15-827 Security and Cryptography
Lecture 6

Models and Tools for Reasoning About Security

Jeannette M. Wing

Plan for Today

- Historical perspective
  - Security models
    - NSA, Orange Book, theorem provers of the 70's
- Current approaches
  - Theorem proving (proof checking, really)
  - Model checking
  - Hybrid approach
- General Remarks

Goals for Today

- How can you use X to reason about security protocols? What are their strengths and limitations?
  - X = theorem prover
  - X = model checker

Taxonomy of Approaches

	Math	Tools
Logics	BAN, GNY, SVO	InaJo, HDM, FDM, Gypsy, PVS, **Isabelle**, Coq, **Revere** (Kindred)
State mach.	Abadi-Tuttle	**FDR, Murϕ** **Brutus**, Interrogator
Algebras	Dolev-Yao	**NRL Protocol Analyzer**

NSA and Formal Verification

- NSA (NCSC) was a major source of funding for formal verification systems in the US in the mid-70's to mid-80's.

- The focus was on building **security kernels** relying on use of a *reference monitor* [Anderson72], which is supposed to satisfy three properties:
  - It must mediate all accesses to objects.
  - It must be tamperproof.
  - It must be small enough to be subjected to *analysis* and test.

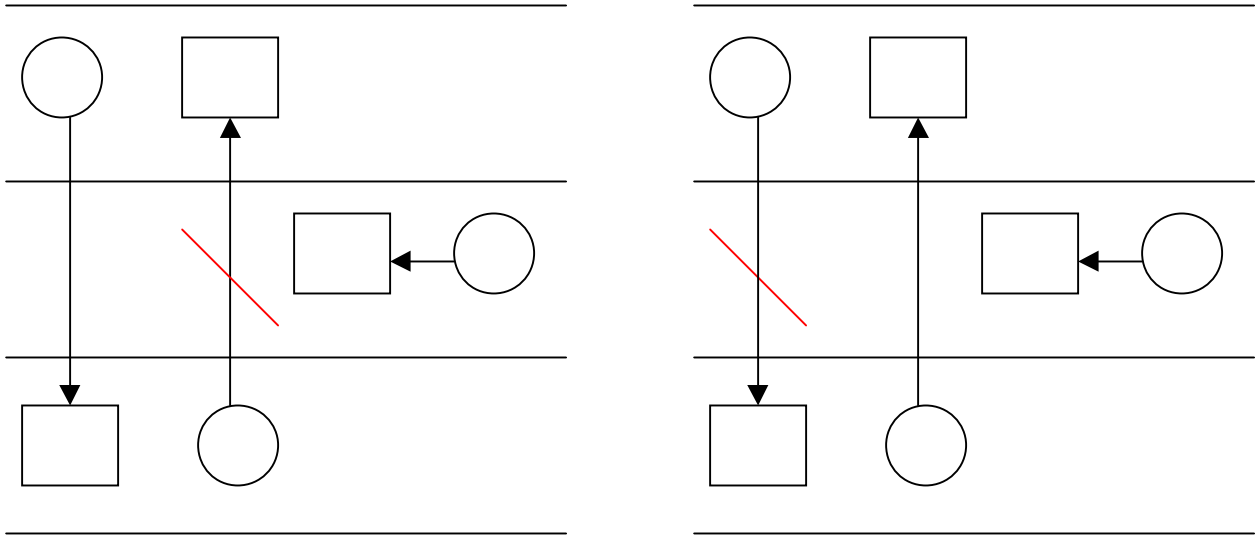
- One form of "*analysis*" is formal verification**.**

Three Security "Models"

- Bell-LaPadula
- Biba integrity
- Clark-Wilson

Bell-LaPadula Model

- Bell-LaPadula model (MITRE)
  - Subjects (e.g., users), objects (e.g., files), access modes (e.g., read, write)
- Pictorially



read write

Simple-security and * Properties

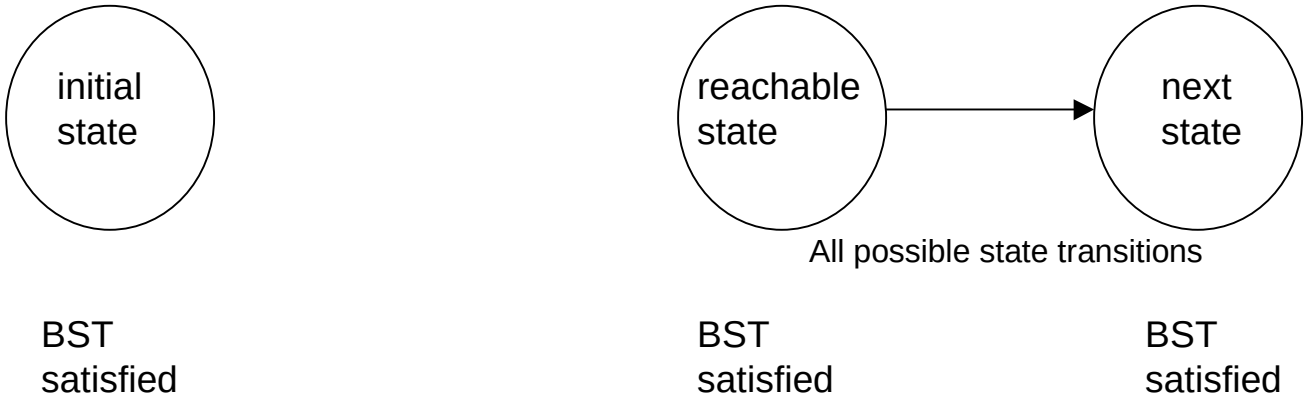
Rationale: Prevents untrusted subjects from simultaneously having read access to information at one level and write access to information at a lower level. I.e.,

No information leakage

- Read down and write up ok
  - Jim (confidential) needs to be able to read data from my (unclassified) files to make a salary decision (confidential).
- **Simple security property** (NRU)
  - Why not? Then I (unclassified) can see Satya's salary (confidential).
- ***-property** (NWD)
  - Why not? Don't want Jim to leak Satya's salary (confidential) to me (unclassified).

What Does it Mean for a System to be “Secure”?

- *Basic Security Theorem*
  - A state is *secure* if it satisfies the simple security property, the *-property (and one other property).
  - A system is *secure* if and only if the initial state is secure and the states before and after each action are secure.
  - Use induction to prove it:



BLP Controversy

- Blind writes allowed, but “nonsense” for O/S.

  $ ls

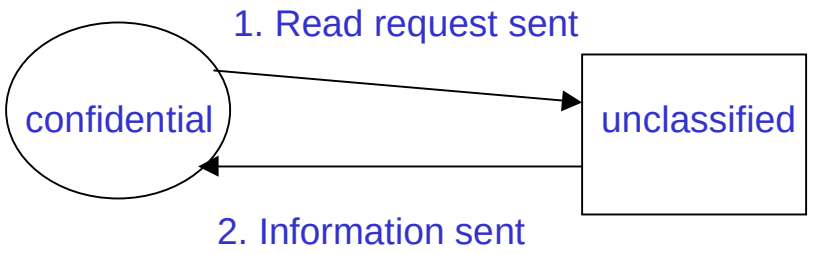
  A

  $ cp A B

  $ cat B

  cat B: not allowed

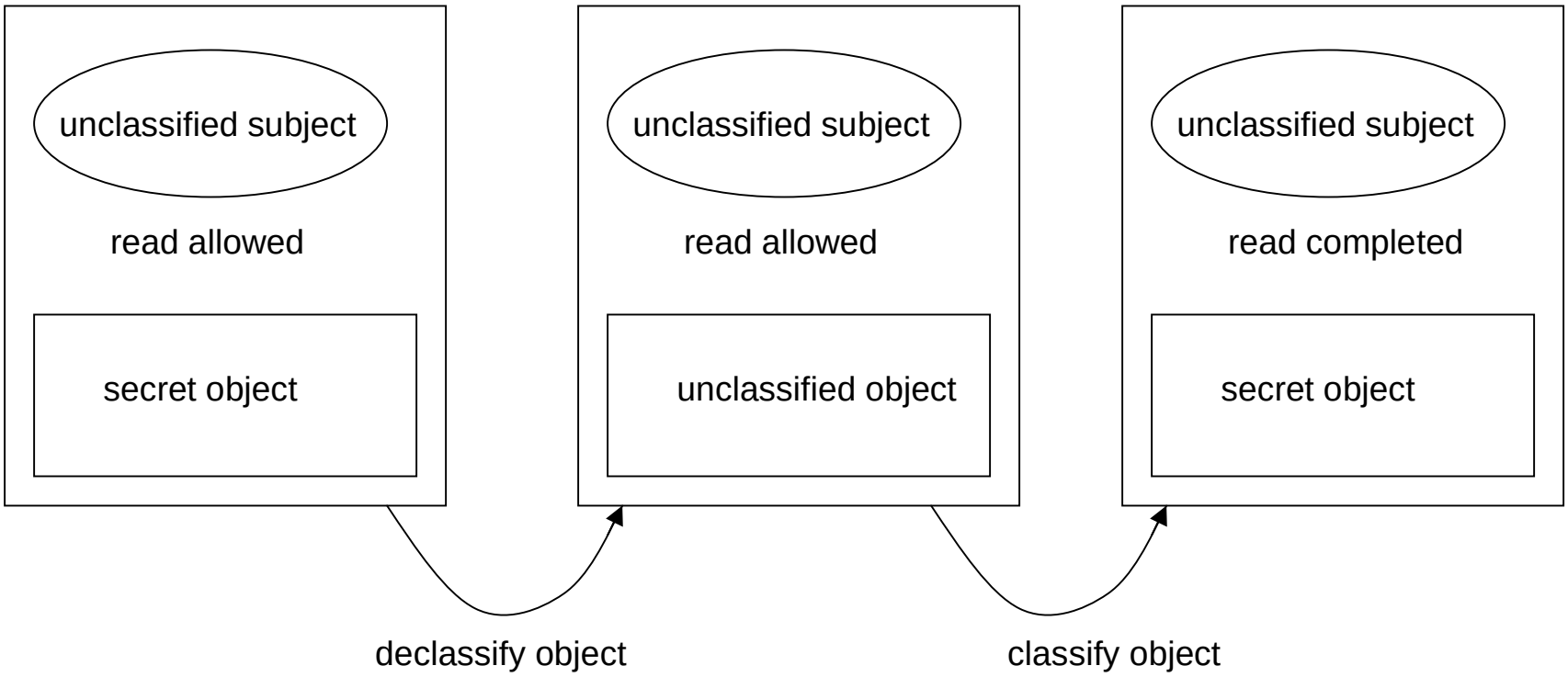
- Remote reads not allowed, but necessary for distributed systems: To do 2 (legal read up), need to do 1 (illegal write down)



Nail in Coffin: McLean's System Z

- In IPL '85 McLean showed that you can prove an obviously insecure system (which satisfies the reverse of the *-property) secure.
    - This dagger-property allows subjects to transfer information from higher security levels to lower security levels.
- In a draft paper McLean constructed an obviously insecure model called System Z that satisfies BST.
    - System Z had only one action that downgrades all subjects and objects to the lowest possible level.
- Conclusion: "We have shown that the BST does nothing to establish that a system is really secure."

System Z


unclassified subject
read allowed
secret object
unclassified subject
read allowed
unclassified object
unclassified subject
read completed
secret object
declassify object
classify object
BLP-secure state
BLP-secure state
BLP-secure state

Biba Integrity Model

- In BLP: NRU, NWD
  - Writes up and reads down could introduce integrity problems
    - I can write over Jim's salary database.
- Biba of MITRE proposed a class of integrity models with the opposite rules.
  - Mandatory Integrity Model: NRD, NWU
    - Disclosure and integrity labels are used in opposite ways in a mixed BLP and Biba system.
  - Subject Low-Water Mark Model
    - Relax NRD: Allow subjects to read down, but first lower their integrity level to that of the object being read.
  - Object Low-Water Mark Model
    - Relax NWU: Lower objects level to that of the subject doing the write.

Homework

Can you think of a situation where it makes sense to have a combination of the BLP and Biba mandatory integrity models?

Clark-Wilson Integrity Model

- Data partitioned into *constrained* and *unconstrained* data items.
- Set of nine rules, based on business “best practices”
  - Rules constrain how integrity of CDIs must be validated, how and by whom CDIs can be changed, and how UDIs can be changed to CDIs
    - Rule 2: Application of a Transformation Procedure (TP) to any CDI must maintain the integrity of that CDI.
    - Rule 4: Subjects can only initiate certain TPs on certain CDIs.
- Can be combined with the Biba model in a practical computer system environment.

The Orange Book

- Hierarchy of security levels
  - D: Minimal
    - Reserved for systems that have been evaluated but failed to meet requirements of higher evaluation classes
  - C: Discretionary (3 classes)
    - Provide “need to know” protection and accountability of subjects and their actions.
    - Like UNIX access control. I.e., users have control.
  - B: Mandatory (2 classes)
    - System, not a user, labels objects according to a fixed policy.
    - Requires system developer to state policy model, provide a spec of TCB, and *evidence to demonstrate the reference monitor has been implemented correctly***.**
  - A: Verified
    - Like B except requires use of **“formal security verification methods”** for the *evidence*.
    - MITRE SCOMP: 8-bit Honeywell XTS-200 proc.
    - Ford Aerospace: Multinet Gateway System

Four Old Theorem Proving Systems (1984)

- FDM (SDC)
  - InaJo: proof by contradiction (tableaux method)
  - nondeterministic state machines
- HDM (SRI)
  - Special: lambda-calculus, second-order pred. calculus, Hoare sentences, Pascal code
  - TP: decision procs. for 1st-order logic and subset of int. arith.
- Gypsy (Texas)
  - Pascal-like program verification system
  - vc generator
- Affirm (USC/ISI)
  - algebraic specifications
  - rewrite rule engine (Knuth-Bendix)

Modern Tools

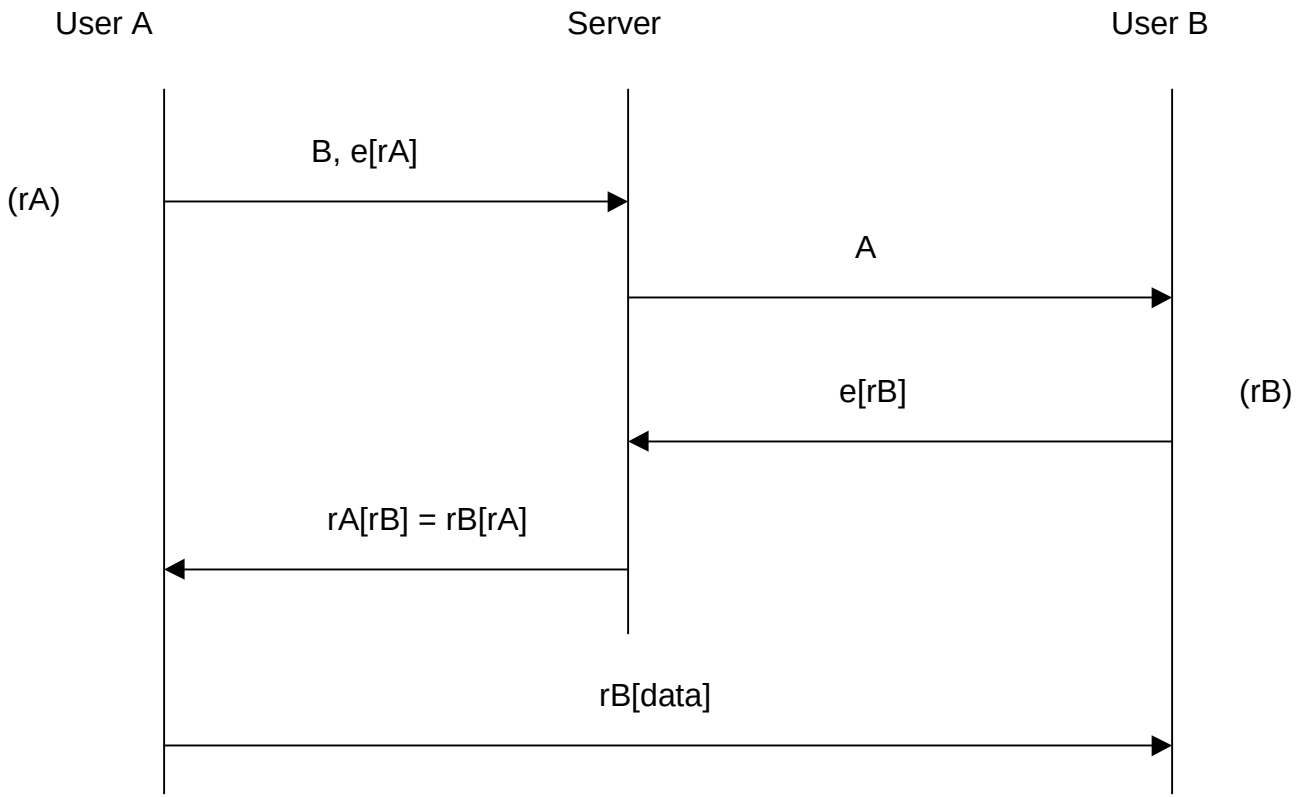
- Security-specific tools
  - NRL Protocol Analyzer
- Model Checking Approaches
  - FDR
  - Murϕ
  - Brutus (Marrero)
- Theorem Proving Approaches
  - Isabelle
  - PVS
- Hybrid Approaches
  - Revere (Kindred)

NRL Protocol Analyzer [Syverson and Meadows 93]

- Background
  - Based on Dolev-Yao term rewriting model.
    - Assumes intruder who can read all message traffic, modify and destroy any message traffic, and perform any operation (e.g., encryption and decryption) available to a legitimate user of the protocol.
    - Assumes there is a set of "words" (e.g., encryption keys possessed by honest principals) that the intruder does not know.
    - Intruder's goal: cause principals to reach a state that is incompatible with protocol's correct operation.
- General Idea
  - Start with a given "bad" state.
    - Use temporal logic to characterize "bad" state.
  - Prove state is not reachable or generate a penetration scenario.
    - Show that intruder can manipulate algebraic system of words in a way that the "bad" state is reached.

TMN Protocol Example

A -> S: $r_A^3 \mod n$
B -> S: $r_B^3 \mod n$
S -> A: r_A *xor* r_B



Specification Input: pre- and post-conditions, event

Honest user A attempts to establish communication with another user, B (honest or not). A sends message containing an encrypted random number and remembers both the number and the name of the intended message recipient.

```
rule(1)
If:
count(user(A,honest)) = [N],
intruderknows([user(A,Y), user(B,X), R]),
then:
count(user(A,honest)) = [s(N)],
intruderlearns([pke(pubkey(server),rand(user(A,honest),N)),
                    user(A,honest),user(B,W)]),
lfact(user(A,honest),N,step1,s(N)) = [user(B,W),rand(user(A,honest),N)],
EVENT:
event(user(A,honest),N,initrequest,s(N)) =
  [user(B,W),rand(user(A,honest),N)].
```

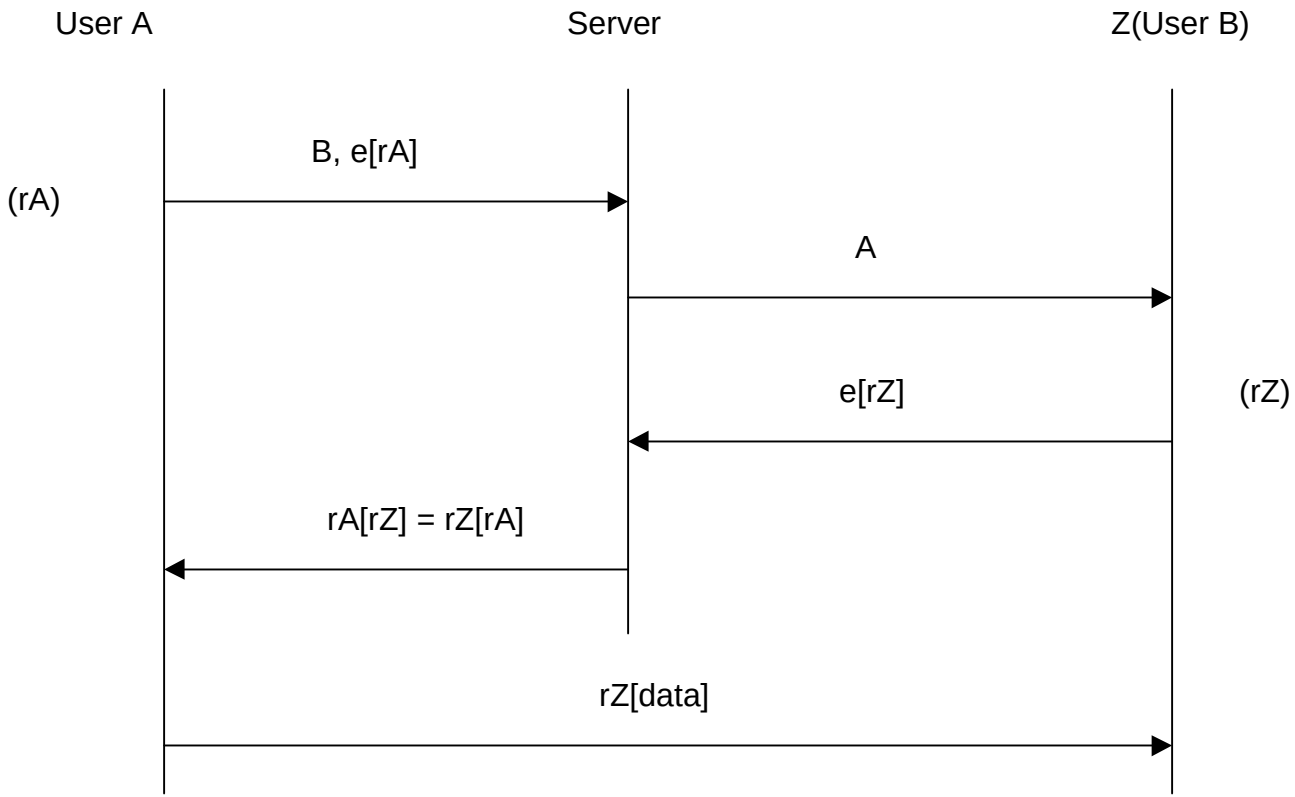
Run NRL Protocol Analyzer on TMN

- In general: Determine how it is possible for system to reach a state in which user(A, honest) believes that some word K is a key and K is known to the intruder.

- Analyzer discovers this scenario:
  - Intruder can impersonate user B to server, convincing A that a key generated by the intruder is a key generated by B.
    - Problem arises because no authentication is used between parties.

TMN Protocol Example

A -> S: $r_A^3 \bmod n$
Z(B) -> S: $r_Z^3 \bmod n$
S -> A: r_A *xor* r_Z
A-> Z(B): ...



NRL Protocol Analyzer

- General features
  - Can handle multiple runs
  - Semi-automatic (has ways to reduce search space so that exhaustive search is possible)
- Found bugs or demonstrated (known) flaws in
  - *Simmons' Selective Broadcast Protocol
  - *Burns and Mitchell's Resource Sharing Protocol
  - Needham-Schroeder Public-Key
  - TMN
  - Encapsulating Security Protocol
  - Draft ISO authentication protocol
- Status
  - Used currently to specify and verify Internet Key Exchange Protocol (IKE), a combination of Internet Security Association and Key Management Protocol (ISAKMP) and Oakley protocols
  - Has improved theorem proving capability, including backward search, forward lemma generator, etc.

Model Checking Approaches

- FDR (trace-based)
- Murϕ, SMV (state-machine based)
- Brutus (application domain specific)

FDR [Roscoe 94]

- Based on Hoare's CSP. Intuitively, (but not technically correct):
  - A process, P, is a set of traces, which are sequences of events.
  - A specification, S, is a predicate defining "allowable" set of traces.
  - Verification requires showing P is contained in S. More technically, does P *refine* S?
- FDR is a model checker for the Failures-Divergences Refinement semantic interpretation of CSP (due to Roscoe and Brookes.)
  - Finite number of processes, events, …

Needham-Schroeder Public-Key Example

Reduced protocol:

1. A -> B: A, B, $\{N_a, A\}_{K_b}$
2. B -> A: B, A, $\{N_a, N_b\}_{K_a}$
3. A -> B: A, B, $\{N_b\}_{K_b}$

- Encode in CSP/FDR syntax:, e.g.,
  1. Comm.Msg1.A.B.Encrypt.Kb.Na.A
- Define set of messages specific to protocol.
- Model Intruder Z as separate process.
- Define system as a set of concurrent processes, A || B || Z
- Define “correct” system to be the set of traces where both sender and receiver are authenticated to each other.

Flaw Found Using FDR

- FDR shows receiver authentication holds, but not sender authentication: B commits to a session with A even though A is not trying to establish a session with B.
- The attack (interleaving of two runs):

  a.1. A -> Z: $A, Z, \{N_a, A\}_{K_z}$

  b.1. Z(A) -> B: $A, B, \{N_a, A\}_{K_b}$

  b.2. B -> Z(A): $B, A, \{N_a, N_b\}_{K_a}$

  a.2. Z -> A: $Z, A, \{N_a, N_b\}_{K_a}$

  a.3. A -> Z: $A, Z, \{N_b\}_{K_z}$

  b.3. Z(A) -> B: $A, B, \{N_b\}_{K_b}$
- The fix: Add sender name to Message 2.
- FDR shows sender authentication holds in fixed protocol.
- Times: “less than two minutes”

Murϕ [Dill et al. 92]

- Nondeterministic state-machine based model checker
  - Used primarily for hardware verification
  - In same family of model checkers as SMV
- State = set of global variables
- State transition is performed by *rules*, each specified by a boolean (enabling) condition and action, which may change values of globals.
- Parallel composition is union of rules.
- Explicitly models Intruder, Z, per protocol.

Examples Done by Mitchell, Mitchell, Stern [97]

- Needham Schroeder Public-Key

ini	res	int	states	time
1	1	1	1706	3.1s
2	2	1	514550	5761.1s

- TMN
  - Rediscovered all known bugs, e.g., Z impersonating B, all in “reasonably short” times (with some careful modeling).
- Kerberos
  - Found bug with |Servers| = 2, exploring 3405 states in 18.5 seconds
    - Name of server that client wants to get a ticket is transmitted unencrypted. Possible replay attack by eavesdropping intruder.
- Current efforts at Stanford to specify and verify SSL 3.0
  - Handshake protocol used for establishing a secure communications channel between a client and server.
  - Found bugs in SSL 2.0.

Brutus [Clarke, Jha, Marrerro 96]

- Finite state-transition model of principals and intruder.
- Checks secrecy and correspondence properties
  - Secrecy: set of terms the intruder is not allowed to obtain (similar to NRL Protocol Analyzer)
  - Correspondence (due to Woo-Lam): If a principal A believes it has finished a protocol run with principal B, then principal B must have begun a protocol run with principal A.
    - More generally, the projection of any trace onto X events and Y events must be derivable from this grammar:

$$S \rightarrow SxSy \mid \varepsilon$$

- Checks all possible traces, each of which is an alternating sequence of global states and actions. Performs depth first search of state space to generate all possible traces.

Brutus cont'd

- Built-in model of intruder (so you don't have to build one from scratch per protocol).

- E.g., the closure B* of a set of messages B:
  - If m in B, then m in B*
  - If m1 in B and m2 in B, then m1 $_o$ m2 in B*. (Pairing)
  - If m1 $_o$ m2 in B then m1 in B* and m2 in B*. (Projection)
  - If m in B* and key k in B* then $\{m\}_k$ in B*. (Encryption)
  - If $\{m\}_k$ in B* and key k^{-1} in B*. (Decryption)

Woo-Lam Example

- The protocol:
  1. A -> B: A
  2. B -> A: N_b
  3. A -> B: $\{N_b\}_{Kas}$
  4. B -> S: $\{A, \{N_b\}_{Kas}\}_{Kbs}$
  5. S -> B: $\{A, N_b\}_{Kbs}$

- Flaw found
  1. Z(A) -> B: A
  2. B -> Z(A): N_b
  3. Z(A) -> B: N_b
  4. B -> Z(S): $\{A, N_b\}_{Kbs}$
  5. Z(S) -> B: $\{A, N_b\}_{Kbs}$

Brutus Current Status

- Applying to iKP (family of e-commerce protocols)
  - Found another type error, in 1KP: bit string for credit card number
  - Verified properties like
    - Proof of transaction authorization by customer.
    - Unauthorized payment is impossible.
    - Order information and credit card numbers are not revealed.
    - Merchant does not gain identity of customer (anonymity).
- Will's Ph.D. thesis, to be defended soon (this term), so stay tuned for more.

Theorem Proving Approaches

- Isabelle
- PVS, Coq

Isabelle [Paulson 94]

- General-purpose theorem prover for higher-order logics (HOL).
  - Not restricted to belief logics.
- Supports induction
  - Not restricted to finite models.
- Inductively defined operators, used to define set H (all traffic enemy observes):
  - *parts*: enumerate all components of a set of messages
  - *analyz*: models decryption of past traffic using available keys
  - *synth*: models generation of spoof messages
- Model a protocol
  - Each event in a trace:

    Says A B X
  - Traces defined inductively, according to protocol steps.

Proving Protocol Properties

- Possibility properties
  - E.g., for all agents A and B, distinct from themselves and distinct from the server, there is a key K, nonce N, and a trace such that the final message B -> A: N_a, $\{N_a, K_{ab}\}_{Ka}$ is sent.
- Unicity theorems
  - Fresh session keys and nonces uniquely identify their message of origin.
- Secrecy theorems
  - If K can be analyzed from a set of session keys and messages, then either it is one of those keys, or it can be analyzed from the messages alone.
  - Proof is non-trivial. Use tactics, induction, forwarding lemmas, rewrite rules, and case splits.
  - "**Proving … took weeks**, partly spent building libraries of tactics and relevant laws. A constant problem…is…**gigantic formulas**."

(Variant of) Otway-Rees Example

1. A -> B: N_a, A, B, $\{N_a, A, B\}_{K_a}$
2. B -> S: N_a, A, B, $\{N_a, A, B\}_{K_a}$, N_b, $\{N_a, A, B\}_{K_b}$
3. S -> B: N_a, $\{N_a, K_{ab}\}_{K_a}$, $\{N_b, K_{ab}\}_{K_b}$
4. B -> A: N_a, $\{N_a, K_{ab}\}_{K_a}$

General approach: Given a trace *evs,* consider all possible extensions of evs. For example, if evs is of the form:

Says A B {Na, A, B, X}

and N_b is a fresh nonce and B =/= S then *evs* may be extended with the event:

Says B S {Na, A, B, X, Nb, {Na, A, B}Kb}

An Attack

1. A -> B: N_a, A, B, $\{N_a, A, B\}_{K_a}$ (intercepted)
1'. C -> A: N_c, C, A, $\{N_c, C, A\}_{K_c}$
2'. A -> S: N_c, C, A, $\{N_c, C, A\}_{K_c}$, N_a', $\{N_c, C, A\}_{K_a}$ (intercepted)
2''. C(A) -> S: N_c, C, A, $\{N_c, C, A\}_{K_c}$, N_a, $\{N_c, C, A\}_{K_a}$
3'. S -> A: N_c, $\{N_c, K_{ca}\}_{K_c}$, $\{N_a, K_{ca}\}_{K_a}$ (intercepted)
4. C(B) -> A: N_a, $\{N_a, K_{ca}\}_{K_a}$

- N_a is used in two roles.
- This attack is more serious than the one discovered by Mao and Boyd where server could detect repetition of nonce.
- This attack cannot occur in original Otway-Rees since N_b is encrypted in second message.

Status of Work

- Other examples
  - Three versions of Otway-Rees
  - Two variants of Yahalom
  - Simplified version of Woo-Lam
  - Needham-Schroeder (symmetric- and public-key)
  - Kerberos
- “Proofs are highly automated”
  - One command can generate tens or hundreds of inferences.
  - Small changes to protocols involve only small changes to proof scripts.

Other Theorem Proving Systems

- PVS
  - B. Dutertre and S. Schneider. "Using a PVS Embedding of CSP to Verify Authentication Protocols", Theorem Proving in Higher Order Logics, TPHOL's 97 , LNCS 1275, pp. 121-136, August 1997.
- Coq
  - D. Bolignano. “An Approach to the Formal Verification of Cryptographic Protocols Using Coq,” Technical Report, INRIA-Rocquencourt, 1996.

Hybrid Approach: Kindred's Theory Generation

- General approach (fast and completely automatic):
  - Take logic like BAN.
  - Encode protocol in BAN.
  - Apply rules till you don't get any new formulae.
  - Result is finite representation of theory of the protocol.
- Verifying a property holds is a simple membership check.
- Other neat aspects:
  - Theory "diff" encourages protocol design ("what if…" scenarios).
  - Applied to classic set of authentication protocols.
  - First to attack BAN interpretation problem formally.
- Darrell's Ph.D. thesis, to be defended soon (this term), so stay tuned for more.

Limitations of Tools and Techniques

- Model checkers
  - Finite state space exploration
    - security domain is different from hardware
- Proof checkers, theorem provers
  - General-purpose ones
    - usual complaints
  - Special-purpose ones
    - logics not appropriate or fully developed
- Specifying properties
    - temporal properties are not all that's needed

Main Criticism

- Formal analysis tools really haven't discovered any major flaws in authentication protocols or systems, yet we know these protocols, standards, and systems based on them are insecure.
- What's the problem?
  - No one has come up with the right tool/technique/logic yet for this problem domain.
  - People have been focusing on the wrong aspects of security protocols/properties.
    - Belaboring the obvious, ignoring the difficult?
  - "Security" is a negative property. (Can't prove the absence of holes.)

Readings, Next Lecture

Topic: Intro to Electronic Commerce Protocols

Readings:

- This Lecture
  - Required: Papers by Kemmerer, Meadows, and Millen; Lowe, Mitchell, Mitchell, and Stern; Paulson.
  - Edification: Papers by Clarke, Jha, and Marrero; Kindred and Wing
- Next Lecture
  - E-commerce papers