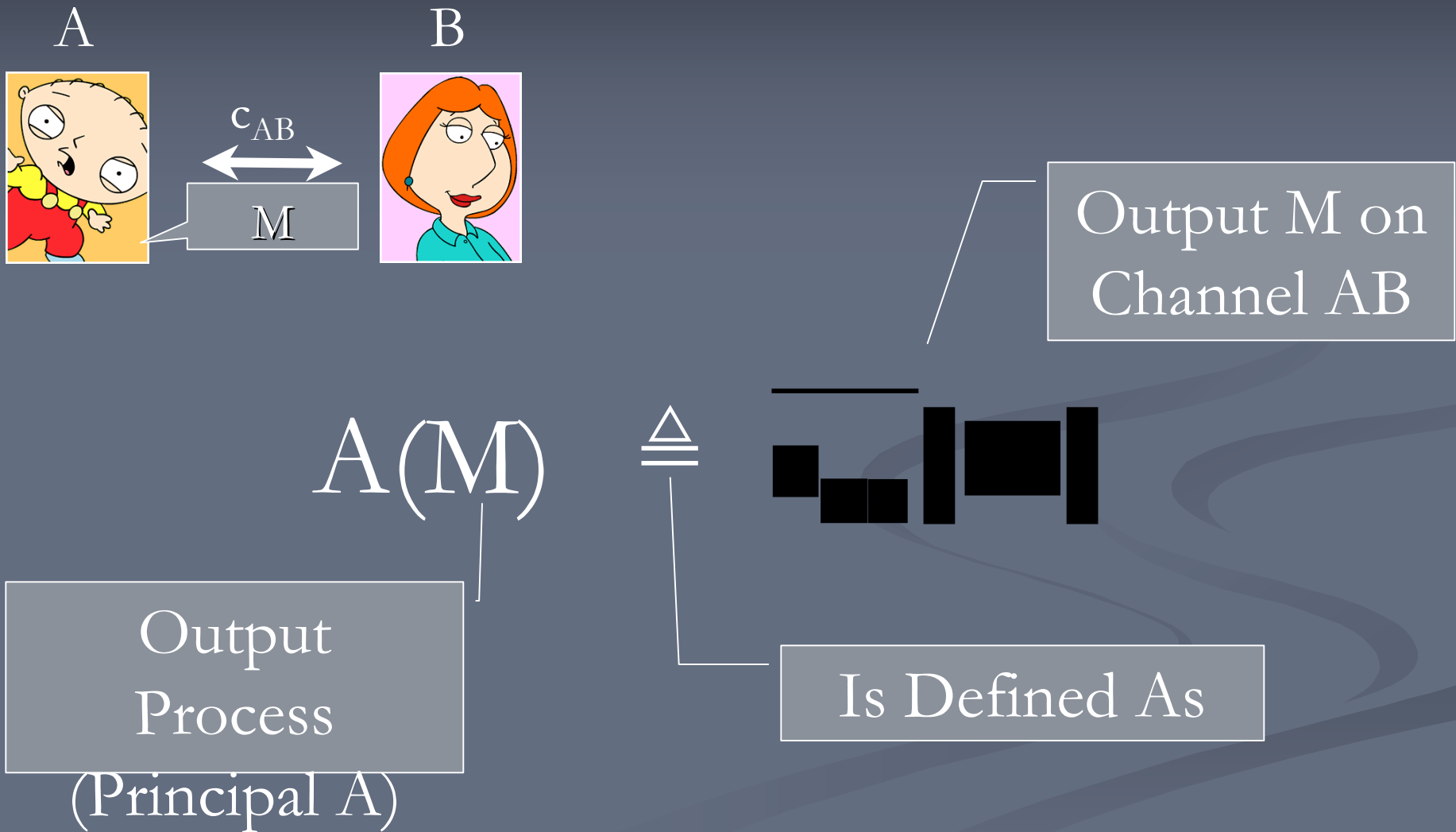


M

Message 1 $A \rightarrow B: M \text{ on } c_{AB}$

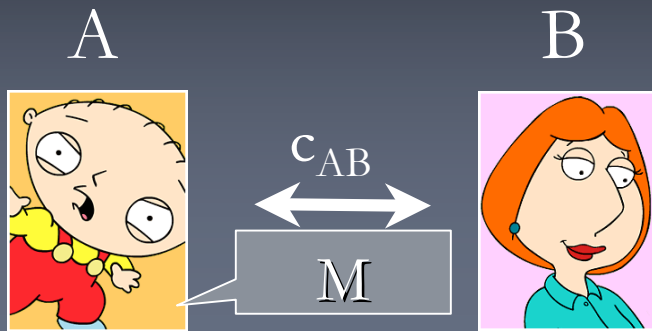
*

Pi Calc: Basic Example (3)



*

Pi Calc: Basic Example (4)



$$A(M) \triangleq \text{[Diagram of a process with a channel c_{AB} and a box M]}$$

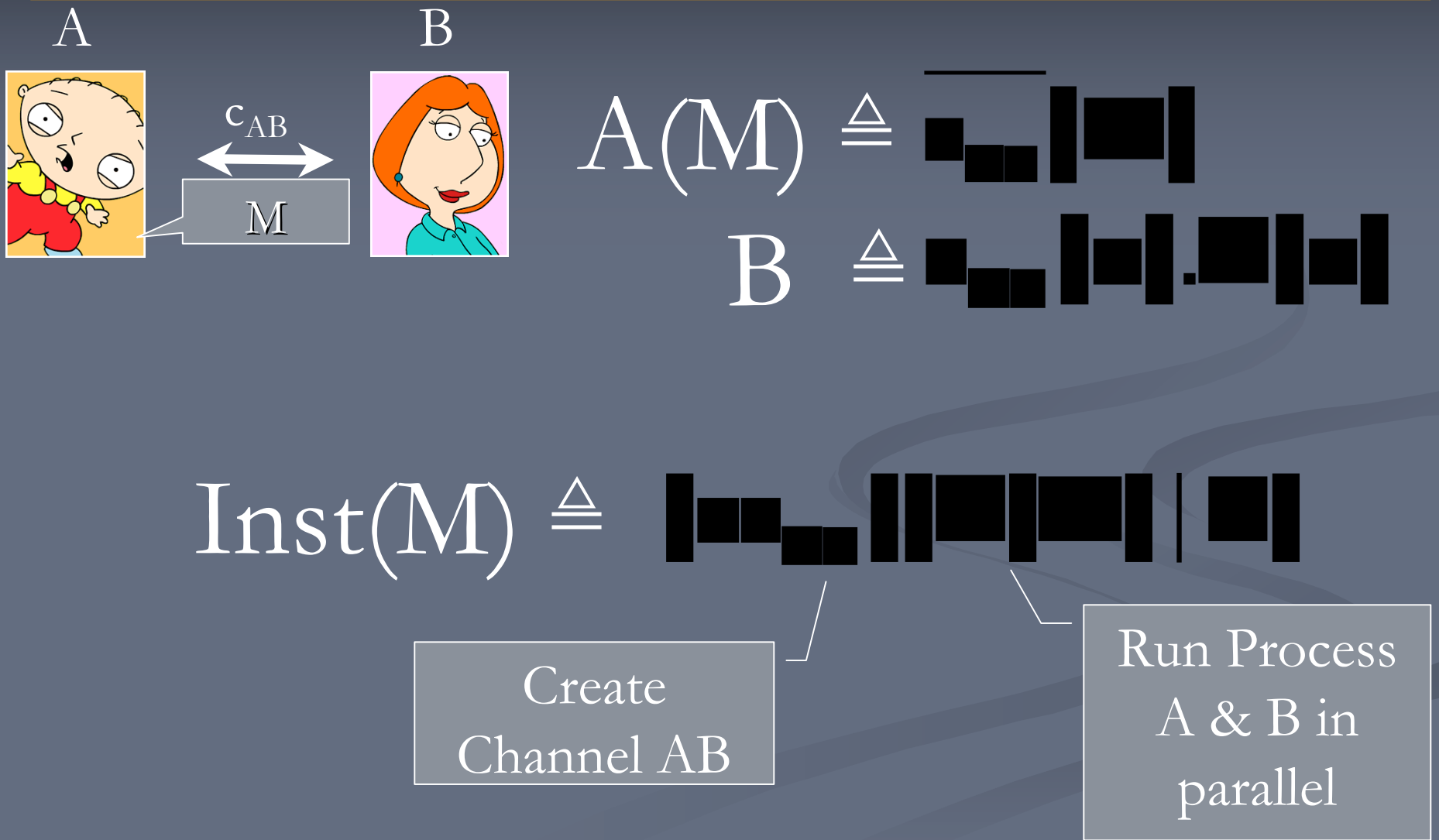
$$B \triangleq \text{[Diagram of a process with a channel c_{AB} and a box M]}$$

Input Process
(Principal B)

Input x from
Channel AB

Apply
F to x

Pi Calc: Basic Example (5)



Basic Example Protocol Final (6)

$$A(M) \triangleq \overline{a} \langle M \rangle$$

$$B \triangleq a(x). \overline{b} \langle x \rangle$$

$$\text{Inst}(M) \triangleq \overline{a} \langle M \rangle \mid a(x). \overline{b} \langle x \rangle$$

Pi Calc: Basic Example Properties

- 1) Authenticity (Integrity)
- 2) Secrecy

We will show why the basic protocol has these properties using informal and then formal syntax

*

Pi Calc: Authenticity (2)

$$A(M) = \overline{c} \langle M \rangle . A$$
$$B = c(x) . F(x) . B$$

Always
M

$$\text{Inst}(M) = \overline{c} \langle M \rangle . c(x) . F(x)$$

Process **B** always applies the function **F** to
The message **M**, that **A** sends.

Why is that?

*

Pi Calc: Authenticity (3)

The restriction operator restricts the channel AB to principal A(M) and B

The Channel AB is Secure

$\text{Inst}(M) \triangleq$ 

Restriction on Channel AB

Scope of the Restriction

Pi Calc: Authenticity (4)

$$A(M) = \overline{c} \langle \text{[blocks]} \rangle$$

$$B = c \langle \text{[blocks]} \rangle$$

Always
M

$$\text{Inst}(M) = \langle \text{[blocks]} \rangle$$

Since only process A and B communicate on c_{AB} , and the only thing being sent on that channel is M, $F(x)$ is really always $F(M)$

Pi Calc: Authenticity (5)

An attacker cannot cause B to apply F to some message other than M.

Pi Calc: Secrecy

$$A(M) = \overline{c} \langle M \rangle$$

$$B = c \langle \cdot \rangle$$

$$\text{Inst}(M) = \overline{c} \langle M \rangle . c \langle \cdot \rangle$$

The message M cannot be read in transit from
Principal A to Principal B
(Since c_{AB} is secure)

Pi Calc: Secrecy (2)

$$A(M) = \overline{\text{[redacted]}}$$

$$B = \text{[redacted]}$$

$$\text{Inst}(M) = \text{[redacted]}$$

If \mathbf{F} does not reveal \mathbf{M} , then the whole protocol does not reveal \mathbf{M}

Pi Calc: Indistinguishability

$$P \approx Q$$

The behaviors of process P and Q
are indistinguishable

Pi Calc: Indistinguishability (2)

$$P \approx Q$$

Internally P and Q might be different. However, a third process R cannot tell the difference between running P and running Q .

Pi Calc: Secrecy (Formally)

We can state the secrecy property using
The concept of indistinguishability.

*

Pi Calc: Secrecy (Formally) (2)

If F does not reveal M , then the whole protocol does not reveal M

If $F(M) \simeq F(M')$ for all M and M' , then
 $\text{Inst}(M) \simeq \text{Inst}(M')$

$F(M) \simeq F(M') = F$ does not reveal M

$\text{Inst}(M) \simeq \text{Inst}(M') = \text{Protocol}$ does not reveal M

Pi Calc: Authenticity (Formally)

To formally show authenticity for our basic protocol we are going to compare the basic protocol to a specification.

*

Pi Calc: Authenticity: Specification

$$A(M) = \overline{\text{send}}(\text{key}, \text{enc}(M))$$

$$B_{\text{spec}} = \text{recv}(\text{key}, \text{dec}(\text{enc}(M)))$$

$$\text{Inst}_{\text{spec}}(M) = \text{send}(\text{key}, \text{enc}(M)) \text{ . } \text{recv}(\text{key}, \text{dec}(\text{enc}(M)))$$

We need to show that our protocol behaves the same as the above specification.

$$\text{Inst}(M) \simeq \text{Inst}_{\text{spec}}(M)$$

Pi Calc: Authenticity Formally

(Remember Informally: An attacker cannot cause B to apply F to some other message.)



Are these two indistinguishable?

Yes, because x is always M since the c_{AB} is secure and M is the only thing sent on it.

Pi Calc: Properties

To sum up

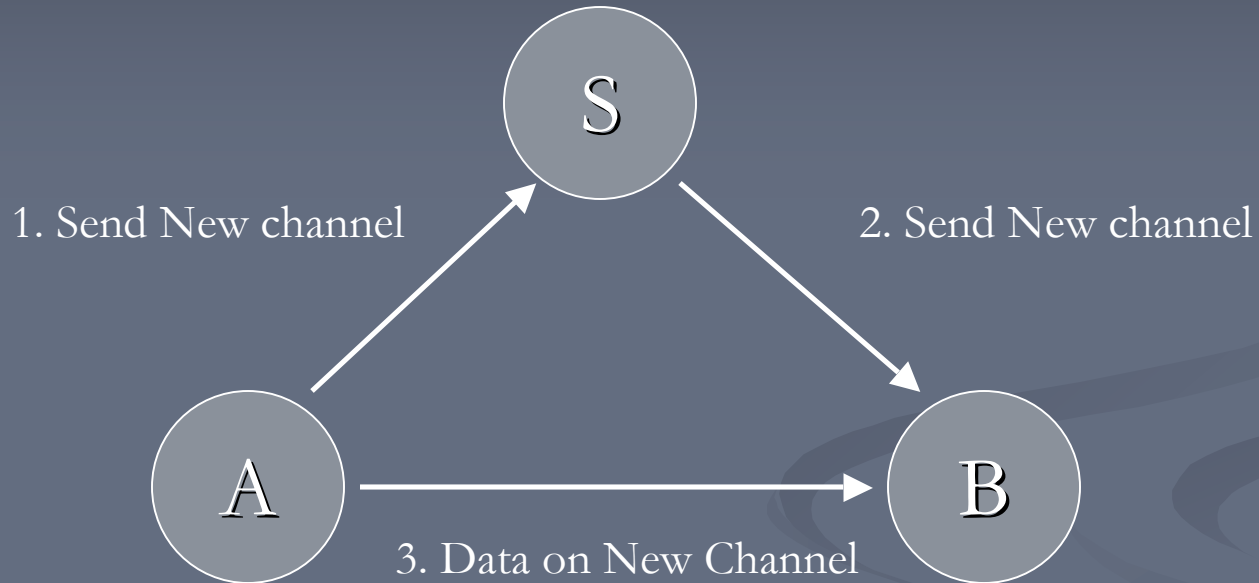
Authenticity: $\text{Inst}(M) \simeq \text{Inst}_{\text{spec}}(M)$, for all M

Secrecy: $\text{Inst}(M) \simeq \text{Inst}(M')$ if $F(M) \simeq F(M')$, for all M and M' .

*

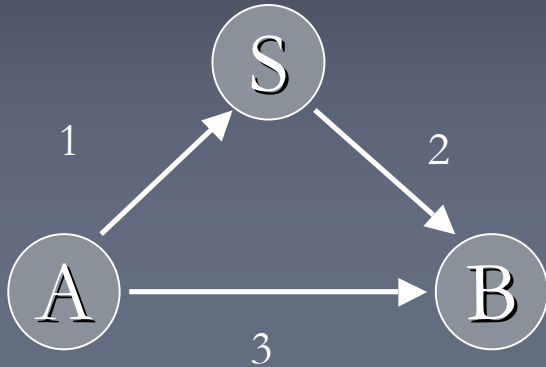
Pi Calc: Channel Establishment

C_{AS} and C_{SB} already exist



Wide Mouth Frog Protocol (Simplified)

Pi Calc: Channel Establishment (2)



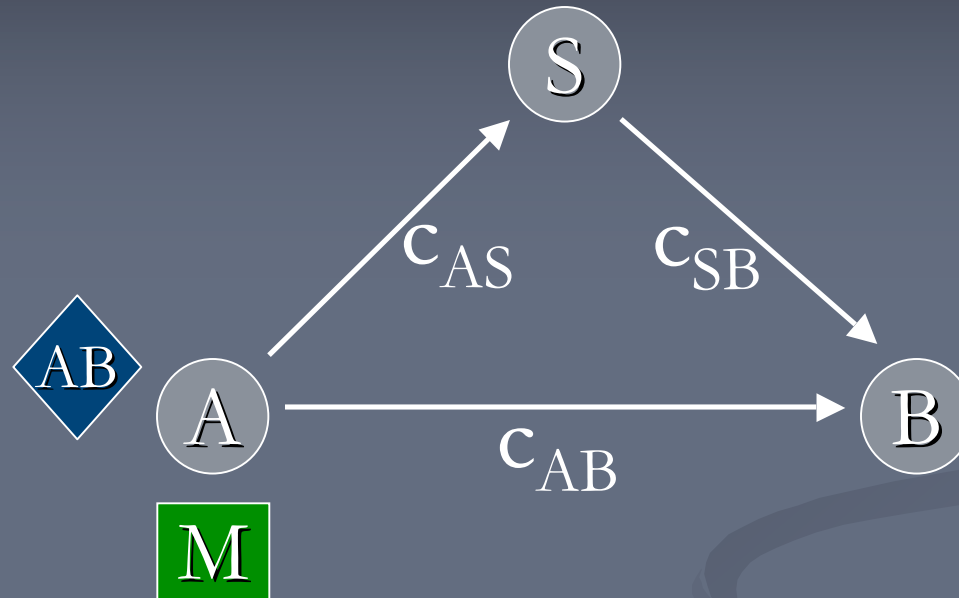
Note: The Message Can be a Channel

Message 1 $A \rightarrow S: c_{AB}$ on c_{AS}

Message 2 $S \rightarrow B: c_{AB}$ on c_{SB}

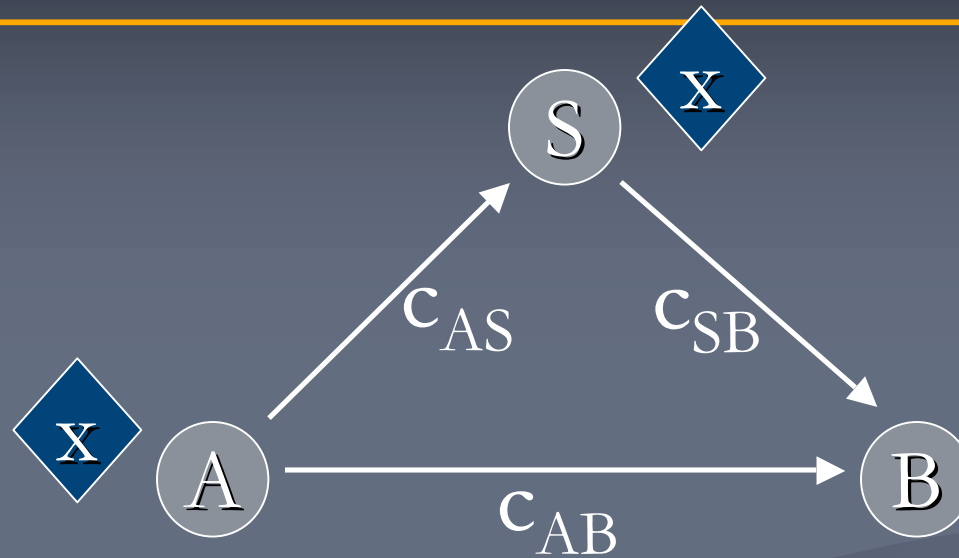
Message 3 $A \rightarrow B: M$ on c_{AB}

Pi Calc: Channel Establishment (3)



$$A(M) = \dots$$

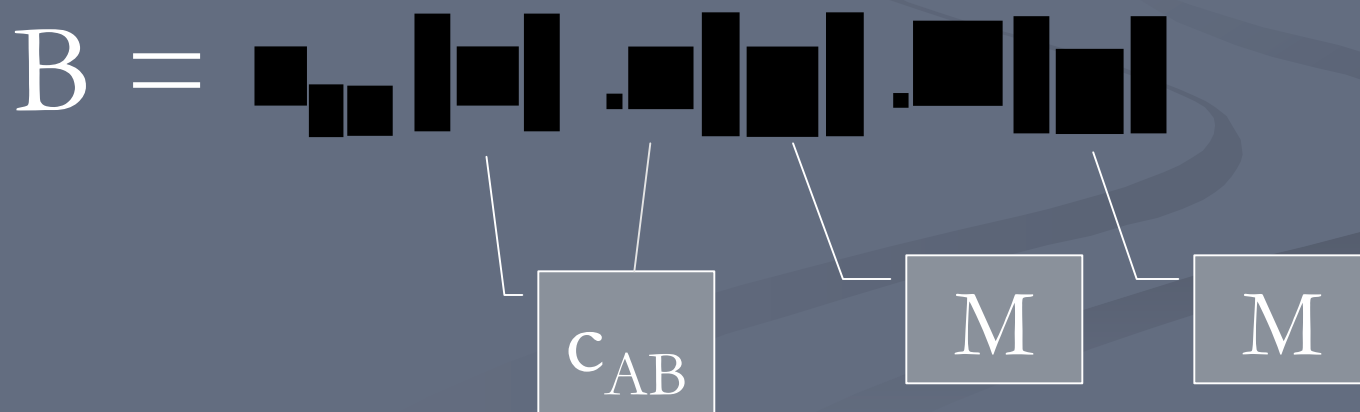
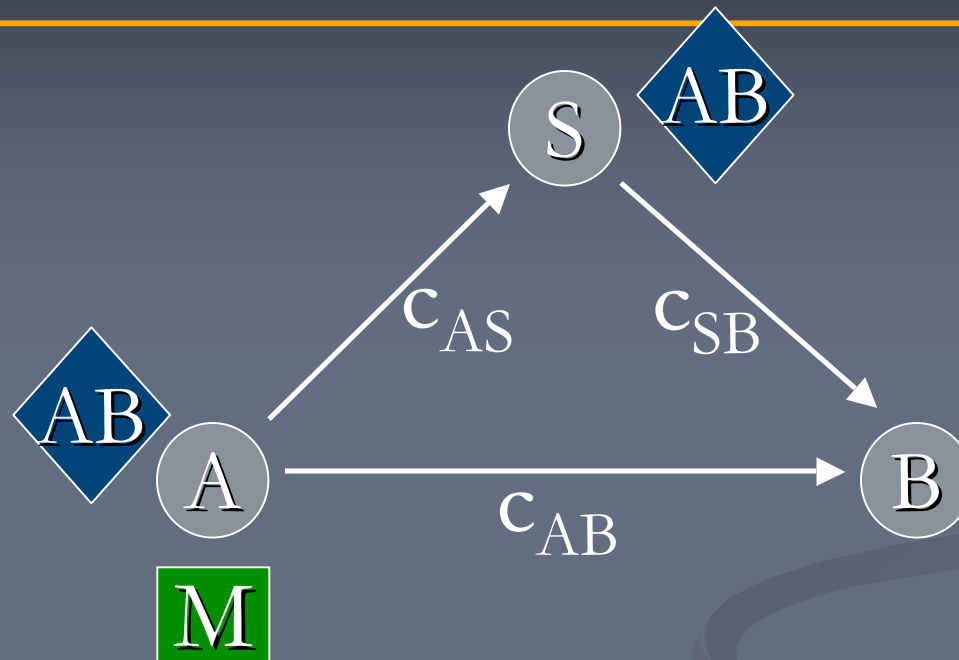
Pi Calc: Channel Establishment (4)



$$S = \overline{c_{AB}}(x) . x(y) . y(z) . \dots$$

The diagram shows a sequence of black bars representing a sequence of actions in a Pi Calculus process. A white box containing the label c_{AB} is connected by a line to the first bar in the sequence, indicating that the process starts with a channel creation action on c_{AB} .

Pi Calc: Channel Establishment (5)



Pi Calc: Channel Establishment Spec

$$A(M) = \text{[Illegible Pi Calculus Expression]}$$

$$S = \text{[Illegible Pi Calculus Expression]}$$

$$B_{\text{spec}} = \text{[Illegible Pi Calculus Expression]}$$

$$Inst_{\text{spec}}(M) = \text{[Illegible Pi Calculus Expression]}$$

Pi Calc: Channel Establishment Spec

In our channel establishment protocol
All three channels are secure.



Pi Calc: Authenticity and Secrecy

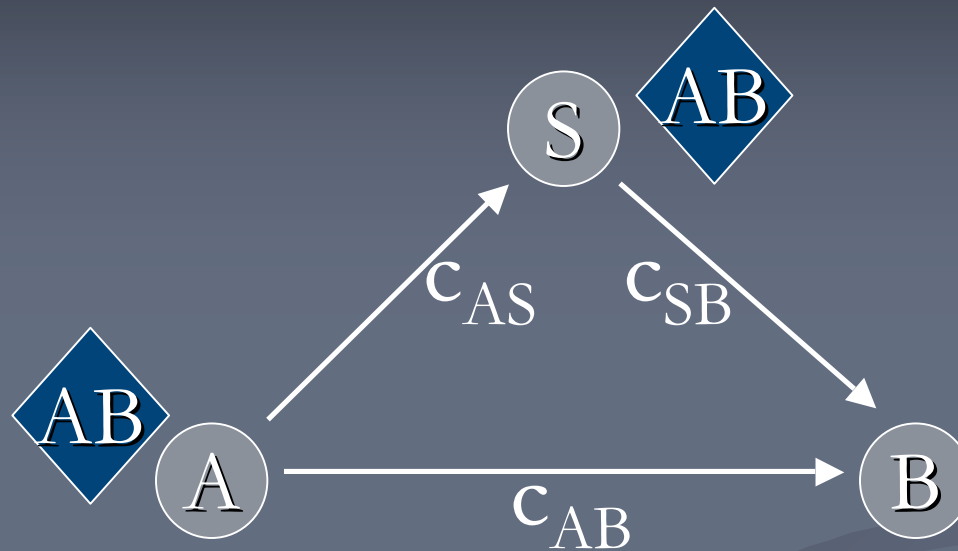
Channel Establishment Protocol

Authenticity: $\text{Inst}(M) \simeq \text{Inst}_{\text{spec}}(M)$, for all M

Secrecy: $\text{Inst}(M) \simeq \text{Inst}(M')$ if $F(M) \simeq F(M')$, for all M and M' .

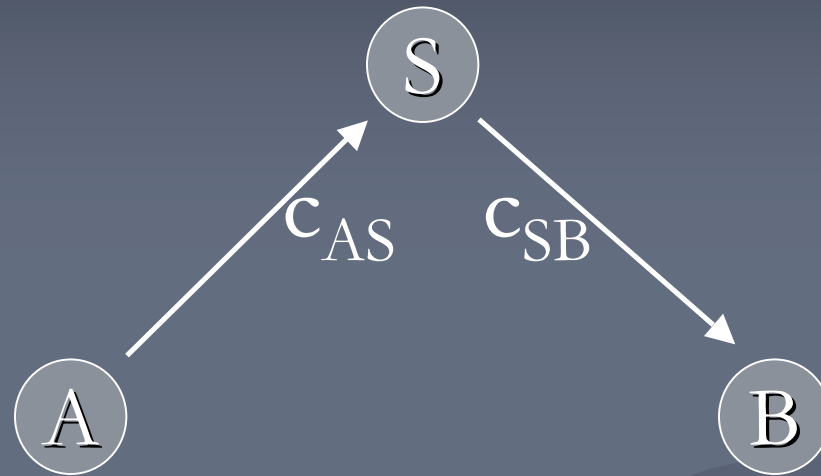
*

Pi Calc: Limitation 1



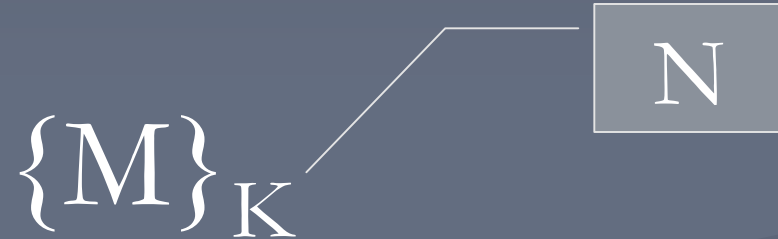
Sending Channels?

Pi Calc: Limitation 2



We require that we have secure channels already established which is almost never the case in the real world.

SPi Calc: Encryption



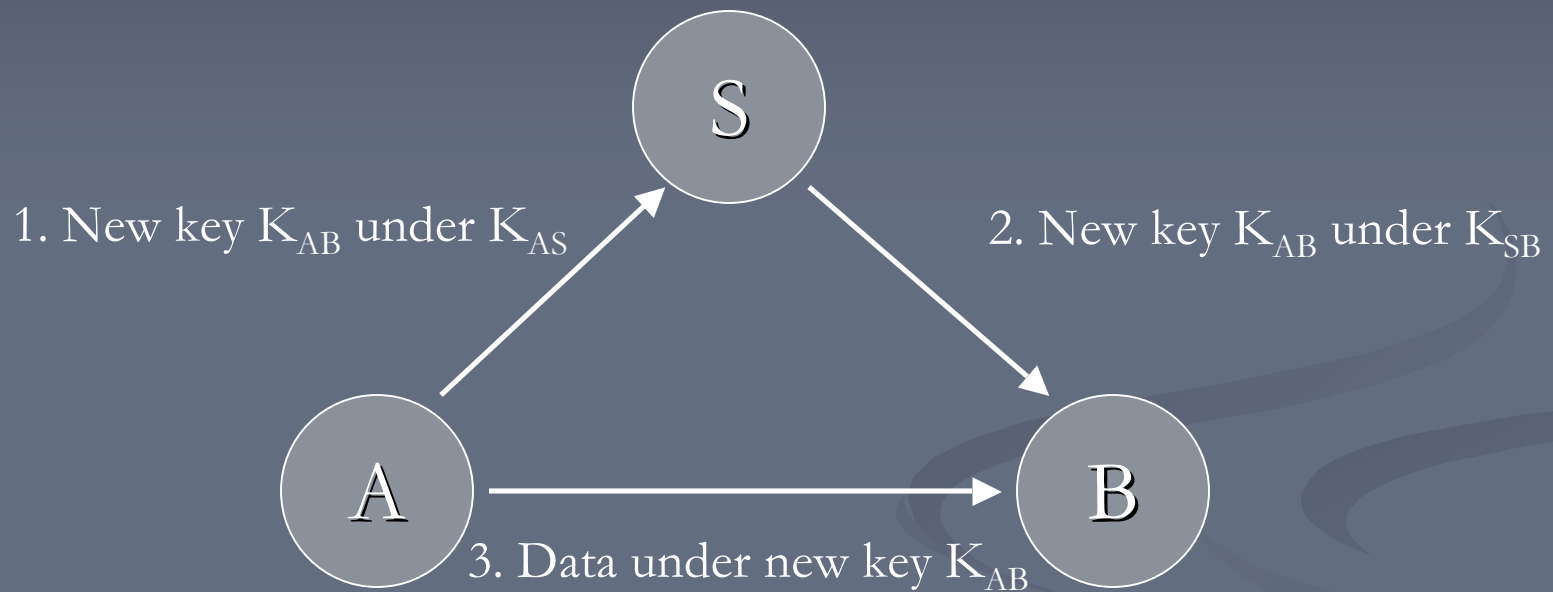
Message M encrypted under key K

SPi Calc: Decryption

case c of $\{m\}_k$ in P

Attempt to decrypt cipher text “c”
with key “k” resulting in plaintext
“m” used by process “P”

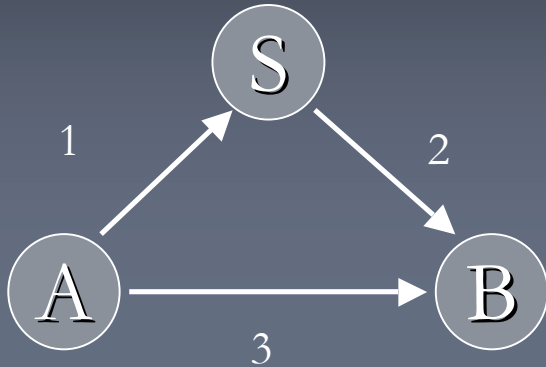
SPi Calc: Cryptographic Ex. (1)



Cryptographic Wide Mouth Frog Protocol (Simplified)

- 1) Uses Keys
- 2) Does not require secure channels

SPi Calc: Cryptographic Ex. (2)

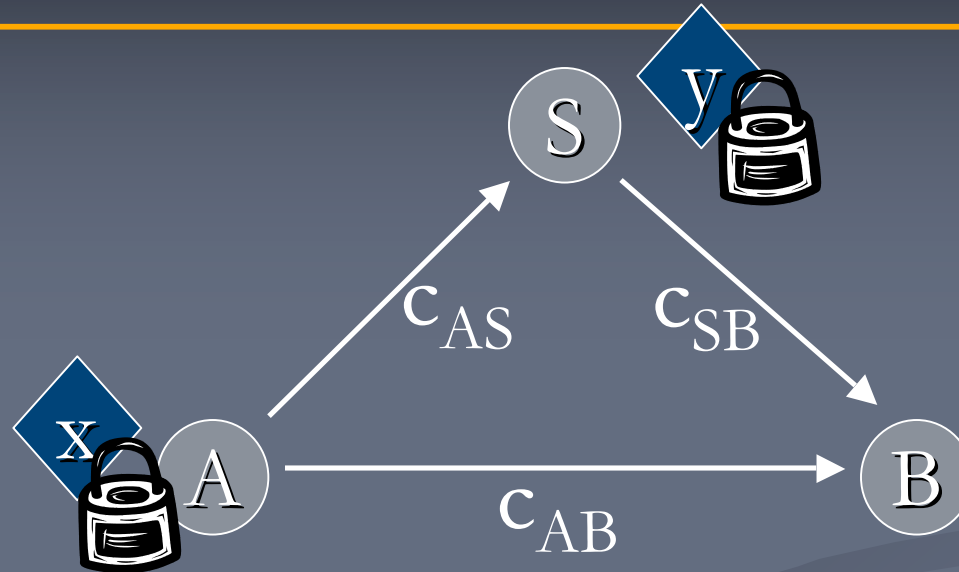


Message 1 $A \rightarrow S: \{K_{AB}\}_{K_{AS}}$ on c_{AS}

Message 2 $S \rightarrow B: \{K_{AB}\}_{K_{SB}}$ on c_{SB}

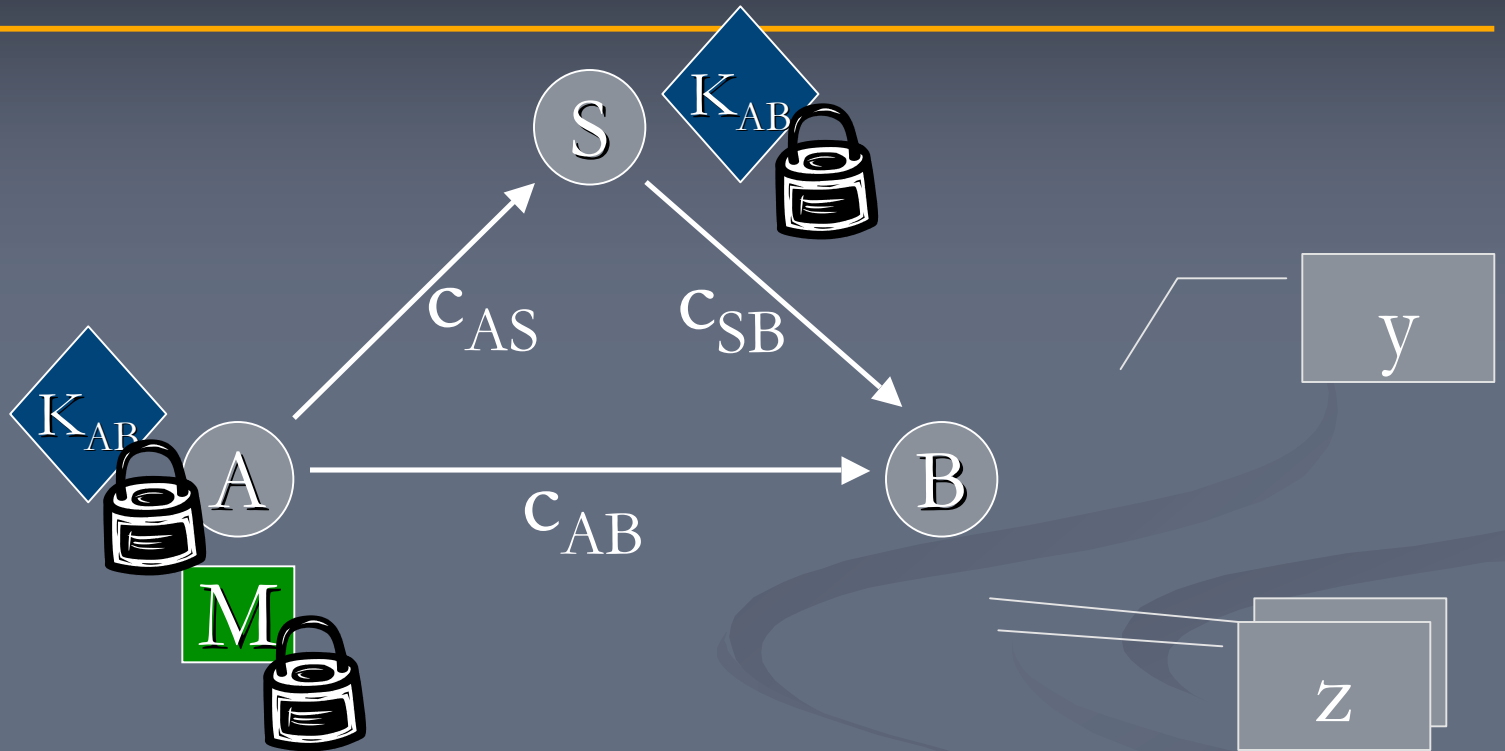
Message 3 $A \rightarrow B: \{M\}_{K_{AB}}$ on c_{AB}

SPi Calc: Cryptographic Ex. (4)



$$S = \text{[Bar chart representation of a signal or data sequence]}$$

SPi Calc: Cryptographic Ex. (5)



$B =$

SPi Calc: Cryptographic Ex. (6)

$$A(M) = \left(\overline{a_1} \cdot \overline{a_2} \cdot \overline{a_3} \cdot \overline{a_4} \cdot \overline{a_5} \cdot \overline{a_6} \cdot \overline{a_7} \cdot \overline{a_8} \cdot \overline{a_9} \cdot \overline{a_{10}} \right)$$

$$S = \overline{a_1} \cdot \overline{a_2} \cdot \overline{a_3} \cdot \overline{a_4} \cdot \overline{a_5} \cdot \overline{a_6} \cdot \overline{a_7} \cdot \overline{a_8} \cdot \overline{a_9} \cdot \overline{a_{10}}$$

$$B = \overline{a_1} \cdot \overline{a_2} \cdot \overline{a_3} \cdot \overline{a_4} \cdot \overline{a_5} \cdot \overline{a_6} \cdot \overline{a_7} \cdot \overline{a_8} \cdot \overline{a_9} \cdot \overline{a_{10}}$$

$$Inst(M) = \overline{a_1} \cdot \overline{a_2} \cdot \overline{a_3} \cdot \overline{a_4} \cdot \overline{a_5} \cdot \overline{a_6} \cdot \overline{a_7} \cdot \overline{a_8} \cdot \overline{a_9} \cdot \overline{a_{10}}$$

SPi Calc: Ideal Protocol

Ideal protocol once again has authenticity

(Remember Informally: An attacker cannot cause B to apply F to some message other than M.)

$$B_{\text{spec}} = \begin{array}{l} \text{[Diagram of a sequence of messages between A and B]} \\ \text{[Diagram of a sequence of messages between A and B]} \end{array}$$

SPi Calc: Authenticity

$$A(M) = \text{Enc}_{K_{AB}}(M)$$

Key AB is
restricted

Since K_{AB} is restricted only A,B and S know K_{AB} (A created it and sent it to B through S)

Remember: S is trusted so no problem there

SPi Calc: Authenticity

$$B = \text{[Diagrammatic representation of a block cipher operation]} \leftarrow K_{AB}$$

F is only called when the decryption works.
The decryption only works when “w” is encrypted with K_{AB} . Therefore, F is only called when “w” is encrypted with K_{AB} .

SPi Calc: Authenticity



F is only called when “w” is encrypted with K_{AB} . Since only A can send a message encrypted with K_{AB} the only time F gets called is when A sends B a message.

SPi Calc: Authenticity

So we can say an attacker cannot cause B to apply F to some message other than M

Authenticity: $\text{Inst}(M) \simeq \text{Inst}_{\text{spec}}(M)$, for all M

SPi Calc: Secrecy

Since the message is encrypted with the restricted K_{AB} we know that as long as F does not reveal M then the whole protocol does not reveal M .

Secrecy: $\text{Inst}(M) \simeq \text{Inst}(M')$ if $F(M) \simeq F(M')$, for all M and M' .

SPi Calc: Problem

In the previous cryptographic protocol we have a problem when the attacker is an active attacker.

Why is that?

SPi Calc: Protocol Limitation

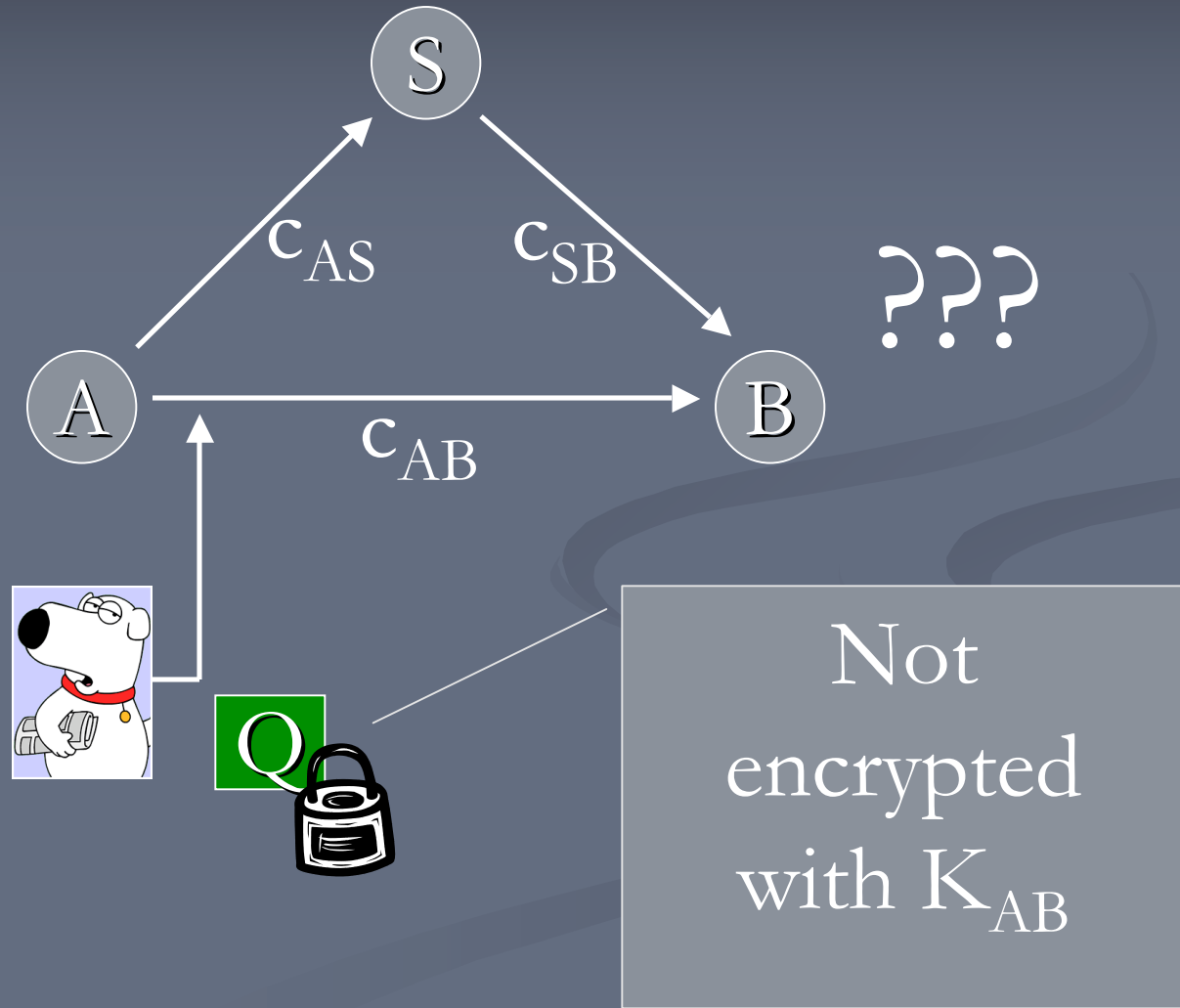
$$Inst(M) = \text{[Redacted]}$$

There is a problem here that has to do with the security of channel AB

There is no restriction on Channel AB



SPi Calc: Protocol Limitation



SPi Calc: Protocol Limitation

Next week we will present a better protocol written in Spi Calculus that can stand up to an active attacker.

SPi Calc: The End

