Sample attacks

Protocol 1:
(1) $A \to B$: $E_B(M)$
(2) $B \to A$: $E_A(M)$

Attack 1:
(1) $A \to B$: $E_B(M)$, inter
(2) $Z \to B$: $E_B(M)$
(3) $B \to Z$: $E_Z(M)$; Z de

Protocol 2:
(1) $A \to B$: $E_B(E_B(M)A)$
(2) $B \to A$: $E_A(E_A(M)B)$

Attack 2:
(1) $Z \to A$: $E_A(E_A(E_A(M$
(2) $A \to Z$: $E_Z(E_Z(E_A(M$
(3) Z decodes $E_A(M)B$, c
(4) $Z \to A$: $E_A(E_A(M)Z)$
(4) $A \to Z$: $E_Z(E_Z(M)A)$

Symmetric and asymmetric key encr

A fundamental problem: establishing a two-way secre
channel between two entities

Symmetric key encryption (secret key, shared key)
– shared key known only by the two participants
– decryption and encryption key related by a simple t
(conventionally considered as being the same)
– examples: Data Encryption Standard (1975, outdat
AES (Rijndael)

Asymmetric key encryption (public key)
– each participant A: has a pair of keys, one is invers
– publică key, K_a / private key K_a^{-1}
– A sends $K_b(K_a^{-1}(M))$: only A can create, only B c
– ex. Rivest-Shamir-Adleman (1976), El-Gamal

Verification of security proto

January 19, 2006

– models of security protocols
– typical examples of protocols and attacks
– modeling in BAN logic
– verification methods

The Dolev-Yao intruder model

Intruders are "active": they can eavesdrop on the co
acquiring messages and doing everything possible to
An intruder:
a) can obtain every message from the network
b) is a legitimate user of the system; in particular, s/
conversation with any user
c) will have the opportunity to receive messages from
(more generally: any user B can become recipient fo

Protocol Models

[Dolev & Yao '83]: stressed the importance of clea
modeling, analysis and verification of protocols:

1) In a public key encryption system:
a) encryption functions cannot be broken
b) public key directory has guaranteed identity
c) everybody has access to all public keys E_X, $\forall X$
d) only X has access to its decryption key D_X

2) A protocol between two entities does not require
a third for encryption or decryption

3) In a uniform protocol, all communicating parties us
sage format

Security Protocols and their Import

Need for secret communication dates back to antiqui
likewise the discovery of ciphers and beginnings of cr

Security involves multiple aspects:
authentification, authorization, integrity, confidentia
tion.

Solutions are complex and reasoning them about the
Security of a protocol must not depend on the secrecy
(no "security through obscurity")
– Subtle errors in existing (and widely used) protoco
covered after very long time (17 years, for Lowe's atta
Schroeder)
⇒ there are great risks in compromising a weak alg
and thus unscrutinized by specialists
⇒ importance of formal verification even greater

Needham-Schroeder: attack

[Lowe '95] finds an error in the public key version (af
(1) $A \to B$: $A, B, \{N_a, A\}_{K_b}$ A asks to communicate
(2) $B \to A$: $B, A, \{N_a, N_b\}_{K_a}$ B replies with nonce N
(3) $A \to B$: $A, B, \{N_b\}_{K_b}$ A confirms reception

Attack with two concurrent sessions: A initiates sessio
I; the latter impersonates A in session β with B
(α.1) $A \to I$: $A, I, \{N_a, A\}_{K_i}$
(β.1) $I(A) \to B$: $A, B, \{N_a, A\}_{K_b}$
(β.2) $B \to I(A)$: $B, A, \{N_a, N_b\}_{K_a}$
(α.2) $I \to A$: $I, A, \{N_a, N_b\}_{K_a}$
(α.3) $A \to I$: $A, I, \{N_b\}_{K_i}$
(β.3) $I(A) \to B$: $A, B, \{N_b\}_{K_b}$

Discovered: with FDR model checker for CSP langua
Correction: including the encrypted name of the sende

Needham-Schroeder shared key pro

(1) $A \to S$: A, B, N_a
A announces to server S the intention to communica
(and guarantees freshness of the message with a non
(2) $S \to A$: $\{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
S sends to A the key K_{ab}, together with an encrypte
A will retransmit to B:
(3) $A \to B$: $\{K_{ab}, A\}_{K_{bs}}$
B extracts the key K_{ab} and announces A by sending a
(4) $B \to A$: $\{N_b\}_{K_{ab}}$
A confirms by retransmitting a message based on N_b
(conventionally, decremented by 1 to avoid replay)
(5) $A \to B$: $\{N_b - 1\}_{K_{ab}}$
Now, both participants know they can communicate

First formal results

Dolev & Yao discuss two types of protocols, defined
operations:

1. Cascade protocols
– encryption with any public key
– decryption only with own key

2. Name stamp protocols. In addition:
– appending a participant's name to a messaje
– deleting a certain participant's name
– deleting any name

Correctness problem becomes a rewriting problem fo
alphabet, decidable in polyomial time
– but undecidable for more complex problems

Types of attacks

"Cryptography is not broken, it is circumvented" – A

Clark & Jacob, "A Survey of Authentication Protocol Literature
• freshness attacks (replay attacks)
– a message (or fragment) from an earlier communi
stored an inserted by the intruder in a new session

• type flaw attacks
– A message is composed of fields, each with a given
(data, nonce, participant name, key value)
– Attack based on accepting a message with another
bit pattern than the one initially sent

Needham-Schroeder: attack

[Denning & Sacco, 1981]
Problem: an intruder who eavesdropped on a previous
B to accept an old key, potentially compromised

Intruder I impersonates A (denoted $I(A)$) and sends t
from the earlier session, with the old key K_c:

(3) $I(A) \to B$: $\{K_c, A\}_{K_{bs}}$
(4) $B \to I(A)$: $\{N_b\}_{K_c}$
(5) $I(A) \to B$: $\{N_b - 1\}_{K_c}$

Danger: I has practically unlimited time to comprom

Correction: timestamps or extra nonce

Authentication protocols

protocols by which participants convince each other c
– and either establish shared secrets (keys) for comm
– or recognize the use of partners' secret keys

– are the most widely studied security protocols in th

Notations: A, B: participants. S: authentication serv
N_a, N_b: "nonce" (from: number once) = random p
generated to avoid reuse of old messages by an intru
$\{X\}_K$: message X cripted with key K

[Needham & Schroeder '78] "Using Encryption for A
Large Networks of Computers": classic article, the fi
importance of formal verification methods

BAN logic: inference rules (cont

Jurisdiction rule:

$$\frac{P \models Q \Rightarrow X,\ P \models Q \models X}{P \models X}$$

Composition: P believes a compound $\Leftrightarrow$ believes the

Projection: P said a compound $\Rightarrow$ said the parts

$$\frac{P \models X,\ P \models Y}{P \models (X,Y)} \quad \frac{P \models (X,Y)}{P \models X} \quad \frac{P \models Q \sim (X}{P \models Q \sim}$$

Decryption rules:

$$\frac{P \models \overset{K}{\mapsto} Q,\ P \lhd \{X\}_{K^{-1}}}{P \lhd X}$$

Bidirectionality of keys and secrets among participant

$$\frac{P \models R \overset{K}{\leftrightarrow} R'}{P \models R' \overset{K}{\leftrightarrow} R}$$

BAN logic: basic notions

$P \models X$	P believes X
$P \lhd X$	P sees (receives, reads) message X
$P \mid\sim X$	P said (sent) X sometime in the past
$P \Rightarrow X$	P has jurisdiction over X. P is an authority on X (e.g., a key) and must be believed
$\sharp(X)$	X is fresh (has not been sent so far)
$P \overset{K}{\leftrightarrow} Q$	P and Q can use shared key K to communi
$\overset{K}{\mapsto} P$	P has public key K
$P \overset{X}{\rightleftharpoons} Q$	X is a secret known only by P and Q
$\{X\}_K$	message X encrypted with key K
$\langle X \rangle_Y$	X combined with secret Y (for identificatio

Types of attacks (cont'd.)

- patallel session attacks
  – two or more concurrent sessions of the same proto
  – messages from a session used to attack another
- implementation-dependent attacks
  – type flaw attacks can be eliminated if the represent components contains redundancy to distinguish the t
  – interaction between protocol and encryption metho of a bit in bitwise encryption)
- binding attacks (key integrity attacks)
  – tampering with partner's public key (replacing it wit
- ... and many others

Example: Needham-Schroeder with sha

(1) $A \to S$: A, B, N_a
(2) $S \to A$: $\{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$
(3) $A \to B$: $\{K_{ab}, A\}_{K_{bs}}$
(4) $B \to A$: $\{N_b\}_{K_{ab}}$
(5) $A \to B$: $\{N_b - 1\}_{K_{ab}}$

We idealize the protocol: instead of bit messages, *formulas*, corresponding to message meaning:

(1) Message 1 is only a request, has no logic value
(2) $S \to A$: $\{N_a, B, (A \overset{K_{ab}}{\leftrightarrow} B), \sharp(A \overset{K_{ab}}{\leftrightarrow} B), \{A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}\}$
(3) $A \to B$: $\{A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{bs}}$
(4) $B \to A$: $\{N_b, (A \overset{K_{ab}}{\leftrightarrow} B)\}_{K_{ab}}$ from B
(5) $A \to B$: $\{N_b, (A \overset{K_{ab}}{\leftrightarrow} B)\}_{K_{ab}}$ from A

BAN logic: inference rules

Rules on meaning of messages:

– for shared keys:

$$\frac{P \models Q \overset{K}{\leftrightarrow} P,\ P \lhd \{X\}_K}{P \models Q \mid\sim X}$$

– for public keys:

$$\frac{P \models \overset{K}{\mapsto} Q,\ P \lhd \{X\}_{K^{-1}}}{P \models Q \mid\sim X}$$

– for shared secrets:

$$\frac{P \models Q \overset{Y}{\rightleftharpoons} P,\ P \lhd \langle X \rangle_Y}{P \models Q \mid\sim X}$$

Rules on freshness of messages

$$\frac{P \models \sharp(X),\ P \models Q \mid\sim X}{P \models Q \models X} \qquad \frac{P \models \sharp(X)}{P \models \sharp(X,Y)}$$

Modeling in BAN logic

[Burrows, Abadi, Needham '89: "A logic of authentic

– most important method for modeling using logic
– a *logic of belief*, as opposed to a *logic of knowledg*
– deals with what every participant *believes* is true

Goal: to express precisely
– initial assumptions about the workings of a protocc
– the final conclusions reached by the participants

Examples:
– what does the protocol achieve ?
– does it need more assumptions than another proto
– does it send/encrypt something which is not neces

Verification: theorem proving

Model checking requires finite model ⇒ finite particip
Theorem provers do not have this limitation

Rewriting of reasoning in Prolog: Interrogator [Miller
NRL Protocol Analyzer [Meadows et al.]
– combination of theorem-proving + model checking
– starts from an error state (should be inaccessible)
– searches backwards using inductive techniques

Athena [Song et al., CMU/Berkeley]
– representation using *strand spaces* based on causal
vidual executions ⇒ reduces state space significantly

Verification: model checking

Protocol is asynchronous composition b/w participan
Intruder can listen to anything, and delete, change or
according to its current knowledge set.

State space is given by execution point for each parti
knowledge set of intruder (set of terms constructed b

Intruder is modeled by a relation $\vdash$ by which the int
messages m from an initial set of information I:
– if $m \in I$ then $I \vdash m$
– concatenation: if $I \vdash m_1$ and $I \vdash m_2$ then $I \vdash m_1 \cdot m$
– projection: if $I \vdash m_1 \cdot m_2$ then $I \vdash m_1$ and $I \vdash m_2$
– encryption: if $I \vdash m$ and $I \vdash k$ then $I \vdash \{m\}_k$
– decription: if $I \vdash \{m\}_k$ and $I \vdash k^{-1}$ then $I \vdash m$

Model checkers: FDR (for CSP), OFMC (on-the-fly)
based)

Reasoning: Needham-Schroeder with sh

We start with the assumptions (denote $P \models X$ and $Q \models$
$A, S \models A \overset{K_{as}}{\leftrightarrow} S \quad B, S \models B \overset{K_{bs}}{\leftrightarrow} S \quad S \models A \overset{K_{ab}}{\leftrightarrow} B$
$A, B \models (S \models\!\!> A \overset{K}{\leftrightarrow} B) \quad A \models (S \models\!\!> \sharp(A \overset{K}{\leftrightarrow} B)) \quad$ (a g
$A \models \sharp(N_a) \quad B \models \sharp(N_b) \quad S \models \sharp(A \overset{K_{ab}}{\leftrightarrow} B)$
From $A \models \sharp(N_a)$, and (2) we deduce:
$A \models S \models A \overset{K_{ab}}{\leftrightarrow} B,\ A \models S \models \sharp(A \overset{K_{ab}}{\leftrightarrow} B)$
and from the jurisdiction rule: $A \models A \overset{K_{ab}}{\leftrightarrow} B \quad A \models \sharp($
After receiving message (3) from A, we deduce: $B \models$
We cannot obtain $B \models A \overset{K_{ab}}{\leftrightarrow} B$ without the premise B
From the freshness of messages (4) and (5) we de
$A \overset{K_{ab}}{\leftrightarrow} B$ and $B \models A \models A \overset{K_{ab}}{\leftrightarrow} B$, thus each participant
that the key is valid, and that the other participant k

Reasoning explicitates the missing premise, which al
to substitute a compromised key.

Verification: theory generation

Problem in model checking: representing potentially in
Theory generation (RVChecker, REVERE [Kindred&V
BAN logic)
– a syntactic method of saturation-based theorem pr
– produces a finite representation of a potentially infi
(all theorems generated from some premises and rule
– termination based on limiting the application of thos
which can generate conclusions of larger size than pr

Combination with model checking:
– a dubious premise found by theory generation: used
attack
– conversely, a counterexample, modeled as logic ded
tify a dubious premise

BAN logic: applicability and limitat

– allows to prove properties about a protocol
– if property can't be proved, there are serious reaso
– can identify dubious/missing/non-explicit premises

But:
– monotone logic: an existing fact cannot be retract
– cannot handle the notion of key confidentiality or
promised (e.g. sending a key in plaintext)