Network Security

Logical Systems for Security Protocol Analysis

Formal Methods

- Why spend time on Formal Methods in a Network Security course?
  - We use Formal Methods to analyze properties about network security components.

Logical System

- A Logical System Consists of:
  - A Language that defines the set of well formed formulas (WFFs).
  - A set of axioms
    - Each of which is a WFF
    - They are assumed/accepted to be true
    - If they are not true, the system is not valid
  - A set of rules of inference that allow transformation from one or more WFFs to a new WFF.

A Logical Thought

- To say of what is that it is not, or of what is not that is, is false, while to say of what is that it is, and of what is not that it is not, is true.
  - Aristotle

- What's true is true, and what's false is false.
  - Snoopy

Formal Methods

Methods that:

1. Translate a syntactic structure into a semantic representation
2. Systematically derive a cognitive representation from a description based exclusively on the form of the description
3. Produce proofs from specifications

First-order Logic

- Model-theoretic
  - Represent some aspect of the real world
  - Used to describe/assist with understanding

- Proof-theoretic
## Logic Review
- **Deduction**: Reasoning from the general to the general or from general to specific
  - Considered analytical
  - A priori because it starts from general principles which are regarded as logically prior to the facts which follow from them
- **Induction**: Reasoning from the specific to the general
  - Concerned with empirical statements, considered synthetic (putting together)

## Knowledge vs Belief
- **Axiom T**
  - If a subject knows P, then P
  - Knowledge systems, not belief
  - Knowledge implies truth, not so belief
- **Doxastic Logic**
  - Logic of belief
  - Most useful in examining trust
- **Epistemic Logic**
  - Logic of knowledge and belief
  - Used in examining security

## Languages
- **Types**
  - Natural
  - Programming
  - Specification
  - Machine
  - Etc.
- **Makeup (formal or informal)**
  - Syntax
  - Semantics

## Abstraction and Perspectives
- **Abstraction**
  - Assembly language "abstracts" machine language
  - C abstracts assembly (or machine) language
  - SQL further abstracts these less abstract languages
- **Perspective**
  - Higher level languages allow the programmer to view the problem from a broader perspective
  - Usually, that means to think of the program in terms of the problem, rather that in terms of the ability of the computer to do what they want.

## Formality
- **Pros**
  - Language can be compiled
  - Language (syntax) can be generated
  - Formal semantics can be automated
- **Cons**
  - Lose expressiveness
  - Formal language cannot map exactly to natural language, so formal language may be hard to learn and miscommunication can occur.
Propositional Logic Rules of Inference

- Conjunction: Given p and given q, infer p & q
- Simplification: Given (p & q) infer p, q
- Identity: p -> p
- Addition: given p, infer p or q
- Transposition: (p -> q) -> (~p -> ~q)
- Modus Ponens (if p -> q, then given p, infer q)
- Modus Tollens (if p -> q, then given ~q, infer ~p)
- Hypothetical syllogism: (p->q) & (q->r) -> (p->r)
- Disjunctive syllogism: ((p or q) & ~p) -> q
- Constructive dilemma: 
  – ((p->q) & (r->s) & (p or r)) -> (q or s)

MODAL LOGICS

- Truth values of propositions can be:
  – Possibly true
  – Possibly false
  – Necessarily true
  – Necessarily false
  – Contingent

LOGIC OF BELIEF

- An agent's state of belief corresponds to the extent to which:
  – Based on its local state
  – the agent can determine what global state it is in
- Agent A_i considers possible when the system is in state s
- An agent does not believe p iff there is at least one global state it considers possible where p does not hold

Logical System

1. Specification Language
2. Rules of Inference

Formal Approach

- Specify the protocol in the logical language
- Apply the axioms and rules of inference to try and prove things

Sound and Complete Logics

- A logical system is COMPLETE if all theorems are derivable from the axioms (if you can prove everything that is true)
- A logical system is SOUND if all theorems that are derivable from the axioms are true (i.e. if you can't prove false things)
Epistemic Concepts In Logical Systems for Protocol Analysis

- A sees X
  - A recognizes X
  - A possesses X
- A said X
- A did not say X
- A believes X
- A controls X
  - A has jurisdiction over X

- fresh X
- K is a good key between A and B
- PK is A’s public key
- X is a secret shared between A and B
- X is signed by A

Messages

- Plaintext
- Numeric
- Computed
- Encrypted
- Signed

Symbols and Predicates in BAN Logic

- A sees X
  - A < X
  - sees(A, X)
- A said X
  - A | X
  - said(A, X)
- A believes X
  - A |= X
  - believes(A, X)

- A controls X
  - A => X
  - controls(A, X)
- fresh X
  - #(X)
  - fresh(X)
- K is a good key
  - A K B
  - goodkey(k, A, B)
- X encrypted under K
  - {X}K

Inference Rules

- Message meaning
  - If:
    1. k is a good key between me and Bob and
    2. I see message X encrypted under k then
    3. I can believe that X came from Bob, i.e. said(Bob, X)

    - goodkey(k, A, B) A sees(A, {X}k) then
    - believes(A, said(B, X))

Symbols in BAN Logic

- A controls X
  - A => X
  - controls(A, X)
- fresh X
  - #(X)
  - fresh(X)
- K is a good key
  - A K B
  - goodkey(k, A, B)
- X encrypted under K
  - {X}K

Inference Rules

- Nonce verification
  - If:
    1. I believe that Bob said X then
    2. I believe that X is fresh and
    3. I can believe that Bob believes X

    - believes(A, fresh(X)) A believes(A, said(B, X)) then
    - believes(A, believes(B, X))
Inference Rules

- Jurisdiction
  - If:
    1. I believe that Sally believes X
    2. I believe that Sally controls X
    3. I can believe that X
  - \( \text{believes}(A, \text{believes}(S, X)) \land \text{believes}(A, \text{controls}(S, X)) \) then \( \text{believes}(A, X) \)

Simplification Axioms

- Concatenated messages
  - If believes \((P, (X, Y)), \) then believes \((P, X)\)
  - If believes \((P, (\text{believes}(Q, (X, Y))), \) then believes \((P, (\text{believes}(Q, X)))\)
  - If believes \((P, (\text{sees}(Q, (X, Y))), \) then believes \((P, (\text{sees}(Q, X)))\)
- Key ordering
  - If believes \((P, (\text{goodkey}(k, X, Y))), \) then believes \((P, (\text{goodkey}(k, X, Y)))\)

Misc Axioms

- Sees (catenation and encryption)
  - If sees \((P, (X, Y)), \) then sees \((P, X)\)
  - If believes \((P, (\text{goodkey}(k, P, \text{anybody}))), \) and sees \((P, (\text{anybody}(X))), \) then sees \((P, (X))\)
- Freshness
  - If believes \((P, (\text{fresh}(X))), \) then believes \((P, (\text{fresh}(X, Y)))\)

Idealizing Protocols

- Sending a goodkey
  - A \( \rightarrow \) B: \{B, kab\}kbs
    - may be idealized as
  - A \( \rightarrow \) B: goodkey(kab, A, B)
  - How did we know this?
    - We know the intention of the protocol
    - We know what S does
  - Idealization is the translation of the action on data into the beliefs and intentions of the participants when the messages are sent.

Idealization Quote From BAN

- Only knowledge of the entire protocol can determine the essential contents of the message. There are guidelines to control what transformations are possible …. Roughly, a real message \( m \) can be interpreted as a formula \( X \) if whenever the recipient gets \( m \) he may deduce that the sender must have believed \( X \) when he sent \( m \).
**Annotating Protocols**

- Once a protocol is idealized, it can be reviewed for "low hanging fruit" knowledge, also known as obvious facts.
- `sees` is a commonly annotated propositions
- For example, if we have an idealized step:
  - A -> B: goodkey(kab,A,B)
  - We can add the annotation that:
    - sees(B, {goodkey(kab,A,B)})

**The BAN Logic Process**

1. Idealize the protocol
2. Annotate the idealized protocol
3. Apply the rules of inference to prove desired properties
   - Desired properties are usually expressed in terms of two-way belief of valid key exchange:
     - `believes(A,goodkey(kab,A,B))` ∧
     - `believes(B,goodkey(kab,A,B))`

**BAN is not Complete**

- A logical system is COMPLETE if all theorems are derivable from the axioms (i.e. if you can prove everything that is true)
- BAN is not complete
- A failed proof does not mean that a protocol is secure (or not secure)
- A failed proof often points to a problem in the protocol

**A Problem With Cleartext**

- Clear text messages cannot contribute to security
- Clear text messages can compromise security
- Consider the following key distribution protocol:
  - A -> S: "I need a new key with Bob"
  - S -> A: kab'
  - A -> B: {kab',na}kab
  - B -> A: {na,nb}kab'
  - A -> B: nb
- There are no mechanisms in BAN Logic to detect the invalidity of this protocol
Trust verses Security

- Security
  - Penetrator's viewpoint
  - Knowledge representation
  - Security (can anyone steal the information?)

Knowledge: P can only KNOW X if X is true. Thus: knows(P,X) infers X

- Trust
  - Legitimate subject's viewpoint
  - Belief representation
  - Functionality (does the protocol deliver the information?)

Belief: believes(P,X) can be true whether X is true or false

Other Logics Developed for Security Protocol Analysis

- GNY: Gong, Needham, and Yaholm
- Moser
- BGNY: Steve Bracken
- SVO: Syverson and van Oorschot
- KPL: Syverson
- CKT5: Bieber

GNY

told(P,X)
possesses(P,X)
once_said(P,X)
believes(P,fresh(X))
believes(P,recognizable(P))
believes(P,suitableSecret(S,P,Q))
believes(P,goodkey(K,Q))
not_originated_here(X)

Review

- Logical Systems
- BAN Logical Language
- BAN Logic Rules of Inference
- Idealization