Logic for Computer Security Protocols

Ante Derek

Outline

- Last lecture
  - Floyd-Hoare logic of programs
  - BAN logic
- Today
  - Compositional Logic for Proving Security Properties of Protocols

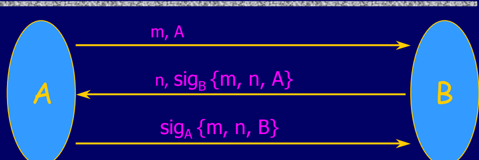
Intuition

- 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 sends m if it received m'
  - If *server not corrupt* and I receive m signed by server, then server received m'

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 specific]
- Alice deduces: Received (B, msg1) ∧ Sent (B, msg2)

Formalizing the Approach

- Language for protocol description
  - Arrows-and-messages are informal.
- Protocol Semantics
  - How does the protocol execute?
- Protocol logic
  - Stating security properties.
- Proof system
  - Formally proving security properties.

Cords

- "protocol programming language"
  - A protocol is described by specifying a "program" for each role
    - 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

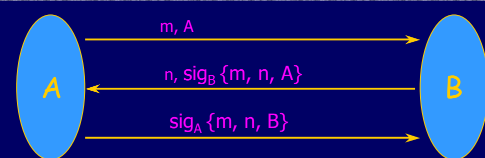
Example: x, $sig_B\{m, x, A\}$ is a term

Actions

send t;	send a term t
receive x;	receive a term into variable x
match t/p(x);	match term t against p(x)

- A Cord is just a sequence of actions
- Notation:
  - we often omit match actions
  - receive $sig_B\{A, n\}$ = receive x; match x/$sig_B\{A, n\}$

Challenge-Response as Cords



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

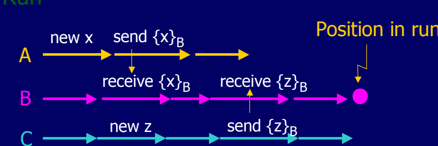
RespCR(B) = [
receive Y, B, {y, Y};
new n;
send B, Y, {n, $sig_B\{y, n, Y\}$};
receive Y, B, $sig_Y\{y, n, B\}$;
]

Cord Spaces

- Cord space is a multiset of cords
- Cords may react
  - via communication
  - via internal actions
- Sample reaction steps:
  - Communication:
    [S; send t; S'] ⊗ [T; receive x; T'] ⇒ [S; S'] ⊗ [T; T'(t/x)]
  - Matching:
    [S; match p(t)/p(x); S'] ⇒ [S; S'(t/x)]

Execution Model

- Initial configuration
  - Protocol is a finite set of roles
  - Set of principals and keys
  - Assignment of ≥1 role to each principal
- Run



Logical assertions

- Modal operator
  - [*actions*]$_P$ ϕ - after actions, P reasons ϕ
- Predicates in ϕ
  - Send(X,m) - principal X sent message m
  - Receive(X,m) - principal X received message m
  - Verify(X,m) - X verified signature m
  - Has(X,m) - X created m or received msg containing m and has keys to extract m from msg
  - Honest(X) - X follows rules of protocol

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)

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

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}}})
)

Security Properties

- Shared secret
  NS |= [InitNS(A, B)]$_A$ Honest(B) ⊃
  (Has(X, m) ⊃ X=A $\wedge$ X=B)

Proof System

- Goal: formally prove properties
- 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
  - $[\text{ new } x\ ]_P\ \ \text{Has}(P,x)$
  - $[\text{ new } x\ ]_P\ \ \text{Has}(Y,x) \supset Y{=}P$
- Actions
  - $[\text{ send } m\ ]_P\ \Diamond\text{Send}(P,m)$
- Knowledge
  - $[\text{receive } m\ ]_P\ \ \text{Has}(P,m)$
- Verify
  - $[\text{ match } x/\text{sig}_X\{m\}\ ]_P\ \Diamond\ \text{Verify}(P,m)$

Reasoning about knowledge

- Pairing
  - $\text{Has}(X, \{m,n\}) \supset \text{Has}(X, m) \wedge \text{Has}(X, n)$
- Encryption
  - $\text{Has}(X, \text{enc}_K(m)) \wedge \text{Has}(X, K^{-1}) \supset \text{Has}(X, m)$

Encryption and signature

- Public key encryption

  $\text{Honest}(X) \wedge \Diamond\text{Decrypt}(Y, \text{enc}_X\{m\}) \supset X{=}Y$

- Signature

  $\text{Honest}(X) \wedge \Diamond\text{Verify}(Y, \text{sig}_X\{m\}) \supset$
  $\exists\, m' \,(\Diamond\text{Send}(X, m') \wedge \text{Contains}(m', \text{sig}_X\{m\})$

Sample inference rules

- Preservation rules

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

- Generic rules

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

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 $A \subseteq R$.

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

- 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.

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

- 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) $\supset$
$\exists$ m' ($\Diamond$Send(B, m') $\wedge$ 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) $\wedge$
$\Diamond$Send(X, m') $\wedge$ Contains(m', sig_x {y, x, Y}) $\wedge$ $\neg$ $\Diamond$New(X, y) $\supset$
m= X, Y, {x, sig_B{y, x, Y}} $\wedge$ $\Diamond$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) $\supset$
  $\Diamond$ 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

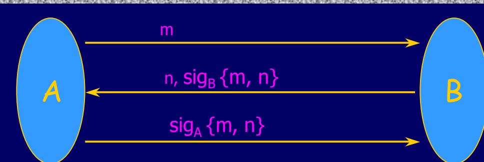
AM1	$(A\ B\ \eta)[\,]_{A,\eta}$ $\mathsf{Has}(A, A, \eta) \wedge \mathsf{Has}(A, B, \eta)$
AN3	$[(\nu m)]_{A,\eta}$ $\mathsf{Fresh}(A, m, \eta)$
AA1	$[\langle A, B, m\rangle]_{A,\eta}$ $\diamond\mathsf{Send}(A, \{A, B, m\}, \eta)$
AA1	$[(B, A, n, \{\|m, n, A\|\}_{\overline{B}})]_{A,\eta}$ $\diamond\mathsf{Receive}(A,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

Table 8. Deductions of A executing Init role of CR

We have a proof. So what?

- Soundness Theorem:
  - if Q |- ϕ then Q |= ϕ
  - If ϕ is a theorem then ϕ is a valid formula
- ϕ holds in any step in any run of protocol Q
  - Unbounded number of participants
  - Dolev-Yao intruder

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) ⊃

   ∃ m' (◇Send(B, m') ∧ 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) ∧
◇Send(X, m') ∧ Contains(m', sig_X {y, x}) ∧ ¬ ◇New(X, y) ⊃
m= X, Z, {x, sig_B{y, x}} ∧ ◇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) ⊃
◇ 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)

Benchmarks

- Can prove authentication for CR
- Proof fails for WCR
- Can prove repaired NSL protocol
- Proof fails for original NS protocol
- Proof fails for a variant of GDOI protocol (C. Meadows, D. Pavlovic)

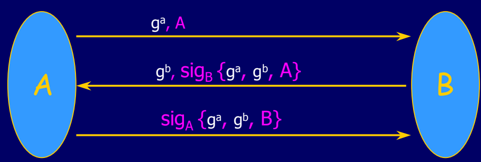
Extensions

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

Derivation system

- Protocol derivation
  - Build security protocols by combining parts from standard sub-protocols
- Proof of correctness
  - Prove protocols correct using logic that follows steps of derivation
- Reuse proofs

ISO-9798-3 Key Exchange



- Authentication
  - Do we need to prove it from scratch?
- Shared secret: g^{ab}

Abstract challenge response

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

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

- Free variables m and n instead of nonces
- Modal form: ϕ [actions] φ
  - precondition: Fresh(A,m)
  - actions: [InitACR $]_A$
  - postcondition: Honest(B) $\supset$ Authentication
- Secrecy is proved from properties of Diffie-Hellman

Parallel protocol composition

- Assume that agents run both CR and NSL using same public/private keys
  - Is authentication property preserved?
- Honesty rule is only protocol specific step in the proof sytem
  - Properties are preserved if the new protocol satisfies honesty invariants

Combining protocols

Γ (over) CR ▶ Honest(X) $\supset$ …

Γ |- CRAuthentication

$\Gamma \cup \Gamma'$ |- CRAuthentication

Γ' (over) NSL ▶ Honest(X) $\supset$ …

Γ' |- NSLAuthentication

$\Gamma \cup \Gamma'$ |- NSLAuthentication

$\Gamma \cup \Gamma'$ |- CRAuthentication $\wedge$ NSLAuthentication

CR • NSL ▶ $\Gamma \cup \Gamma'$

||

CR • NSL ▶ CRAuthentication $\wedge$ NSLAuthentication

Current work

- Formalize protocol refinements and transformations
- Automate proofs