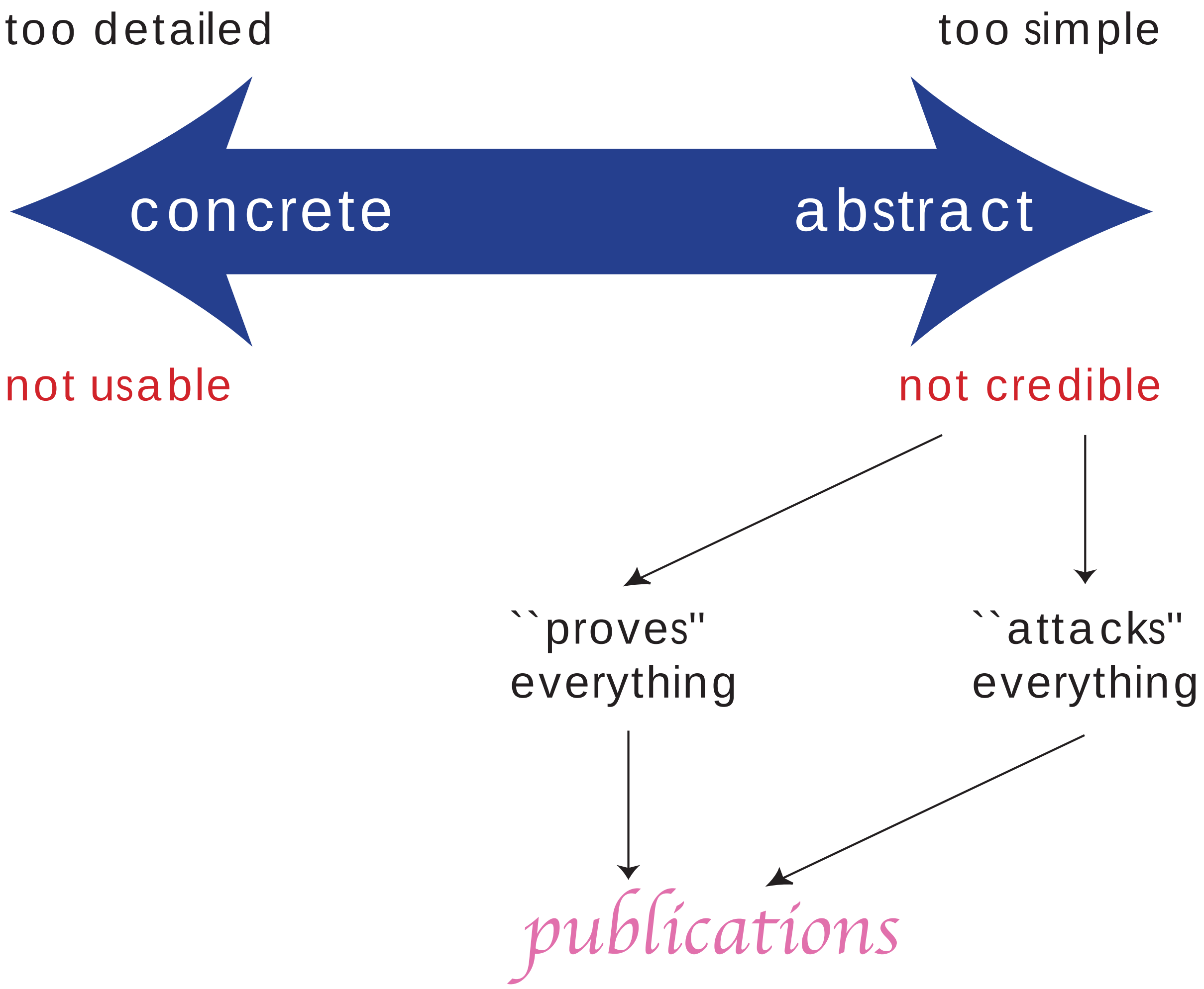
Proving Security Protocols Correct

Lawrence C. Paulson

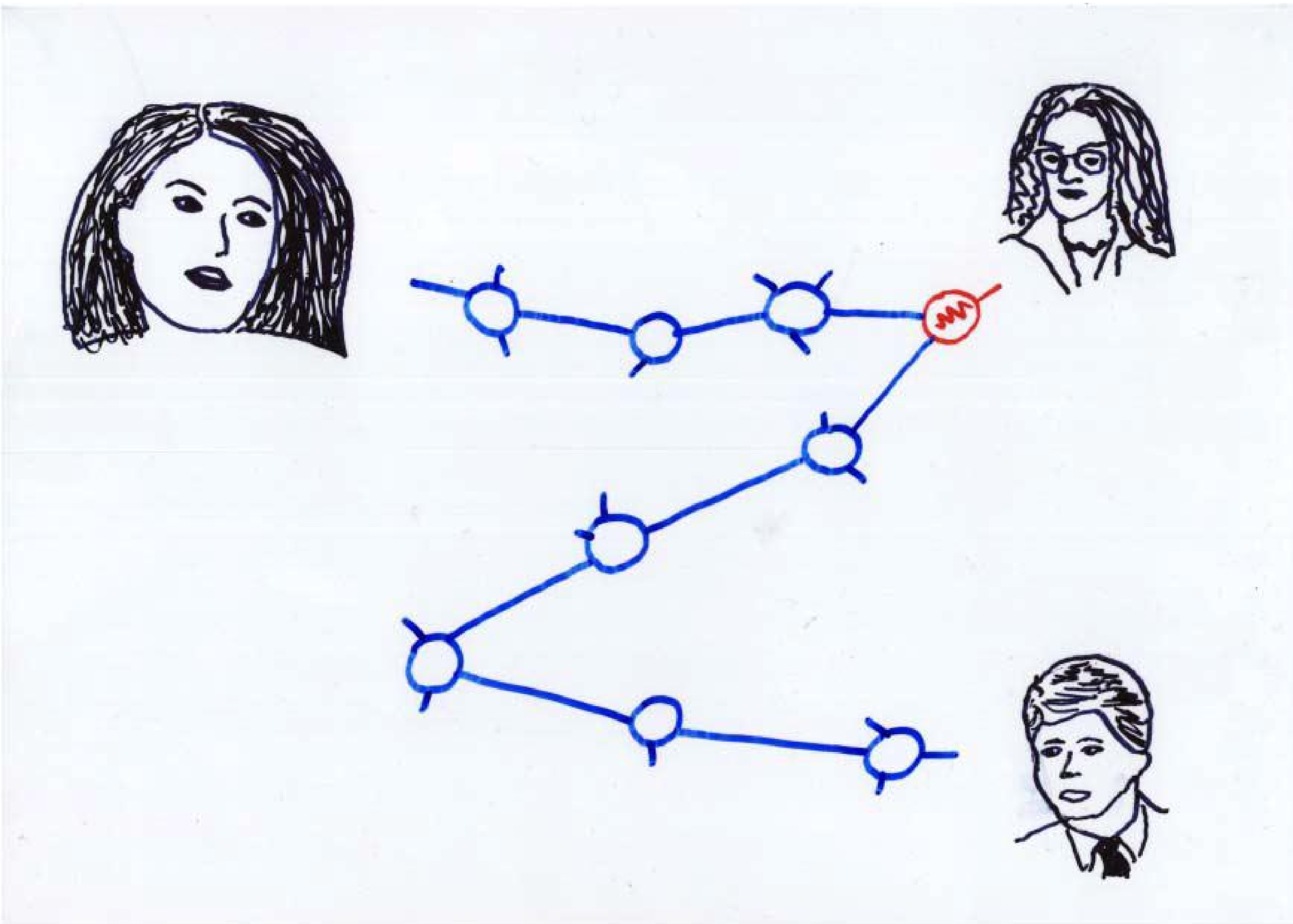
Computer Laboratory

UNIVERSITY OF CAMBRIDGE

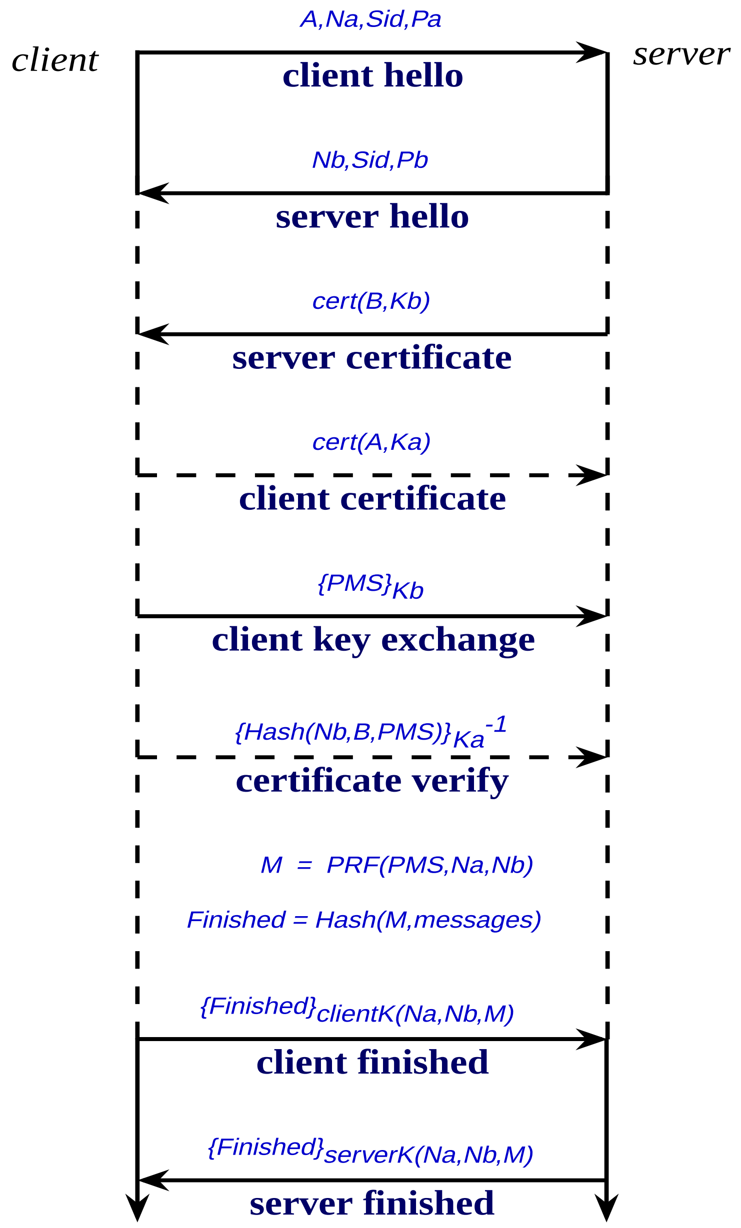
How Detailed Should a Model Be?



Case Study: the Plight of Monica and Bill



An Internet Security Protocol (TLS)



Why Are Security Protocols Often Wrong?

- they are TRIVIAL programs built from simple primitives, BUT they are complicated by
- concurrency
- a hostile environment
  – a bad user controls the network
- obscure concepts
- vague specifications
  – we have to guess what is wanted

Typical Protocol Goals

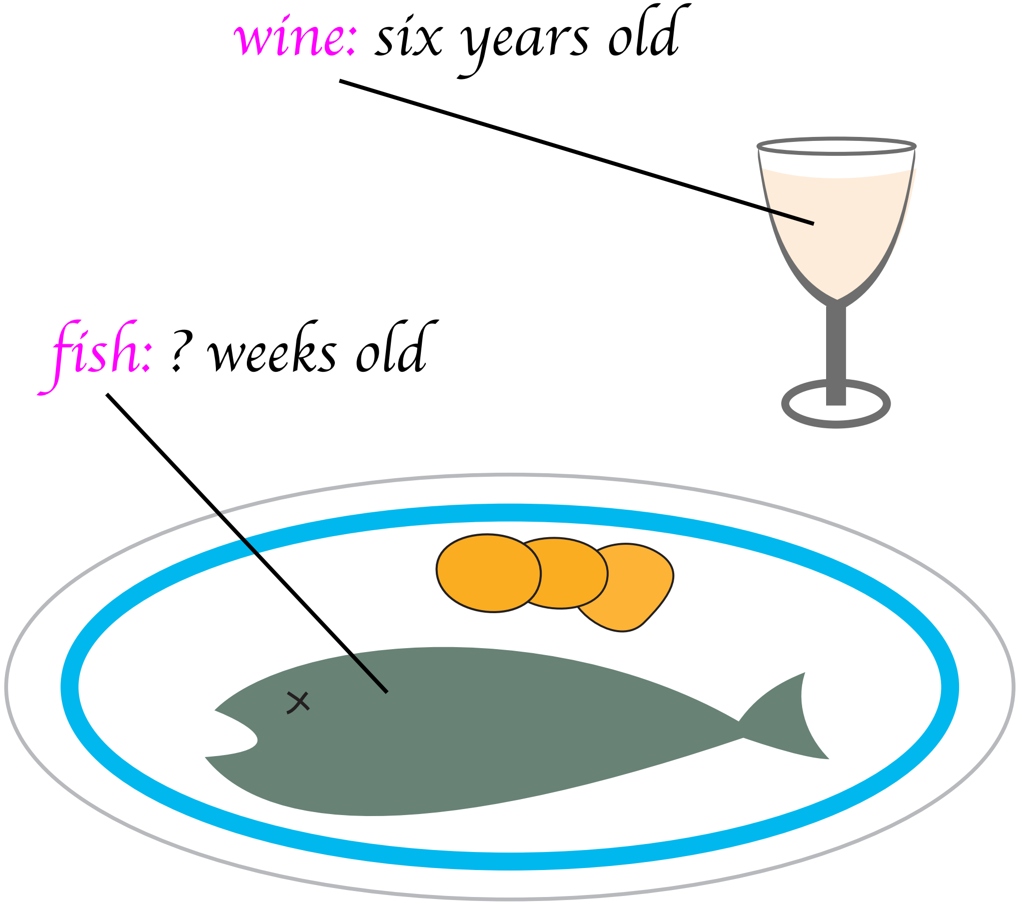
- Authenticity: who sent it?
- Integrity: has it been altered?
- Secrecy: who can receive it?
- Anonymity
- Non-repudiation . . .

all SAFETY *properties*

What Are Session Keys?

- used for a *single session*
- not safeguarded forever
- distributed using long-term keys
- could eventually become compromised
- can only be trusted if FRESH

Freshness, or Would You Eat This Fish?



Packaging a Session Key for Bill

session key

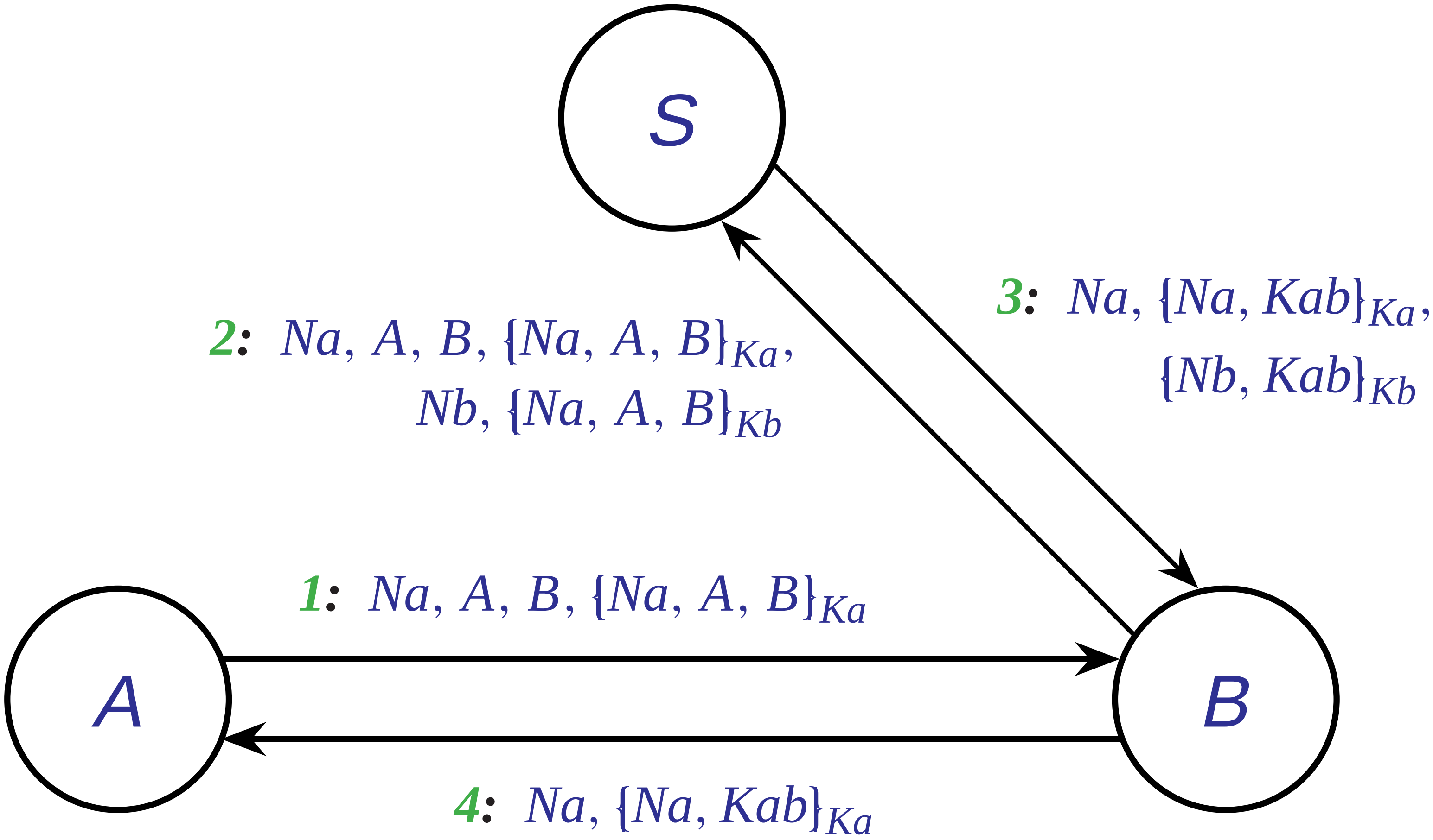
$$\{K, A, Nb\}_{Kb}$$

sealed using
Bill's key

person it's
shared with

nonce specified
by Bill:
proof of freshness

A Bad Variant of the Otway-Rees Protocol



A Splicing Attack with Interleaved Runs

1. $A \rightarrow C_B : Na, A, B, \{Na, A, B\}_{Ka}$

$1'$. $C \rightarrow A : Nc, C, A, \{Nc, C, A\}_{Kc}$

$2'$. $A \rightarrow C_S : Nc, C, A, \{Nc, C, A\}_{Kc}, Na', \{Nc, C, A\}_{Ka}$

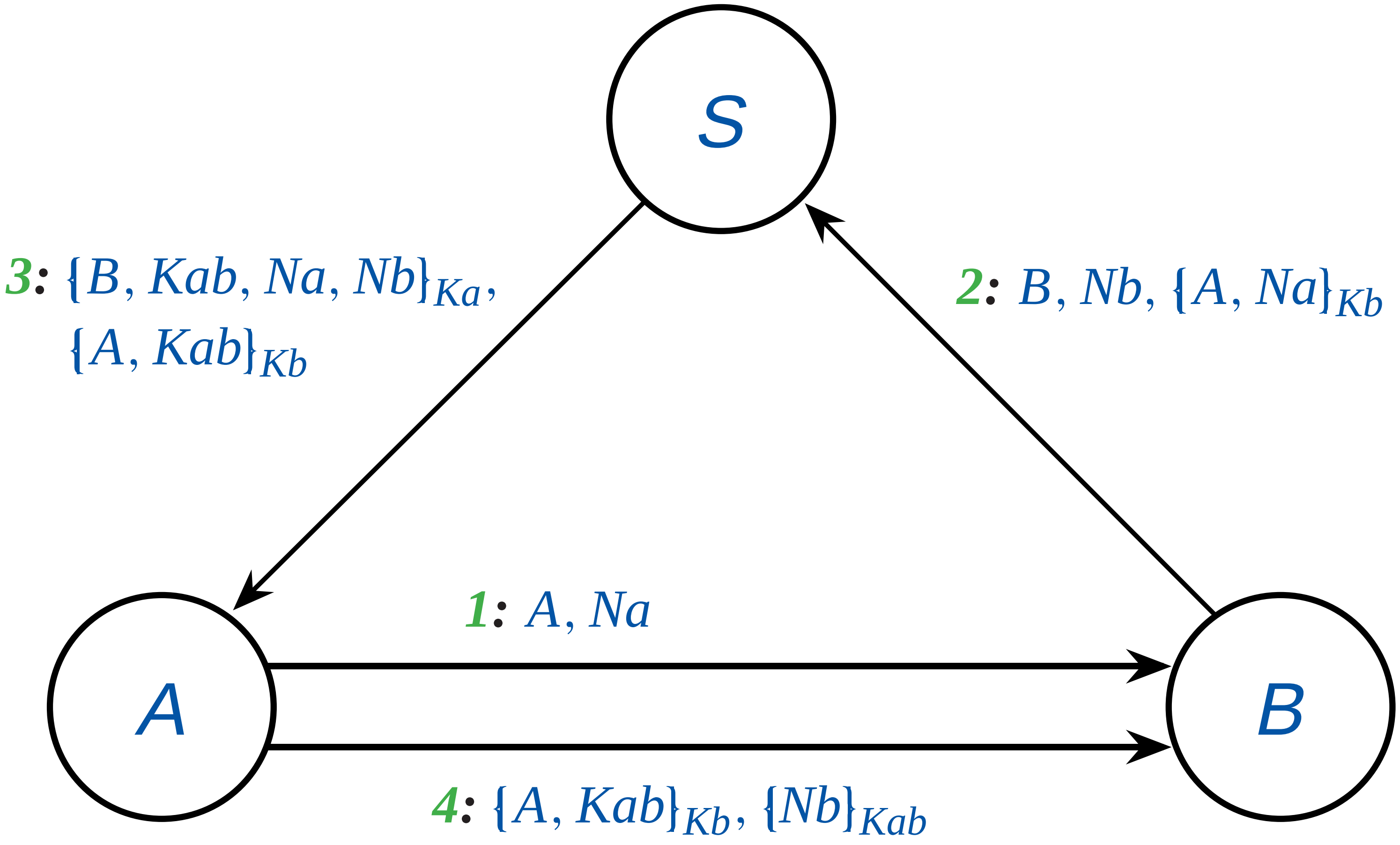
$2''$. $C_A \rightarrow S : Nc, C, A, \{Nc, C, A\}_{Kc}, Na, \{Nc, C, A\}_{Ka}$

$3'$. $S \rightarrow C_A : Nc, \{Nc, Kca\}_{Kc}, \{Na, Kca\}_{Ka}$

4. $C_B \rightarrow A : Na, \{Na, Kca\}_{Ka}$

Alice thinks the key Kca is shared with Bill, but it's shared with Carol!

A Bad Variant of the Yahalom Protocol



A Replay Attack

1. $C_A \rightarrow B : A, Nc$
2. $B \rightarrow C_S : B, Nb, \{A, Nc\}_{Kb}$
4. $C_A \rightarrow B : \{A, K\}_{Kb}, \{Nb\}_K$

Carol has broken the old key, K. She makes Bill think it is shared with Alice.

Verification Method I: Authentication Logics

BAN logic: Burrows, Abadi, Needham (1989)

Short proofs using high-level primitives:

Nonce N is fresh

Key Kab is good

Agent S can be trusted

- good for freshness
- not-so-good for secrecy or splicing attacks

Verification Method II: State Enumeration

Specialized tools	(Meadows)
General model-checkers	(Lowe)

Model protocol as a finite-state system

- automatically finds splicing attacks
- freshness is hard to model

Try using formal proof!

Why An Operational Model?

- good fit to informal protocol proofs: *inductive*
- simple foundations
- readable protocol specifications
- easily explained to security experts
- easily mechanized using *Isabelle*

An Overview of Isabelle

- uses higher-order logic as a logical framework
- generic treatment of inference rules
- logics supported include ZF set theory & HOL
- powerful simplifier & classical reasoner
- strong support for *inductive definitions*

Overview of the Model

- Traces of events
  - A sends B message X
  - A receives X
  - A stores X
- A powerful attacker
  - is an accepted user
  - attempts all possible splicing attacks
  - has the same specification in all protocols

Agents and Messages

$$agent \quad A, B, \ldots = \text{Server} \mid \text{Friend}\, i \mid \text{Spy}$$

$$\begin{aligned} message \; X, Y, \ldots = \; & \text{Agent}\, A \\ \mid \; & \text{Nonce}\, N \\ \mid \; & \text{Key}\, K \\ \mid \; & \{X, X'\} \qquad \text{compound message} \\ \mid \; & \text{Crypt}\, K\, X \end{aligned}$$

free algebras: we assume PERFECT ENCRYPTION

Functions over Sets of Messages

- $\mathsf{parts}\, H$: message components

  $$\mathsf{Crypt}\, K\, X \mapsto X$$

- $\mathsf{analz}\, H$: accessible components

  $$\mathsf{Crypt}\, K\, X,\ K^{-1} \mapsto X$$

- $\mathsf{synth}\, H$: expressible messages

  $$X,\ K \mapsto \mathsf{Crypt}\, K\, X$$

RELATIONS are traditional, but FUNCTIONS give us an equational theory

Operational Definition: $\mathsf{analz}\, H$

$$\frac{\mathsf{Crypt}\, K\, X \in \mathsf{analz}\, H \qquad K^{-1} \in \mathsf{analz}\, H}{X \in \mathsf{analz}\, H}$$

$$\frac{X \in H}{X \in \mathsf{analz}\, H} \qquad \frac{\{X, Y\} \in \mathsf{analz}\, H}{X \in \mathsf{analz}\, H} \qquad \frac{\{X, Y\} \in \mathsf{analz}\, H}{Y \in \mathsf{analz}\, H}$$

Typical derived law:

$$\mathsf{analz}\, G \cup \mathsf{analz}\, H \subseteq \mathsf{analz}(G \cup H)$$

Operational Definition: synth H

$$\frac{X \in H}{X \in \text{synth}\, H} \qquad \text{Agent}\, A \in \text{synth}\, H$$

$$\frac{X \in \text{synth}\, H \qquad Y \in \text{synth}\, H}{\{X, Y\} \in \text{synth}\, H} \qquad \frac{X \in \text{synth}\, H \qquad K \in H}{\text{Crypt}\, K\, X \in \text{synth}\, H}$$

- agent names can be guessed
- nonces & keys cannot be!

A Few Equations

$$\mathsf{parts}(\mathsf{parts}\, H) = \mathsf{parts}\, H \qquad \text{transitivity}$$

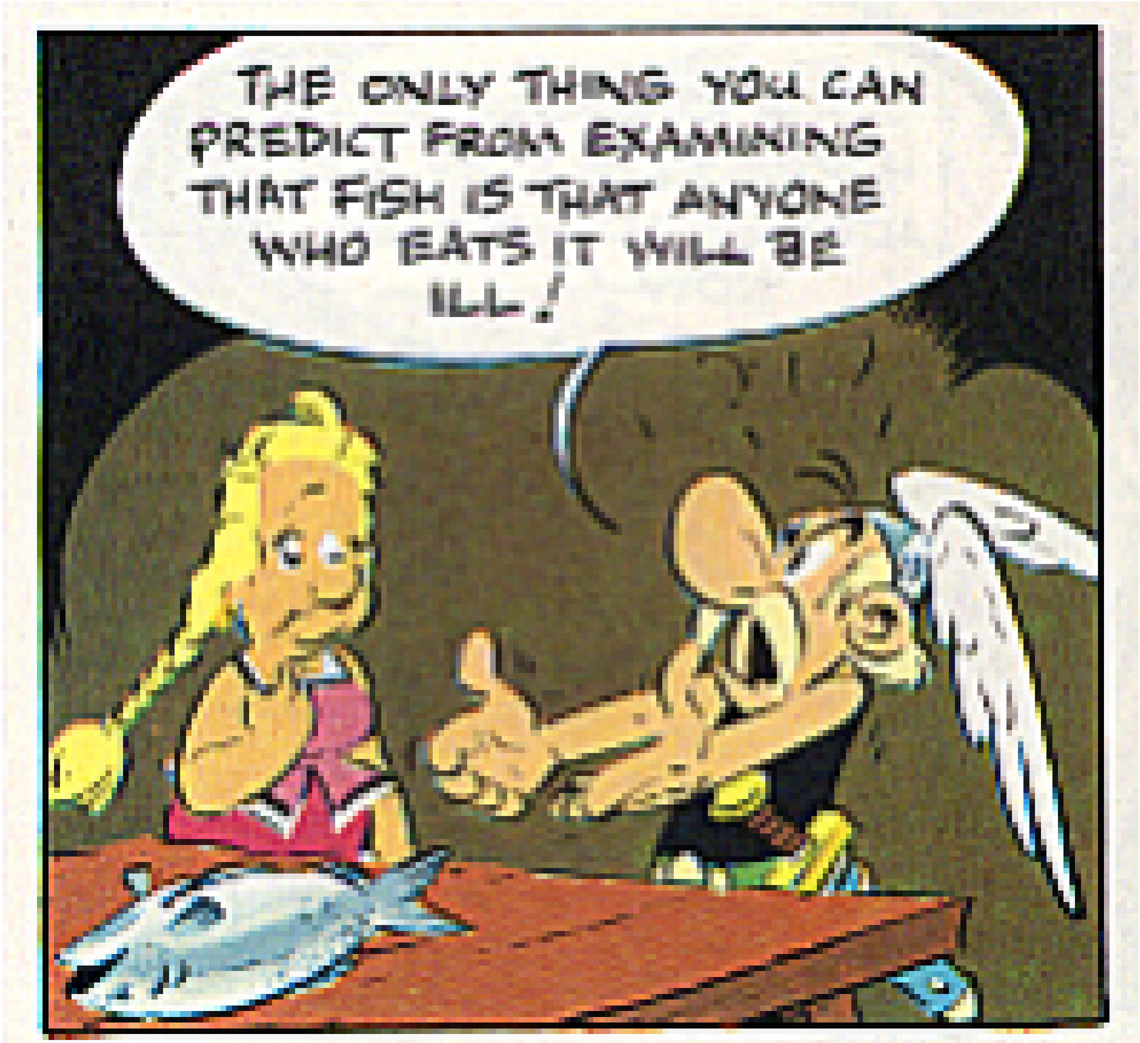
$$\mathsf{analz}(\mathsf{synth}\, H) = \mathsf{analz}\, H \cup \mathsf{synth}\, H \qquad \text{“cut elimination”}$$

Symbolic Evaluation:

$$\mathsf{analz}(\{\mathsf{Crypt}\, K\, X\} \cup H) =$$

$$\begin{cases} \{\mathsf{Crypt}\, K\, X\} \cup \mathsf{analz}(\{X\} \cup H) & \text{if } K^{-1} \in \mathsf{analz}\, H \\ \{\mathsf{Crypt}\, K\, X\} \cup \mathsf{analz}\, H & \text{otherwise} \end{cases}$$

What About Freshness?



Modelling Attacks and Key Losses

If $X \in \text{synth}(\text{analz}(\text{spies}\, evs))$

may add $\text{Says}\,\text{Spy}\; B\; X$ (Fake rule)

If the server distributes session key K

may add $\text{Notes}\,\text{Spy}\, \{Na, Nb, K\}$ (Oops rule)

Nonces show the TIME of the loss

Overview of Results

- facts proved by induction & classical reasoning
- simplifying analz H: case analysis, big formulas
- handles REAL protocols: TLS, Kerberos, . . .
- lemmas reveal surprising protocol features
- failed proofs can suggest attacks

Proofs require days or weeks of effort

Generalizing induction formulas is hard!

The Recursive Authentication Protocol

- designed in industry (APM Ltd)
- novel recursive structure: variable length
- VERIFIED by Paulson
  – assuming perfect encryption
- ATTACKED by Ryan and Schneider
  – using the specified encryption (XOR)

Doesn't proof give certainty? Not in the real world!

So Then, How Detailed Should a Model Be?

- detailed enough to answer the relevant questions
- abstract enough to fit our budget
- model-checking is almost free
  (thanks to Lowe, Roscoe, Schneider)
- formal proofs give more, but cost more

Don't let *theory* displace **reality**