TECS Week 2005

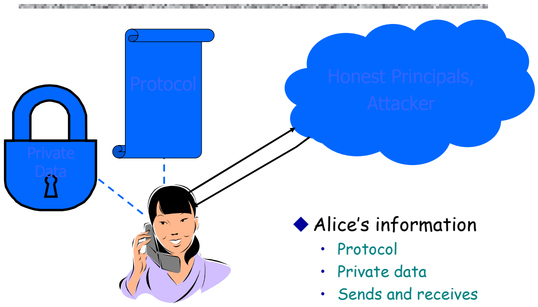
Protocol Composition Logic

John Mitchell
Stanford

Intuition

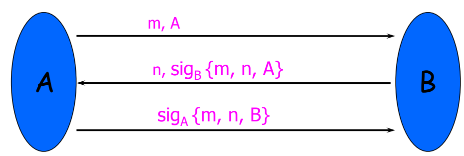
- Reason about local information
  - I chose a new number
  - I sent it out encrypted
  - I received it decrypted
  - Therefore: someone decrypted it
- Incorporate knowledge about protocol
  - Protocol: Server only answers if sent a request
  - If *server not corrupt* and
    - I receive an answer from the server, then
    - the server must have received a request

Intuition: Picture



- Alice's information
  - Protocol
  - Private data
  - Sends and receives

Example: Challenge-Response



- Alice reasons: if Bob is honest, then:
  - only Bob can generate his signature. [protocol independent]
  - if Bob generates a signature of the form $sig_B\{m, n, A\}$,
    - he sends it as part of msg2 of the protocol and
    - he must have received msg1 from Alice [protocol dependent]
  - Alice deduces: Received (B, msg1) Λ Sent (B, msg2)

Formalizing the Approach

- Language for protocol description
  - Write program for each role of protocol
- Protocol logic
  - State security properties
  - Specialized form of temporal logic
- Proof system
  - Formally prove security properties
  - Supports modular proofs

Cords

- Protocol programming language
  - Server = [receive x; new n; send {x, n}]
- Building blocks
  - Terms
    - names, nonces, keys, encryption, …
  - Actions
    - send, receive, pattern match, …

Terms

t ::=	c	constant term
	x	variable
	N	name
	K	key
	t, t	tupling
	$sig_K\{t\}$	signature
	$enc_K\{t\}$	encryption

Example: x, $sig_B\{m, x, A\}$ is a term

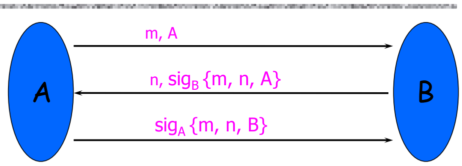
Actions and Cords

- Actions
  - send t; — send a term t
  - receive x; — receive a term into variable x
  - match t/p(x); — match term t against p(x)
- Cord
  - Sequence of actions
- Notation
  - Some match actions are omitted in slides

    receive sigB{A, n} means

    receive x; match x/sigB{A, n}

Challenge-Response as Cords

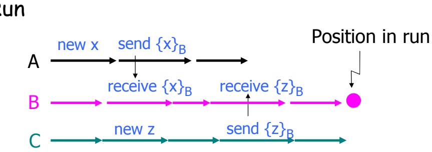



```
InitCR(A, X) = [
    new m;
    send A, X, {m, A};
    receive X, A, {x, sigX{m, x, A}};
    send A, X, sigA{m, x, X}};
]
```

```
RespCR(B) = [
    receive Y, B, {y, Y};
    new n;
    send B, Y, {n, sigB{y, n, Y}};
    receive Y, B, sigY{y, n, B}};
]
```

Execution Model

- Protocol
  - Cord gives program for each protocol role
- Initial configuration
  - Set of principals and keys
  - Assignment of ≥1 role to each principal
- Run



Formulas true at a position in run

- Action formulas

  a ::= Send(P,m) | Receive (P,m) | New(P,t)
  | Decrypt (P,t) | Verify (P,t)
- Formulas

  φ ::= a | Has(P,t) | Fresh(P,t) | Honest(N)
  | Contains(t_1, t_2) | $\neg\varphi$ | $\varphi_1 \wedge \varphi_2$ | $\exists x\, \varphi$
  | $\bigcirc\varphi$ | $\Diamond\varphi$
- Example

  After(a,b) = $\Diamond(b \wedge \bigcirc\Diamond a)$

Modal Formulas

- After actions, postcondition

  $[\text{ actions }]_P\ \varphi$ where P = ⟨princ, role id⟩
- Before/after assertions

  $\varphi\ [\text{ actions }]_P\ \psi$
- Composition rule

$$\frac{\varphi\,[\,S\,]_P\,\psi \qquad \psi\ [\,T\,]_P\,\theta}{\varphi\,[\,ST\,]_P\,\theta}$$

Note: same P in all formulas

Security Properties

- **Authentication for Initiator**

  CR |= [InitCR(A, B)]$_A$ Honest(B) ⊃
  ActionsInOrder(
  Send(A, {A,B,m}),
  Receive(B, {A,B,m}),
  Send(B, {B,A,{n, sig$_B$ {m, n, A}}}),
  Receive(A, {B,A,{n, sig$_B$ {m, n, A}}})
)
- **Shared secret**

  NS |= [InitNS(A, B)]$_A$ Honest(B) ⊃
  (Has(X, m) ⊃ X=A ∧ X=B)

Semantics

- Protocol Q
  - Defines set of roles (e.g, initiator, responder)
  - Run R of Q is sequence of actions by principals following roles, plus attacker
- Satisfaction
  - Q, R |= [*actions*]$_P$ ϕ

    Some role of P in R does exactly *actions* and ϕ is true in state after *actions* completed
  - Q |= [*actions*]$_P$ ϕ

    Q, R |= [*actions*]$_P$ ϕ for all runs R of Q

Proof System

- Goal: prove properties formally
- Axioms
  - Simple formulas provable by hand
- Inference rules
  - Proof steps
- Theorem
  - Formula obtained from axioms by application of inference rules

Sample axioms about actions

- New data
  - [new x]$_P$ Has(P,x)
  - [new x]$_P$ Has(Y,x) ⊃ Y=P
- Actions
  - [send m]$_P$ ◇Send(P,m)
- Knowledge
  - [receive m]$_P$ Has(P,m)
- Verify
  - [match x/sig$_X${m}]$_P$ ◇ Verify(P,m)

Reasoning about knowledge

- Pairing
  - Has(X, {m,n}) ⊃ Has(X, m) ∧ Has(X, n)
- Encryption
  - Has(X, enc$_K$(m)) ∧ Has(X, K^{-1}) ⊃ Has(X, m)

Encryption and signature

- Public key encryption

  Honest(X) ∧ ◇Decrypt(Y, enc$_X${m}) ⊃ X=Y
- Signature

  Honest(X) ∧ ◇Verify(Y, sig$_X${m}) ⊃
  ∃ m' (◇Send(X, m') ∧ Contains(m', sig$_X${m})

Sample inference rules

◆ Preservation rules

$$\frac{\psi\ [\text{ actions }]_P\ Has(X, t)}{\psi\ [\text{ actions; action }]_P\ Has(X, t)}$$

◆ Generic rules

$$\frac{\psi\ [\text{ actions }]_P\ \phi \qquad \psi\ [\text{ actions }]_P\ \varphi}{\psi\ [\text{ actions }]_P\ \phi \wedge \varphi}$$

Bidding conventions (motivation)

◆ Blackwood response to 4NT
- 5♣ : 0 or 4 aces
- 5♦ : 1 ace
- 5♥ : 2 aces
- 5♠ : 3 aces

◆ Reasoning
- If my partner is following Blackwood, then if she bid 5♥, she must have 2 aces

Honesty rule (rule scheme)

$\forall$roles R of Q. $\forall$ initial segments A $\subseteq$ R.

$$\frac{Q\ |\text{-}\ \ [\ A\ ]_X\ \phi}{Q\ |\text{-}\ Honest(X) \supset\ \phi}$$

- This is a finitary rule:
  - Typical protocol has 2-3 roles
  - Typical role has 1-3 receives
  - Only need to consider A waiting to receive

Honesty rule (example use)

$\forall$roles R of Q. $\forall$ initial segments A $\subseteq$ R.

$$\frac{Q\ |\text{-}\ \ [\ A\ ]_X\ \phi}{Q\ |\text{-}\ Honest(X) \supset\ \phi}$$

- Example use:
  - If Y receives a message from X, and Honest(X) $\supset$ (Sent(X,m) $\supset$ Received(X,m')) then Y can conclude Honest(X) $\supset$ Received(X,m'))

Correctness of CR

InitCR(A, X) = [
new m;
send A, X, {m, A};
receive X, A, {x, sig_X{m, x, A}};
send A, X, sig_A{m, x, X}};
]

RespCR(B) = [
receive Y, B, {y, Y};
new n;
send B, Y, {n, sig_B{y, n, Y}};
receive Y, B, sig_Y{y, n, B}};
]

CR |- [InitCR(A, B) $]_A$ Honest(B) $\supset$
ActionsInOrder(
Send(A, {A,B,m}),
Receive(B, {A,B,m}),
Send(B, {B,A,{n, sig_B {m, n, A}}}),
Receive(A, {B,A,{n, sig_B {m, n, A}}})
)

Correctness of CR – step 1

InitCR(A, X) = [
new m;
send A, X, {m, A};
receive X, A, {x, sig_X{m, x, A}};
send A, X, sig_A{m, x, X}};
]

RespCR(B) = [
receive Y, B, {y, Y};
new n;
send B, Y, {n, sig_B{y, n, Y}};
receive Y, B, sig_Y{y, n, B}};
]

1. A reasons about it's own actions

CR |- [InitCR(A, B) $]_A$
$\Diamond$ Verify(A, sig_B {m, n, A})

Correctness of CR – step 2

InitCR(A, X) = [
 new m;
 send A, X, {m, A};
 receive X, A, {x, sig_X{m, x, A}};
 send A, X, sig_A{m, x, X}};
]

RespCR(B) = [
 receive Y, B, {y, Y};
 new n;
 send B, Y, {n, sig_B{y, n, Y}};
 receive Y, B, sig_Y{y, n, B}};
]

2. Properties of signatures

CR |- [InitCR(A, B)]$_A$ Honest(B) ⊃
∃ m′ (◇Send(B, m′) ∧ Contains(m′, sig_B {m, n, A})

Correctness of CR – Honesty

InitCR(A, X) = [
 new m;
 send A, X, {m, A};
 receive X, A, {x, sig_X{m, x, A}};
 send A, X, sig_A{m, x, X}};
]

RespCR(B) = [
 receive Y, B, {y, Y};
 new n;
 send B, Y, {n, sig_B{y, n, Y}};
 receive Y, B, sig_Y{y, n, B}};
]

Honesty invariant

CR |- Honest(X) ∧
◇Send(X, m′) ∧ Contains(m′, sig_x {y, x, Y}) ∧ ¬ ◇New(X, y) ⊃
m= X, Y, {x, sig_B{y, x, Y}} ∧ ◇Receive(X, {Y, X, {y, Y}})

Correctness of CR – step 3

InitCR(A, X) = [
 new m;
 send A, X, {m, A};
 receive X, A, {x, sig_X{m, x, A}};
 send A, X, sig_A{m, x, X}};
]

RespCR(B) = [
 receive Y, B, {y, Y};
 new n;
 send B, Y, {n, sig_B{y, n, Y}};
 receive Y, B, sig_Y{y, n, B}};
]

3. Use Honesty rule

CR |- [InitCR(A, B)]$_A$ Honest(B) ⊃
 ◇ Receive(B, {A,B,m}),

Correctness of CR – step 4

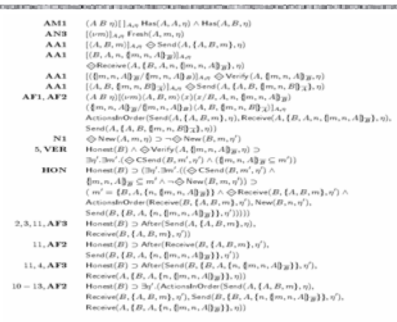
InitCR(A, X) = [
 new m;
 send A, X, {m, A};
 receive X, A, {x, sig_X{m, x, A}};
 send A, X, sig_A{m, x, X}};
]

RespCR(B) = [
 receive Y, B, {y, Y};
 new n;
 send B, Y, {n, sig_B{y, n, Y}};
 receive Y, B, sig_Y{y, n, B}};
]

4. Use properties of nonces for temporal ordering

CR |- [InitCR(A, B)]$_A$ Honest(B) ⊃ Auth

Complete proof



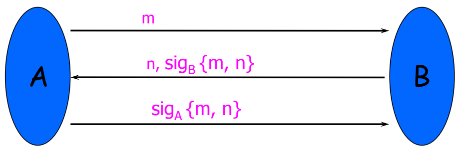
AM1	$(A\ B\ \eta)[\ ]_{A,\eta}$ Has$(A, A, \eta) \wedge$ Has(A, B, η)
AN3	$[(\nu m)]_{A,\eta}$ Fresh(A, m, η)
AA1	$[\langle A, B, m\rangle]_{A,\eta}$ ◇Send$(A, \{A, B, m\}, \eta)$
AA1	$[(B, A, n, \{\!|m, n, A|\!\}_{\bar{B}})]_{A,\eta}$ ◇Receive$(A, \{B, A, n, \{\!|m, n, A|\!\}_{\bar{B}}\}, \eta)$
AA1	$[(\{\!|m, n, A|\!\}_{\bar{B}}/\{\!|m, n, A|\!\}_{B})]_{A,\eta}$ ◇Verify$(A, \{\!|m, n, A|\!\}_{\bar{B}}, \eta)$
AA1	$[\langle A, B, \{\!|m, n, B|\!\}_{\bar{X}}\rangle]_{A,\eta}$ ◇Send$(A, \{A, B, \{\!|m, n, B|\!\}_{\bar{X}}\}, \eta)$
AF1, AF2	$(A\ B\ \eta)[(\nu m)\langle A, B, m\rangle(x)(x/B, A, n, \{\!|m, n, A|\!\}_{\bar{B}})$ $(\{\!|m, n, A|\!\}_{\bar{B}}/\{\!|m, n, A|\!\}_{B})\langle A, B, \{\!|m, n, B|\!\}_{\bar{X}}\rangle]_{A,\eta}$ ActionsInOrder(Send$(A, \{A, B, m\}, \eta)$, Receive$(A, \{B, A, n, \{\!|m, n, A|\!\}_{\bar{B}}\}, \eta)$, Send$(A, \{A, B, \{\!|m, n, B|\!\}_{\bar{X}}\}, \eta)$)
N1	◇New$(A, m, \eta) \supset \neg$◇New$(B, m, \eta')$
5, VER	Honest$(B) \wedge$ ◇Verify$(A, \{\!|m, n, A|\!\}_{\bar{B}}, \eta) \supset$ $\exists\eta'.\exists m'.$(◇CSend$(B, m', \eta') \wedge (\{\!|m, n, A|\!\}_{\bar{B}} \subseteq m')$)
HON	Honest$(B) \supset (\exists\eta'.\exists m'.(($◇CSend$(B, m', \eta') \wedge$ $\{\!|m, n, A|\!\}_{\bar{B}} \subseteq m' \wedge \neg$◇New$(B, m, \eta')) \supset$ $(m' = \{B, A, \{n, \{\!|m, n, A|\!\}_{\bar{B}}\}\} \wedge$ ◇Receive$(B, \{A, B, m\}, \eta') \wedge$ ActionsInOrder(Receive$(B, \{A, B, m\}, \eta')$, New(B, n, η'), Send$(B, \{B, A, \{n, \{\!|m, n, A|\!\}_{\bar{B}}\}\}, \eta')$))))
2, 3, 11, AF3	Honest$(B) \supset$ After(Send$(A, \{A, B, m\}, \eta)$, Receive$(B, \{A, B, m\}, \eta')$)
11, AF2	Honest$(B) \supset$ After(Receive$(B, \{A, B, m\}, \eta')$, Send$(B, \{B, A, \{n, \{\!|m, n, A|\!\}_{\bar{B}}\}\}, \eta')$)
11, 4, AF3	Honest$(B) \supset$ After(Send$(B, \{B, A, \{n, \{\!|m, n, A|\!\}_{\bar{B}}\}\}, \eta')$, Receive$(A, \{B, A, \{n, \{\!|m, n, A|\!\}_{\bar{B}}\}\}, \eta)$)
10 – 13, AF2	Honest$(B) \supset \exists\eta'.$(ActionsInOrder(Send$(A, \{A, B, m\}, \eta)$, Receive$(B, \{A, B, m\}, \eta')$, Send$(B, \{B, A, \{n, \{\!|m, n, A|\!\}_{\bar{B}}\}\}, \eta')$, Receive$(A, \{B, A, \{n, \{\!|m, n, A|\!\}_{\bar{B}}\}\}, \eta)$))

Table 8. Deductions of A executing Init role of CR

What does proof tell us?

- Soundness Theorem:
  - If Q |- ϕ then Q |= ϕ
  - If ϕ is provable about protocol Q, then ϕ is true about protocol Q.
- ϕ true in every run of Q
  - Dolev-Yao intruder
  - Unbounded number of participants

Weak Challenge-Response



InitWCR(A, X) = [
  new m;
  send A, X, {m};
  receive X, A, {x, sig_X{m, x}};
  send A, X, sig_A{m, x}};
]

RespWCR(B) = [
  receive Y, B, {y};
  new n;
  send B, Y, {n, sig_B{y, n}};
  receive Y, B, sig_Y{y, n}};
]

Correctness of WCR – step 1

InitWCR(A, X) = [
  new m;
  send A, X, {m};
  receive X, A, {x, sig_X{m, x}};
  send A, X, sig_A{m, x}};
]

RespWCR(B) = [
  receive Y, B, {y};
  new n;
  send B, Y, {n, sig_B{y, n}};
  receive Y, B, sig_Y{y, n}};
]

1. A reasons about it's own actions

WCR |- [InitWCR(A, B)]$_A$
◇ Verify(A, sig_B {m, n})

Correctness of WCR – step 2

InitWCR(A, X) = [
  new m;
  send A, X, {m};
  receive X, A, {x, sig_X{m, x}};
  send A, X, sig_A{m, x}};
]

RespWCR(B) = [
  receive Y, B, {y};
  new n;
  send B, Y, {n, sig_B{y, n}};
  receive Y, B, sig_Y{y, n}};
]

2. Properties of signatures

CR |- [InitCR(A, B)]$_A$ Honest(B) $\supset$
$\exists$ m' (◇Send(B, m') $\wedge$ Contains(m', sig_B {m, n, A})

Correctness of WCR – Honesty

InitWCR(A, X) = [
  new m;
  send A, X, {m};
  receive X, A, {x, sig_X{m, x}};
  send A, X, sig_A{m, x}};
]

RespWCR(B) = [
  receive Y, B, {y};
  new n;
  send B, Y, {n, sig_B{y, n}};
  receive Y, B, sig_Y{y, n}};
]

Honesty invariant

CR |- Honest(X) $\wedge$
◇Send(X, m') $\wedge$ Contains(m', sig_x {y, x}) $\wedge$ $\neg$ ◇New(X, y) $\supset$
m= X, Z, {x, sig_B{y, x}} $\wedge$ ◇Receive(X, {Z, X, {y, Z}})

Correctness of WCR – step 3

InitWCR(A, X) = [
  new m;
  send A, X, {m};
  receive X, A, {x, sig_X{m, x}};
  send A, X, sig_A{m, x}};
]

RespWCR(B) = [
  receive Y, B, {y};
  new n;
  send B, Y, {n, sig_B{y, n}};
  receive Y, B, sig_Y{y, n}};
]

3. Use Honesty rule

WCR |- [InitWCR(A, B)]$_A$ Honest(B) $\supset$
◇ Receive(B, {Z,B,m}),

Result

- WCR does not have the strong authentication property for the initiator
- Counterexample
  - Intruder can forge senders and receivers identity in first two messages
    - A -> X(B) m
    - X(C) -> B m
    - B -> X(C) n, sig_B(m, n)
    - X(B) ->A n, sig_B(m, n)

Extensions

- Add Diffie-Hellman primitive
  - Can prove authentication and secrecy for key exchange protocols (STS, ISO-97898-3)
- Add symmetric encryption, hashing
  - Can prove authentication for ISO-9798-2, SKID3

Composition Rules

- Prove assertions from invariants

$$\Gamma \vdash \varphi\,[\ldots]_P\,\psi$$

- Invariant weakening rule

$$\frac{\Gamma \vdash \varphi\,[\ldots]_P\,\psi}{\Gamma \cup \Gamma' \vdash \varphi\,[\ldots]_P\,\psi}$$

If combining protocols, extend assertions to combined invariants

- Prove invariants from protocol

$$\frac{Q \blacktriangleright \Gamma \qquad Q' \blacktriangleright \Gamma}{Q \bullet Q' \blacktriangleright \Gamma}$$

Use honesty (invariant) rule to show that both protocols preserve assumed invariants

Combining protocols

$DH \blacktriangleright \overbrace{\text{Honest}(X) \supset \ldots}^{\Gamma}$ $\qquad$ $CR \blacktriangleright \overbrace{\text{Honest}(X) \supset \ldots}^{\Gamma'}$

$\Gamma \vdash$ Secrecy $\qquad$ $\Gamma' \vdash$ Authentication

$\Gamma \cup \Gamma' \vdash$ Secrecy $\qquad$ $\Gamma \cup \Gamma' \vdash$ Authentication

$\Gamma \cup \Gamma' \vdash \text{Secrecy} \wedge \text{Authentication}$

$DH \bullet CR \blacktriangleright \Gamma \cup \Gamma'$

$\parallel$

$ISO \blacktriangleright \text{Secrecy} \wedge \text{Authentication}$

Protocol Templates

- Protocols with function variables instead of specific operations
  - One template can be instantiated to many protocols
- Advantages:
  - proof reuse
  - design principles/patterns

Extending Formalism

- Language Extension
  - Add function variables to term language for cords and logic (HOL)
- Semantics
  - $Q \models \varphi \Leftrightarrow \sigma Q \models \sigma\varphi$, for all substitutions σ eliminating all function variables
- Soundness Theorem
  - Every provable formula is valid

Example

Challenge-Response Template

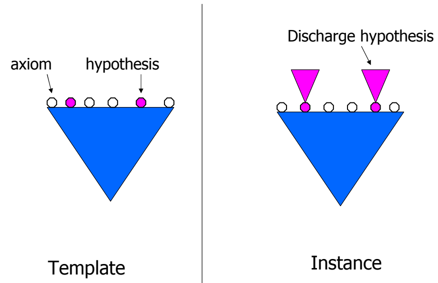
$A \rightarrow B$: m
$B \rightarrow A$: n, F(B,A,n,m)
$A \rightarrow B$: G(A,B,n,m)

Abstraction

ISO-9798-2	SKID3	ISO-9798-3
$A \rightarrow B$: m	$A \rightarrow B$: m	$A \rightarrow B$: m
$B \rightarrow A$: n, E_{KAB}(n,m,B)	$B \rightarrow A$: n, H_{KAB}(n,m,B)	$B \rightarrow A$: n, sig_B(n,m,A)
$A \rightarrow B$: E_{KAB}(n,m)	$A \rightarrow B$: H_{KAB}(n,m,A)	$A \rightarrow B$: sig_A(n,m,B)

Instantiation

Proof Structure



Modular proof techniques (2)

- **Combining protocol templates**
  - If protocol P is a hypotheses-respecting instance of two different templates, then it has the properties of both.
- **Benefits:**
  - Modular proofs of properties
  - Formalization of protocol refinements

Refinement Example Revisited

Encrypt Signatures

$A \to B$: g^a, A
$B \to A$: g^b, E_K { sig_B {g^a, g^b, A} }
$A \to B$: E_K { sig_A {g^a, g^b, B} }

- Two templates:
  - Template 1: authentication + shared secret
    - (Preserves existing properties; proof reused)
  - Template 2: identity protection (encryption)
    - (Adds new property)

Authenticated key exchange

AKE1

$A \to B$: g^a, A
$B \to A$: g^b, F(B,A,g^b,g^a)
$A \to B$: G(A,B,g^a,g^b)

ISO-9798-3, JFKi

- Shared secret
- Stronger authentication
- Identity protection for B
- Non-repudiation

AKE2

$A \to B$: g^a
$B \to A$: g^b, F(B,g^b,g^a), F'(B,g^{ab})
$A \to B$: G(A,g^a, g^b), G'(A,g^{ab})

STS, JFKr, IKEv2, SIGMA

- Shared secret
- Weaker authentication
- Identity protection for A
- Repudiability

H. Krawczyk: The Cryptography of the IPSec and IKE Protocols [CRYPTO'03]

Sample projects using this method

- Key exchange
  - STS family, JFK, IKEv2
  - Diffie-Hellman -> MQV
  - GDOI [Meadows, Pavlovic]
- Work in progress, mostly done
  - SSL verification
  - Wireless 802.11i
- Implementation of logic
  - Student project, using Isabelle

Symbolic vs Computational model

- Suppose Γ |- $[\text{actions}]_X\ \varphi$
  - If a protocol P satisfies invariants Γ, then if X does *actions*, φ will be true
- Symbolic soundness
  - No idealized adversary acting against "perfect" cryptography can make φ fail
- Computational soundness
  - No probabilistic polytime adversary can make φ fail with nonegligible probability