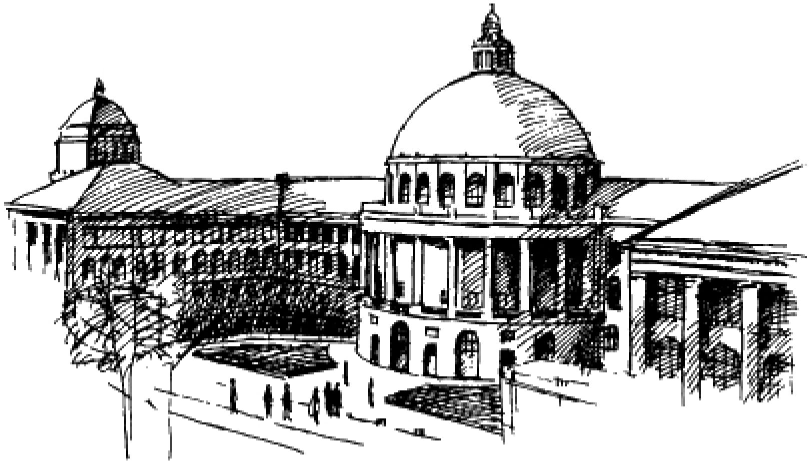
Information Security
From an Art to a Science

David Basin
ETH Zürich



An Increasingly Common Incident

Slammer Worm Crashes Ohio Nuke Network

The Slammer worm penetrated a private computer network at Ohio's Davis-Besse nuclear power plant in January and disabled a safety monitoring system for nearly five hours, despite a belief by plant personnel that the network was protected by a firewall.

Users noticed slow performance on Davis-Besse's business network at 9:00 a.m. By 4:00 p.m., power plant workers noticed a slowdown on the plant network. At 4:50 p.m., the congestion created by the worm's scanning crashed the plant's computerized display panel, called the Safety Parameter Display System. This system monitors the most crucial safety indicators at a plant, like coolant systems, core temperature sensors, and external radiation sensors.

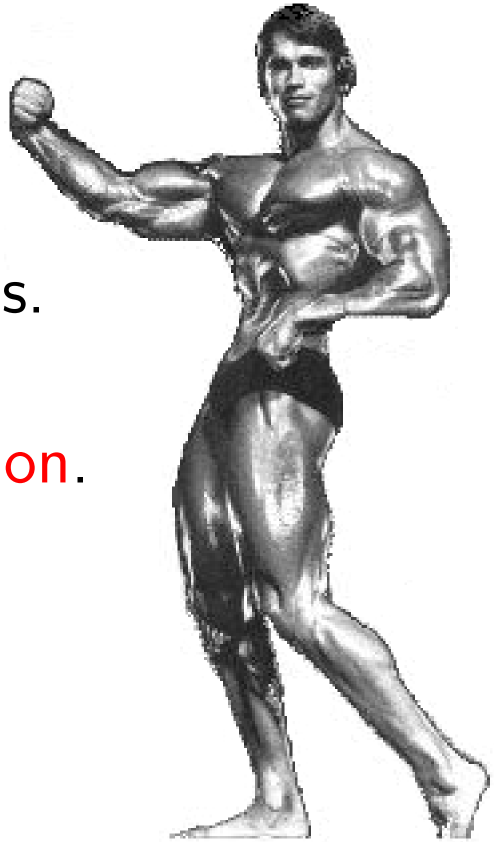
— *Kevin Poulsen, SecurityFocus Aug 19 2003*

What is being done about this?

What can be done about this?

Information Security as an Art

- Analyze threats to information assets and their risk



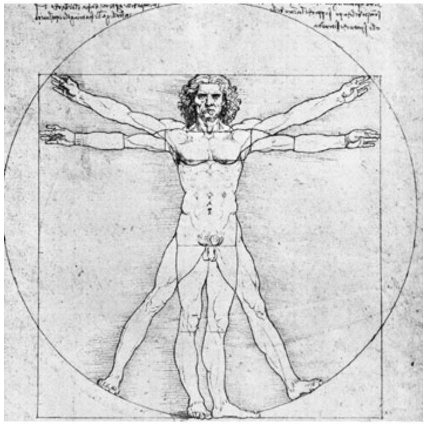
- Employ countermeasures to reduce risks, e.g.,
  - ▶ Harden your OS, shutting down unneeded services.
  - ▶ Strengthen your network perimeter with firewalls.
  - ▶ Require strong passwords and strong authentication.

- Such practices are good and useful.

  But after your system is strong and hard ...

  what can you actually say about its security?

Information Security as a Science



- Science: the discovery and knowing of something which can be demonstrated and verified within a community.

- Example: modern cryptography. Precise formalism for

  1. defining what security is and
  2. making verifiable statements that (possible under well-defined assumptions) algorithms are secure.

  Cryptography is very rarely the weak link in application security.

- What are analogs for other subareas of Information Security?

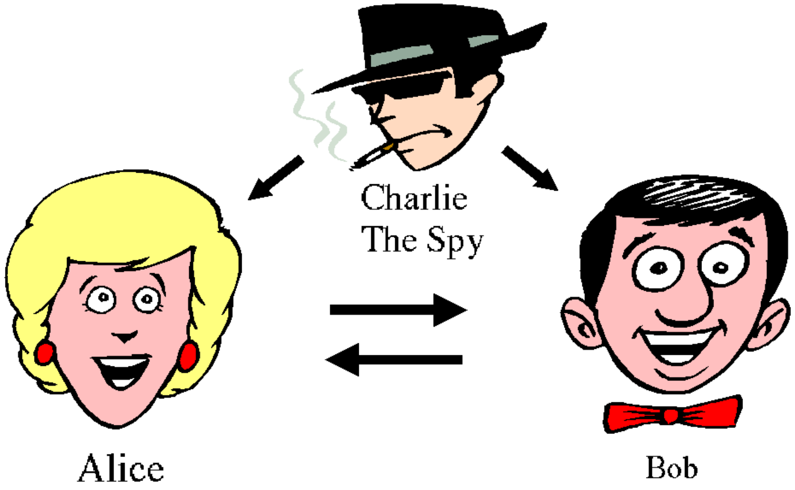
  How do we scale analysis from building blocks to larger systems?

An Example: Security Protocols

- Play a central role in securing networked information systems.

  E-commerce, wireless communication, ubiquitous computing, . . .

- An example



$A \rightarrow B$: "Send $10.000 to account XYZ."

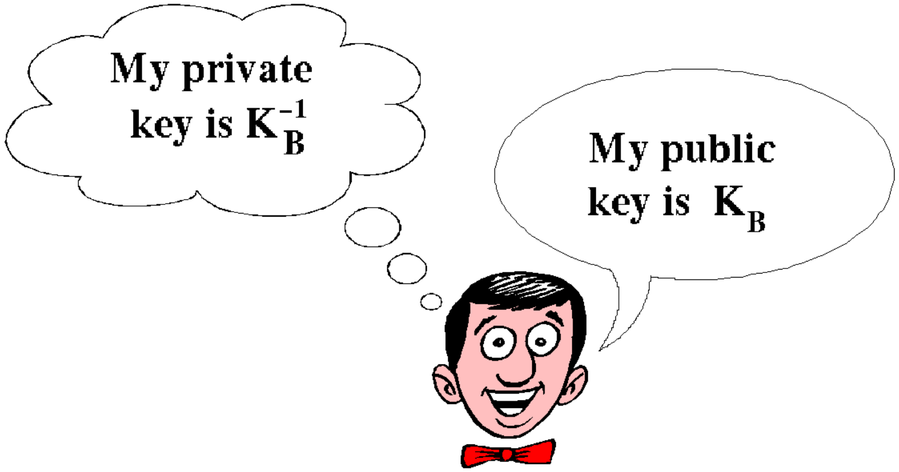
$B \rightarrow A$: "I'll transfer it now."

Authentication: How does B know he is really speaking with A?

- Related problems: confidentiality, integrity, accountability, etc.

Building Blocks for Security Protocols

Cryptographic Procedures: encryption of messages



$$\{\{M\}_{K_B}\}_{K_B^{-1}} = M$$

(Pseudo-)Random Number Generators: to generate “Nonces”, e.g. for “Challenge-Response”

Protocols: recipe for exchanging messages

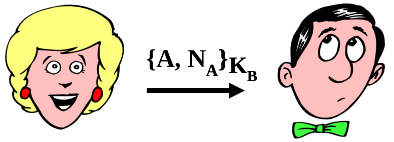
Steps like: *A sends B his name together with the message M. The pair $\{A, M\}$ is encrypted with B's key.*

$$A \rightarrow B : \{A, M\}_{K_B}$$

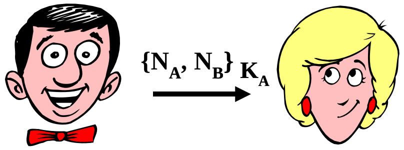
An Authentication Protocol (Needham-Schroeder)

$$\begin{array}{lll} 1. & A \rightarrow B: & \{A, N_A\}_{K_B} \\ 2. & B \rightarrow A: & \{N_A, N_B\}_{K_A} \\ 3. & A \rightarrow B: & \{N_B\}_{K_B} \end{array}$$

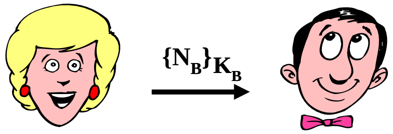
Translation:



“I am Alice and here is my Nonce (as challenge) N_A.”



“Here is your Nonce N_A and I also have one for you.”



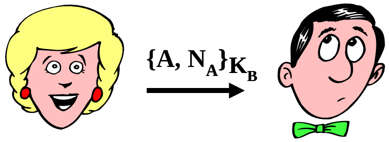
“I got it! It is N_B.”

Protocols are typically small and convincing

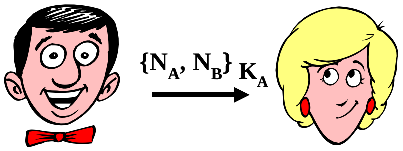
An Authentication Protocol (Needham-Schroeder)

$$\begin{array}{lll} 1. & A \rightarrow B: & \{A, N_A\}_{K_B} \\ 2. & B \rightarrow A: & \{N_A, N_B\}_{K_A} \\ 3. & A \rightarrow B: & \{N_B\}_{K_B} \end{array}$$

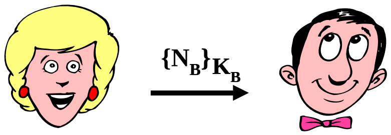
Translation:



"I am Alice and here is my Nonce (as challenge) N_A."



"Here is your Nonce N_A and I also have one for you."

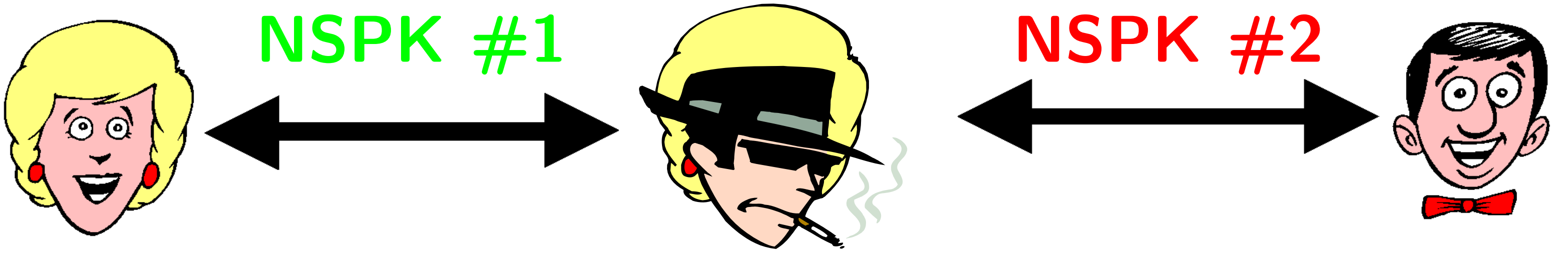


"I got it! It is N_B."

Protocols are typically small and convincing and (very) often wrong!

Man-in-the-Middle Attack

$$\begin{aligned} A \rightarrow B &: \{A, N_A\}_{K_B} \\ B \rightarrow A &: \{N_A, N_B\}_{K_A} \\ A \rightarrow B &: \{N_B\}_{K_B} \end{aligned}$$

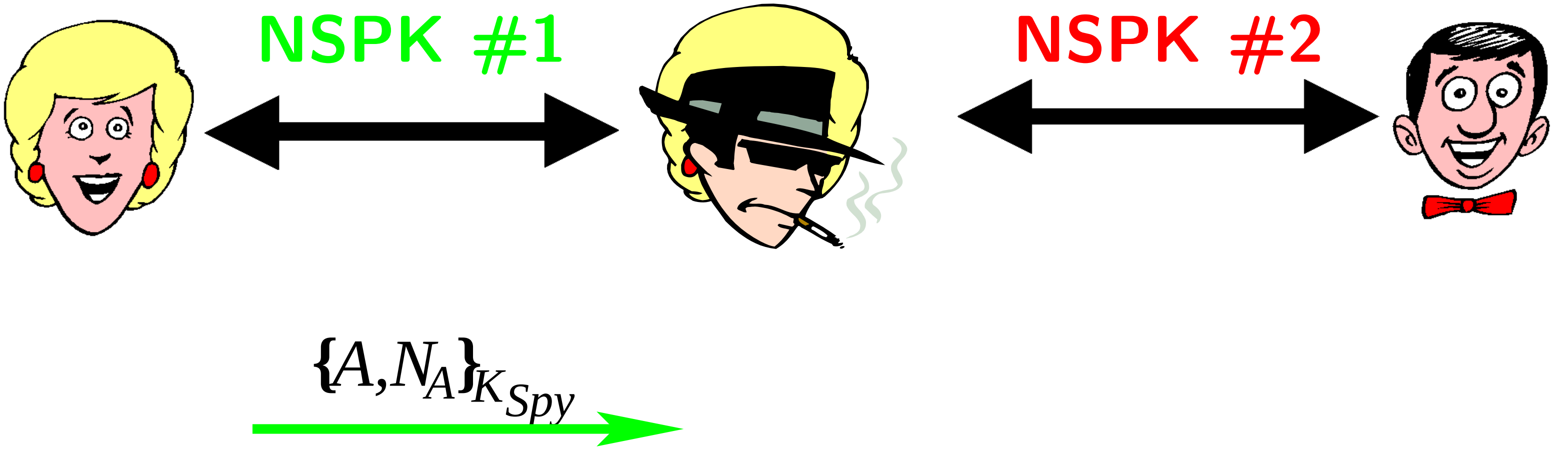


Man-in-the-Middle Attack

$A \rightarrow B : \{A, N_A\}_{K_B}$

$B \rightarrow A : \{N_A, N_B\}_{K_A}$

$A \rightarrow B : \{N_B\}_{K_B}$



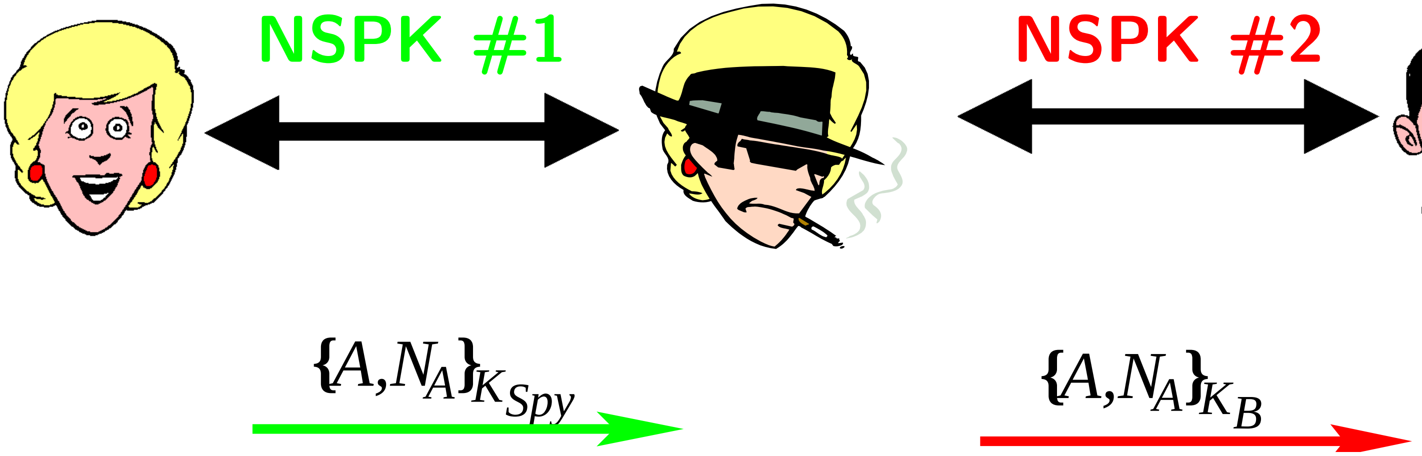
NSPK #1

NSPK #2

$\{A,N_A\}_{K_{Spy}}$

Man-in-the-Middle Attack

$$A \rightarrow B : \{A, N_A\}_{K_B}$$
$$B \rightarrow A : \ \{N_A, N_B\}_{K_A}$$
$$A \rightarrow B : \ \{N_B\}_{K_B}$$



NSPK #1

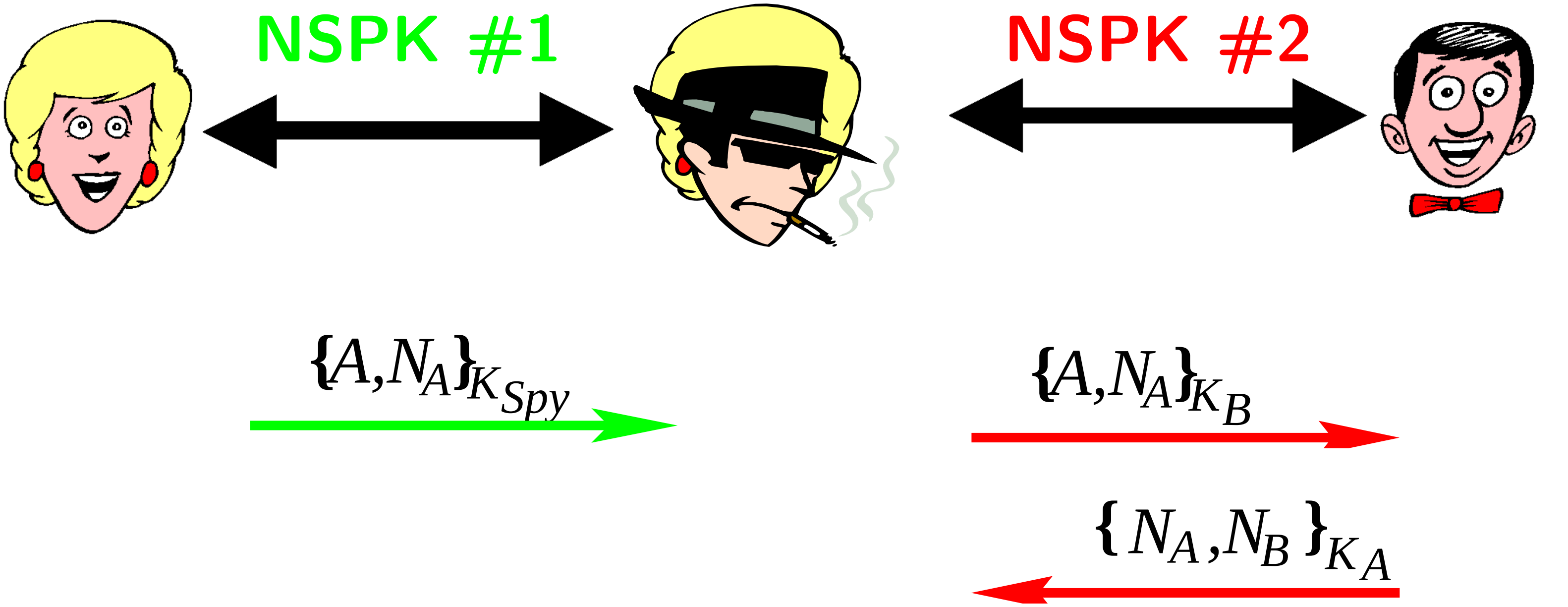
NSPK #2

$\{A, N_A\}_{K_{Spy}}$

$\{A, N_A\}_{K_B}$

Man-in-the-Middle Attack

$$A \rightarrow B : \{A, N_A\}_{K_B}$$
$$B \rightarrow A : \{N_A, N_B\}_{K_A}$$
$$A \rightarrow B : \{N_B\}_{K_B}$$



NSPK #1

NSPK #2

$\{A,N_A\}_{K_{Spy}}$

$\{A,N_A\}_{K_B}$

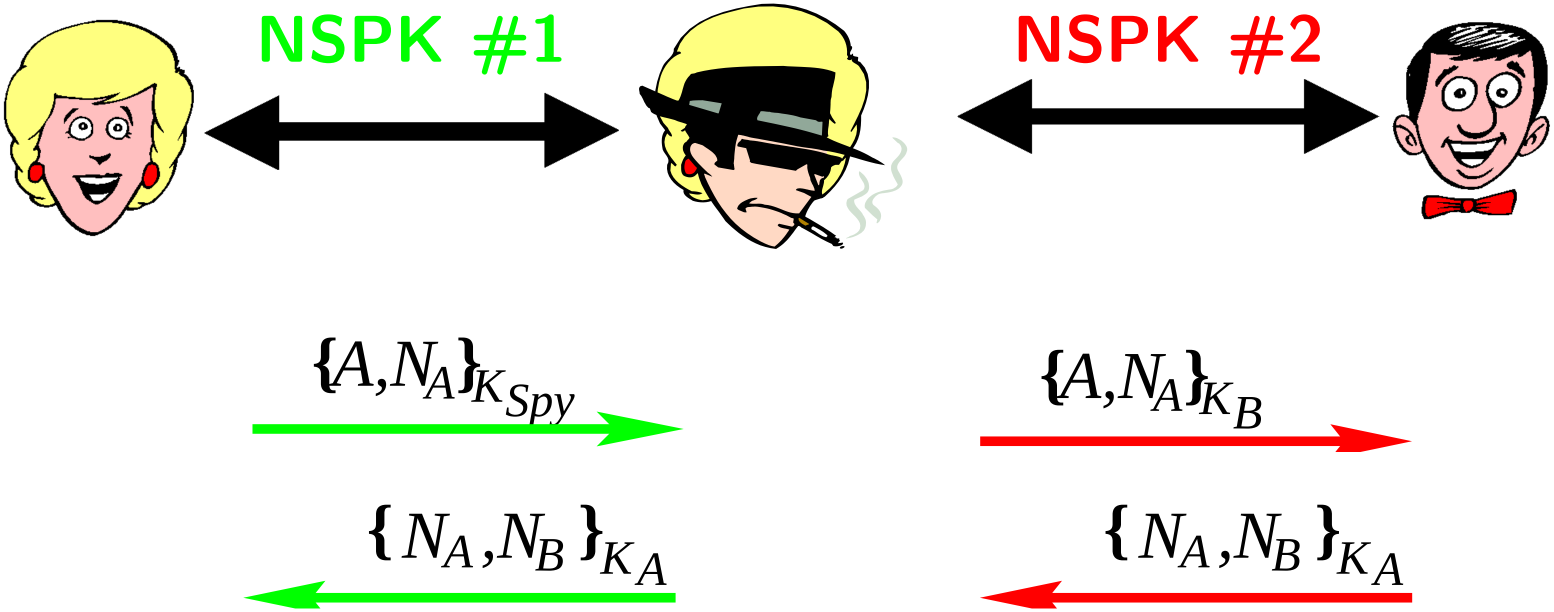
$\{N_A,N_B\}_{K_A}$

Man-in-the-Middle Attack

$$A \rightarrow B : \{A, N_A\}_{K_B}$$

$$B \rightarrow A : \{N_A, N_B\}_{K_A}$$

$$A \rightarrow B : \{N_B\}_{K_B}$$



NSPK #1

NSPK #2

$\{A,N_A\}_{K_{Spy}}$

$\{N_A,N_B\}_{K_A}$

$\{A,N_A\}_{K_B}$

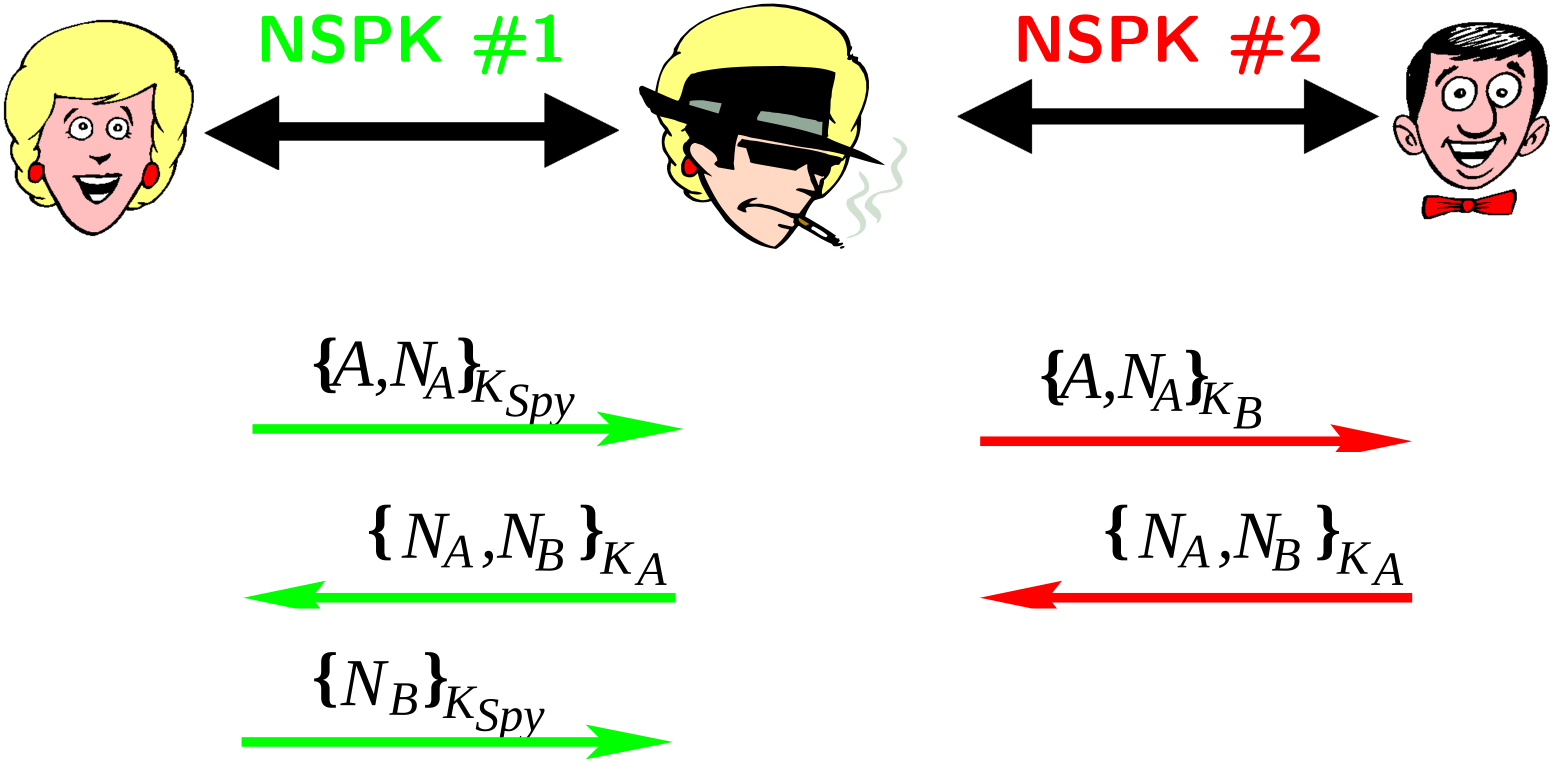
$\{N_A,N_B\}_{K_A}$

Man-in-the-Middle Attack

$A \rightarrow B : \{A, N_A\}_{K_B}$

$B \rightarrow A : \{N_A, N_B\}_{K_A}$

$A \rightarrow B : \{N_B\}_{K_B}$



NSPK #1

NSPK #2

$\{A,N_A\}_{K_{Spy}}$

$\{N_A,N_B\}_{K_A}$

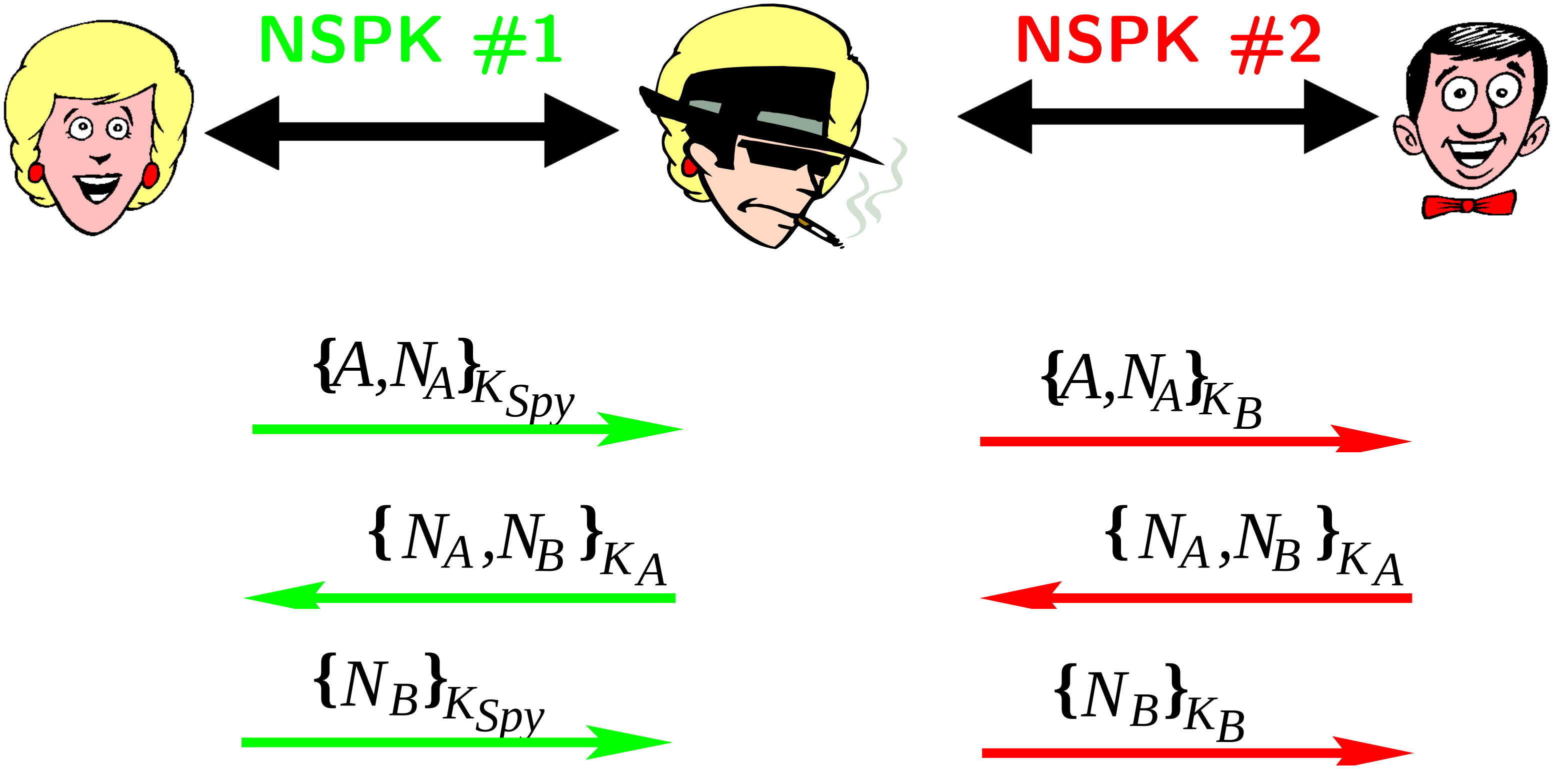
$\{N_B\}_{K_{Spy}}$

$\{A,N_A\}_{K_B}$

$\{N_A,N_B\}_{K_A}$

Man-in-the-Middle Attack

$$A \rightarrow B : \{A, N_A\}_{K_B}$$
$$B \rightarrow A : \{N_A, N_B\}_{K_A}$$
$$A \rightarrow B : \{N_B\}_{K_B}$$



B believes he is speaking with A!

What Went Wrong?

- Problem in step 2

$$B \rightarrow A : \{N_A, N_B\}_{K_A}$$

Agent B should also give his name: $\{N_A, N_B, B\}_{K_A}$.

- Is the improved version now correct?

A Formal Model

- Focus on communication traces.

$$A \longrightarrow B : \; M_1$$

$$B \longrightarrow A : \; M_2$$

$$\vdots$$

A Formal Model

- Focus on communication traces.

$$\begin{array}{c} A \rightarrow B : M_1 \\ C \rightarrow D : P_1 \\ B \rightarrow A : M_2 \\ D \rightarrow C : P_2 \\ \vdots \end{array}$$

A Formal Model

- Focus on communication traces.

$$
\begin{array}{l}
A \longrightarrow B : \; M_1 \\
C \longrightarrow D : \; P_1 \\
Spy \longrightarrow A : \; M_2 \\
D \longrightarrow A : \; P_2 \\
\qquad\quad \vdots
\end{array}
$$

A Formal Model

- Focus on communication traces.

$$\begin{array}{l} A \rightarrow B : M_1 \\ C \rightarrow D : P_1 \\ Spy \rightarrow A : M_2 \\ D \rightarrow A : P_2 \\ \quad\quad \vdots \end{array}$$

- Interleaving trace semantics: a protocol describes a set of traces.

  Interleaving of (partial) runs of the protocol and messages from the attacker.

A Formal Model

- Focus on communication traces.

$$\begin{array}{l} A \rightarrow B :\ M_1 \\ C \rightarrow D :\ P_1 \\ Spy \rightarrow A :\ M_2 \\ D \rightarrow A :\ P_2 \\ \quad\vdots \end{array}$$

- Interleaving trace semantics: a protocol describes a set of traces.

  Interleaving of (partial) runs of the protocol and messages from the attacker.

- Example: Needham-Schroeder is the smallest set P where:

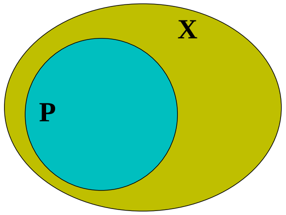
0.	$\langle\rangle \in P$	
1.	$t,\ A \rightarrow B : \{A, N_A\}_{K_B} \in P$	if $t \in P$ and $fresh_t(N_A)$
2.	$t,\ B \rightarrow A : \{N_A, N_B\}_{K_A} \in P$	if $t \in P$, $fresh_t(N_B)$, and $A' \rightarrow B : \{A, N_A\}_{K_B} \in t$
3.	$t,\ A \rightarrow B : \{N_B\}_{K_B} \in P$	if $t \in P$, $A \rightarrow B : \{A, N_A\}_{K_B} \in t$ and $B' \rightarrow A : \{N_A, N_B\}_{K_A} \in t$
4.	$t,\ Spy \rightarrow B : X \in P$	if $t \in P$ and $X \in synthesize\ (analyze\ (sees\ t))$

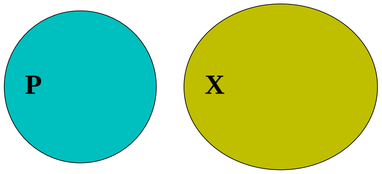
Modeling (cont.)

- A property also correspond to set of traces.
  Authentication for A: If (1) A used N_A to start a protocol run and with B (2) received N_A back, then B sent N_A back.

$$\begin{array}{lcll}
AauthenticatesB(t) & \equiv & \textbf{If} & A \rightarrow B : \{A, N_A\}_{K_B} \in t \ \ \textbf{and} \\
 & & & B' \rightarrow A : \{N_A, N_B\}_{K_A} \in t \\
 & & \textbf{then} & B \rightarrow A : \{N_A, N_B\}_{K_A} \in t \\
SpyattacksA(t) & \equiv & \neg AauthenticatesB(t) &
\end{array}$$

- Hence the correctness of protocols has an exact meaning.
  Every [no] trace of the protocol P has property X.

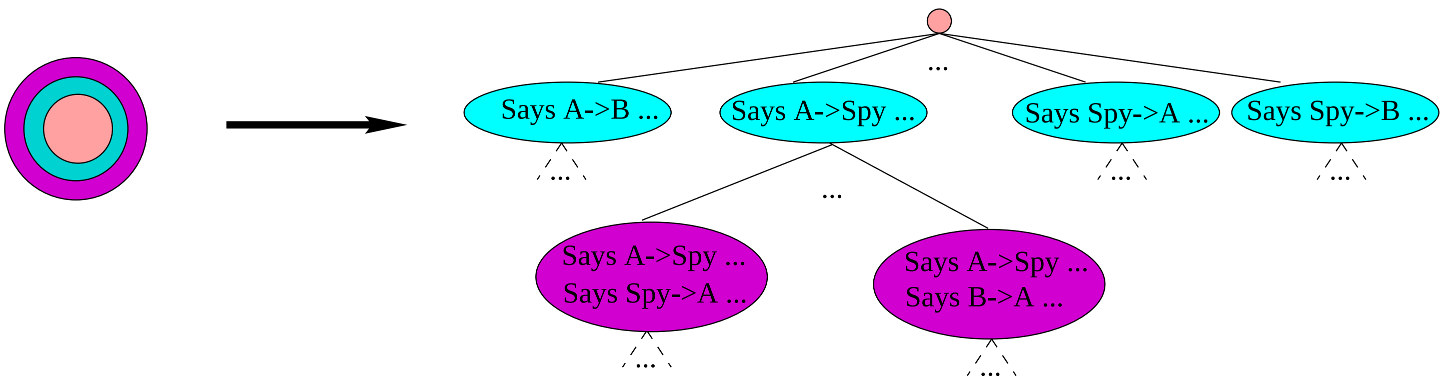




- Every proposition is either true or false.
  How do we determine which holds?

Finding Flaws using State Enumeration

- Inductive definition corresponds to an infinite tree.



- Properties now correspond to a subset of nodes, e.g., $SpyattacksA(t)$.
- State enumeration can be used to find attacks in the infinite tree.
- Challenge: Naive search is hopeless!

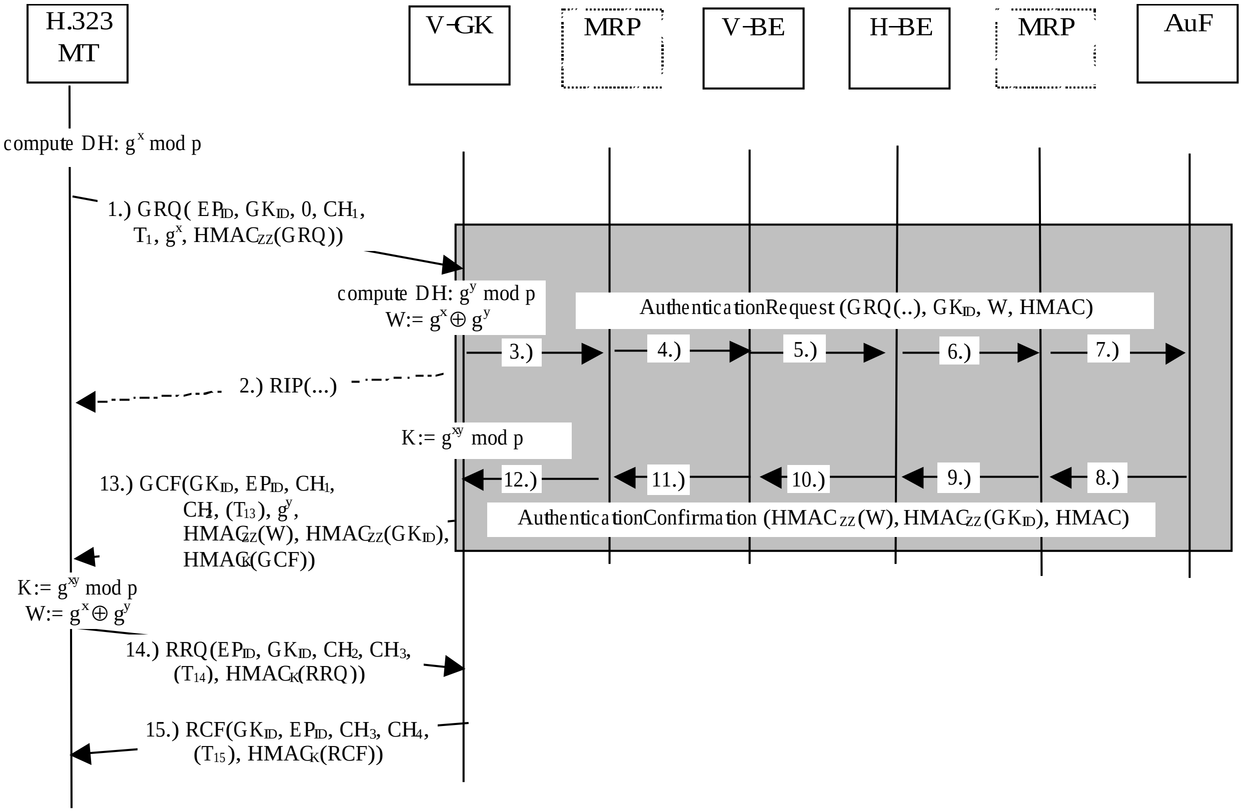
  Solutions involve advances in algorithms and data structures for searching very large states spaces. (See `www.inf.ethz.ch/~basin`)

OFMC/AVISPA Tool

- Ideas implemented in the On-the-Fly Model-Checker.
  - Rich language for specifying security protocols and properties.
  - Supports symmetric and asymmetric keys, cryptographic hash functions, key-tables, user-definable algebraic functions, etc.

Input	Output (<1 second)
PROTOCOL Needham-Schroeder; Identifiers A, B: user; Na, Nb: nonce; Ka, Kb: public_key; Messages 1. A -> B: {A,Na}Kb 2. B -> A: {Na,Nb}Ka 3. A -> B: {Nb}Kb Intruder_knowledge Spy, a, b, ka, kb, kspy; Goal correspondence_between A B;	A -> Spy: {A,Na}Kspy Spy -> B: {A,Na}Kb B -> A: {Na,Nb}Ka A -> Spy: {Nb}Kspy Spy -> B {Nb}Kb

H.530 — Mobile Multi-media Protocol



H.323 MT

V-GK

MRP

V-BE

H-BE

MRP

AuF

compute DH: g^x mod p

1.) GRQ(EP_{ID}, GK_{ID}, 0, CH_1, T_1, g^x, $HMAC_{ZZ}$(GRQ))

compute DH: g^y mod p

W:= $g^x \oplus g^y$

AuthenticationRequest (GRQ(..), GK_{ID}, W, HMAC)

3.)

4.)

5.)

6.)

7.)

2.) RIP(...)

K:= g^{xy} mod p

12.)

11.)

10.)

9.)

8.)

13.) GCF(GK_{ID}, EP_{ID}, CH_1, CH_2, (T_{13}), g^y, $HMAC_{ZZ}$(W), $HMAC_{ZZ}$(GK_{ID}), $HMAC_K$(GCF))

AuthenticationConfirmation ($HMAC_{ZZ}$(W), $HMAC_{ZZ}$(GK_{ID}), HMAC)

K:= g^{xy} mod p

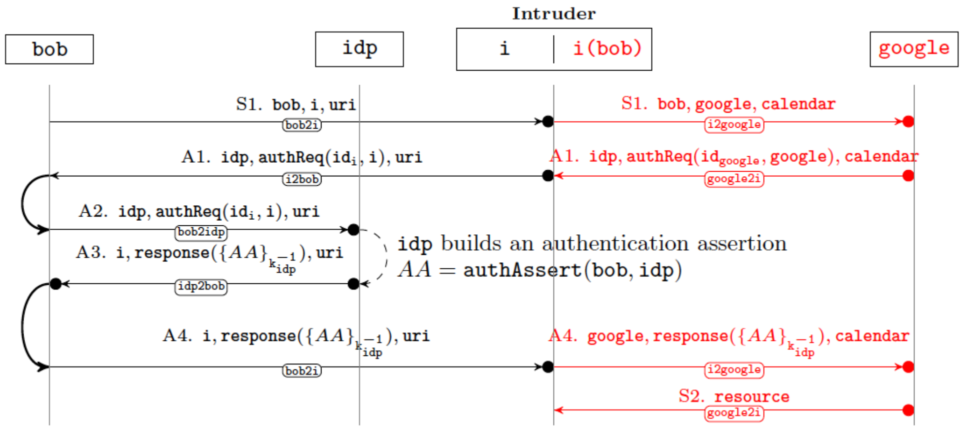
W:= $g^x \oplus g^y$

14.) RRQ(EP_{ID}, GK_{ID}, CH_2, CH_3, (T_{14}), $HMAC_K$(RRQ))

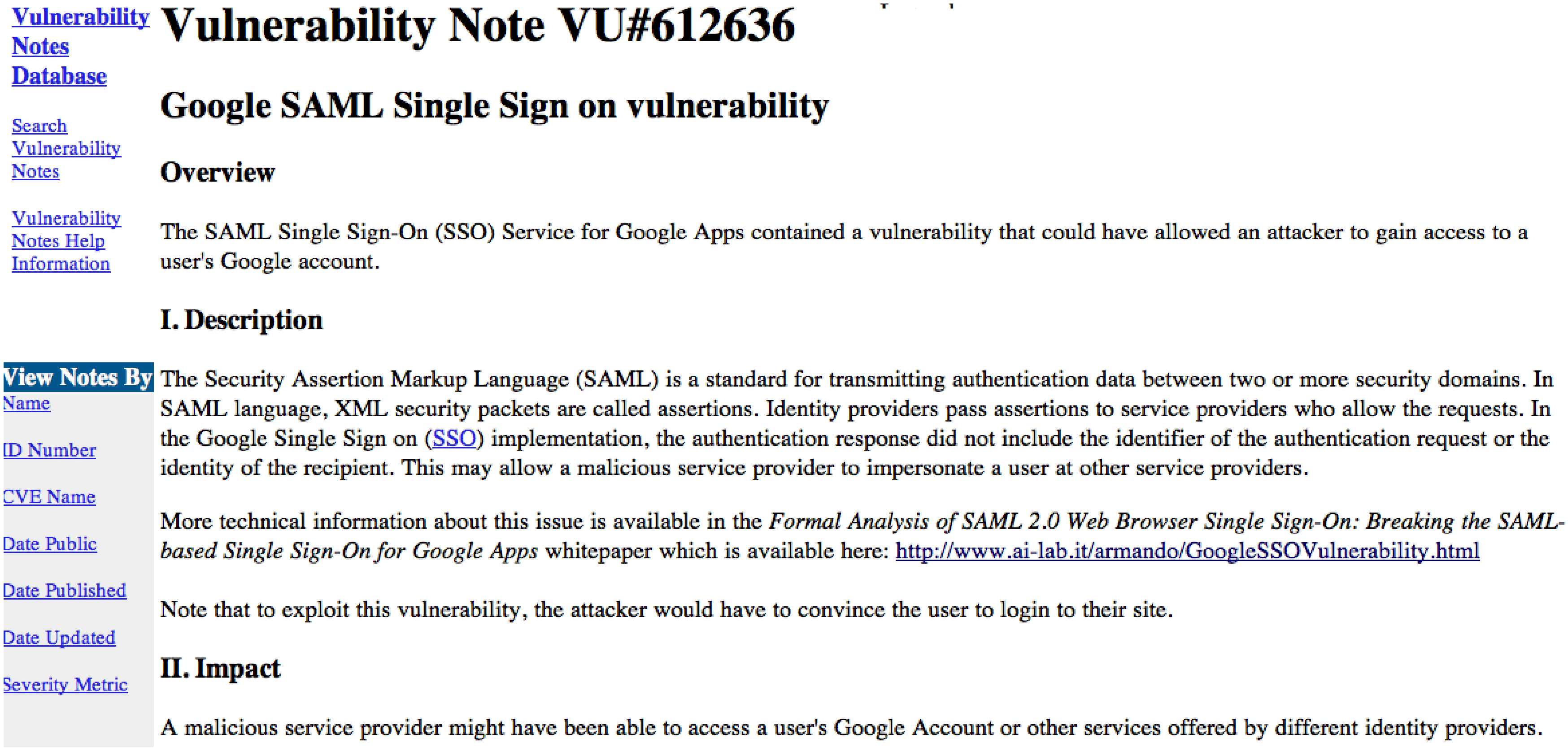
15.) RCF(GK_{ID}, EP_{ID}, CH_3, CH_4, (T_{15}), $HMAC_K$(RCF))

- Protocol developed by Siemens, ca. 1 year
- Flaw found using OFMC, ca. 1 day
- New design/patents/etc., ca. 1 year

Google Single Sign On


Intruder
bob
idp
i
i(bob)
google
S1. bob, i, uri
bob2i
S1. bob, google, calendar
i2google
A1. idp, authReq(id_i, i), uri
i2bob
A1. idp, authReq(id_{google}, google), calendar
google2i
A2. idp, authReq(id_i, i), uri
bob2idp
A3. i, response($\{AA\}_{k_{idp}^{-1}}$), uri
idp2bob
idp builds an authentication assertion
AA = authAssert(bob, idp)
A4. i, response($\{AA\}_{k_{idp}^{-1}}$), uri
bob2i
A4. google, response($\{AA\}_{k_{idp}^{-1}}$), calendar
i2google
S2. resource
google2i

US-CERT
UNITED STATES COMPUTER EMERGENCY READINESS TEAM



Vulnerability Notes Database

Search Vulnerability Notes

Vulnerability Notes Help Information

View Notes By

Name

ID Number

CVE Name

Date Public

Date Published

Date Updated

Severity Metric

Vulnerability Note VU#612636

Google SAML Single Sign on vulnerability

Overview

The SAML Single Sign-On (SSO) Service for Google Apps contained a vulnerability that could have allowed an attacker to gain access to a user's Google account.

I. Description

The Security Assertion Markup Language (SAML) is a standard for transmitting authentication data between two or more security domains. In SAML language, XML security packets are called assertions. Identity providers pass assertions to service providers who allow the requests. In the Google Single Sign on (SSO) implementation, the authentication response did not include the identifier of the authentication request or the identity of the recipient. This may allow a malicious service provider to impersonate a user at other service providers.

More technical information about this issue is available in the *Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps* whitepaper which is available here: http://www.ai-lab.it/armando/GoogleSSOVulnerability.html

Note that to exploit this vulnerability, the attacker would have to convince the user to login to their site.

II. Impact

A malicious service provider might have been able to access a user's Google Account or other services offered by different identity providers.

Vulnerability found using AVISPA tool in 2008.

Summary of Protocol Example

- **Information Security as a Science:** Example used Formal Methods to make precise statements about system security.

  In this case, about the security of protocols with respect to a particular model (of cryptography, intruder powers, ...).

- Practical relevance: reducing protocol design errors. Benefits:

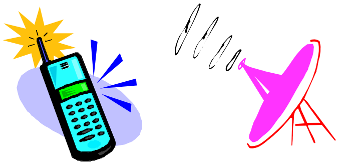
  Money: security updates are costing hundreds of millions of CHF.

  Time: protocols are delayed by years.

  Acceptance: eroding confidence in Internet and new applications.

- AVISPA tool used worldwide.
  100+ downloads and used for teaching at 10+ Universities.

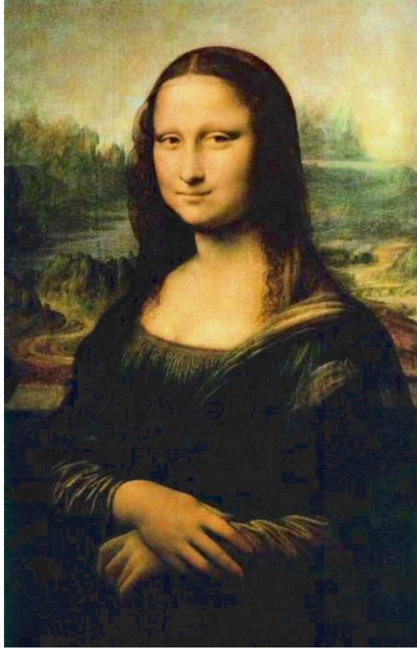
Scope of Ideas



- Tool supported protocol standardization (AVISPA/AVANTSSAR)
- Model Driven Security: generating security architectures from security-design models
- Usage Control
  - Specification language for usage control requirements
  - Associated enforcement monitors and architectures
- Specification-based security testing
- Machine-learning techniques for access control

See **www.infsec.ethz.ch** for more on these topics.

Art or Science?



- Some areas of Information Security will always remain an art. Others are inherently imprecise.

  This is often the case when humans are in-the-loop, e.g., security policy definition or intrusion detection.

- But in many cases it is possible and desirable to apply rigorous scientific methods to construct and analyze secure systems.

  $\Longrightarrow$ Requires work in foundations, tool support, and applications.

- Scope of methods is wide.

Happy Birthday Günter

Danke Günter

- For supporting me during my start-up in Information Security.
- For the joy of co-teaching and co-organization in Freiburg.
- For inspiring me, within and outwidth research.
- For your generosity.
- For your friendship.