

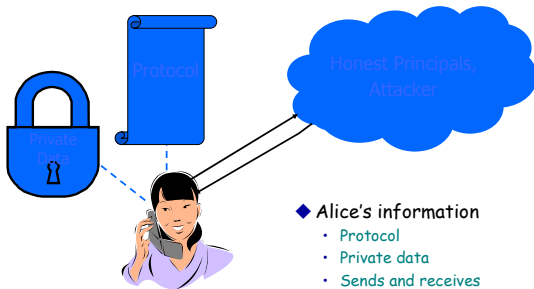
## Protocol Composition Logic

John Mitchell  
Stanford

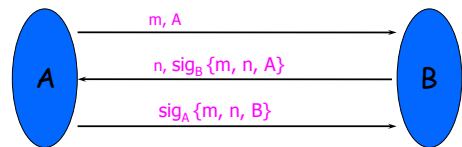
## Intuition

- ◆ 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



## 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  $\text{sig}_B\{m, n, A\}$ ,
    - he sends it as part of msg2 of the protocol and
    - he must have received msg1 from Alice [protocol dependent]
  - Alice deduces:  $\text{Received}(B, \text{msg1}) \wedge \text{Sent}(B, \text{msg2})$

## 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

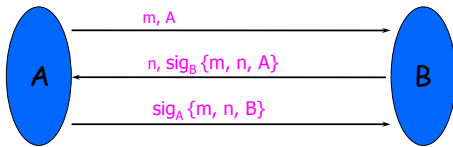
$t ::= c$	constant term
$x$	variable
$N$	name
$K$	key
$t, t$	tupleing
$\text{sig}_k\{t\}$	signature
$\text{enc}_k\{t\}$	encryption

Example:  $x, \text{sig}_B\{m, x, A\}$  is a term

## Actions and Cords

- ◆ **Actions**
  - $\text{send } t;$  send a term  $t$
  - $\text{receive } x;$  receive a term into variable  $x$
  - $\text{match } t/p(x);$  match term  $t$  against  $p(x)$
- ◆ **Cord**
  - Sequence of actions
- ◆ **Notation**
  - Some match actions are omitted in slides
    - $\text{receive } \text{sig}_B\{A, n\}$  means
    - $\text{receive } x; \text{match } x/\text{sig}_B\{A, n\}$

## Challenge-Response as Cords



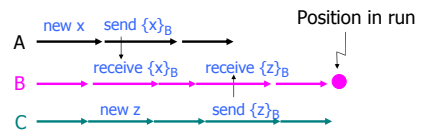
```

InitCR(A, X) = [
  new n;
  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\};
]
  
```

## Execution Model

- ◆ **Protocol**
  - Cord gives program for each protocol role
- ◆ **Initial configuration**
  - Set of principals and keys
  - Assignment of  $\geq 1$  role to each principal
- ◆ **Run**



## Formulas true at a position in run

- ◆ **Action formulas**
  - $a ::= \text{Send}(P, m) \mid \text{Receive}(P, m) \mid \text{New}(P, t)$
  - $\mid \text{Decrypt}(P, t) \mid \text{Verify}(P, t)$
- ◆ **Formulas**
  - $\varphi ::= a \mid \text{Has}(P, t) \mid \text{Fresh}(P, t) \mid \text{Honest}(N)$
  - $\mid \text{Contains}(t_1, t_2) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists x \varphi$
  - $\mid \bigcirc\varphi \mid \diamond\varphi$
- ◆ **Example**
  - $\text{After}(a, b) = \diamond(b \wedge \bigcirc\diamond a)$

## Modal Formulas

- ◆ **After actions, postcondition**
    - $[\text{actions}]_P \varphi$  where  $P = (\text{princ}, \text{role id})$
  - ◆ **Before/after assertions**
    - $\varphi [\text{actions}]_P \psi$
  - ◆ **Composition rule**

$$\frac{\varphi [S]_P \psi \quad \psi [T]_P \theta}{\varphi [ST]_P \theta}$$
- Note: same P in all formulas*

## Security Properties

### ◆ Authentication for Initiator

$$\begin{aligned} CR \models [ \text{InitCR}(A, B) ]_A \text{Honest}(B) \supset \\ \text{ActionsInOrder}(\quad \\ \quad \text{Send}(A, \{A, B, m\}), \\ \quad \text{Receive}(B, \{A, B, m\}), \\ \quad \text{Send}(B, \{B, A, \{n, \text{sig}_B\{m, n, A\}\}\}), \\ \quad \text{Receive}(A, \{B, A, \{n, \text{sig}_B\{m, n, A\}\}\}) \\ \quad ) \end{aligned}$$

### ◆ Shared secret

$$\begin{aligned} NS \models [ \text{InitNS}(A, B) ]_A \text{Honest}(B) \supset \\ (\text{Has}(X, m) \supset X=A \wedge X=B) \end{aligned}$$

## 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

- $Q, R \models [ \text{actions} ]_P \phi$   
Some role of P in R does exactly *actions* and  $\phi$  is true in state after *actions* completed
- $Q \models [ \text{actions} ]_P \phi$   
 $Q, R \models [ \text{actions} ]_P \phi$  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

- $[ \text{new } x ]_P \text{Has}(P, x)$
- $[ \text{new } x ]_P \text{Has}(Y, x) \supset Y=P$

### ◆ 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

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

## Sample inference rules

### ◆ Preservation rules

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

### ◆ Generic rules

$$\frac{\psi [ \text{actions} ]_p \phi \quad \psi [ \text{actions} ]_p \varphi}{\psi [ \text{actions} ]_p \phi \wedge \varphi}$$

## 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  $A \subseteq R$ .

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

- 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  $A \subseteq R$ .

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

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

## Correctness of CR

<pre>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}; ]</pre>	<pre>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}; ]</pre>
---	---

CR  $\vdash [ \text{InitCR}(A, B) ]_A \text{Honest}(B) \supset$   
 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

<pre>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}; ]</pre>	<pre>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}; ]</pre>
---	---

### 1. A reasons about it's own actions

CR  $\vdash [ \text{InitCR}(A, B) ]_A$   
 $\diamond \text{Verify}(A, \text{sig}_B \{m, n, A\})$

## 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

$CR \vdash [ \text{InitCR}(A, B) ]_A \text{Honest}(B) \supset$   
 $\exists m' (\diamond \text{Send}(B, m') \wedge \text{Contains}(m', \text{sig}_B\{m, n, A\}))$

## 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

$CR \vdash \text{Honest}(X) \wedge$   
 $\diamond \text{Send}(X, m') \wedge \text{Contains}(m', \text{sig}_x\{y, x, Y\}) \wedge \neg \diamond \text{New}(X, y) \supset$   
 $m = X, Y, \{x, \text{sig}_x\{y, x, Y\}\} \wedge \diamond \text{Receive}(X, \{Y, X, \{y, Y\}\})$

## 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

$CR \vdash [ \text{InitCR}(A, B) ]_A \text{Honest}(B) \supset$   
 $\diamond \text{Receive}(B, \{A, B, m\}),$

## 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

$CR \vdash [ \text{InitCR}(A, B) ]_A \text{Honest}(B) \supset \text{Auth}$

## Complete proof

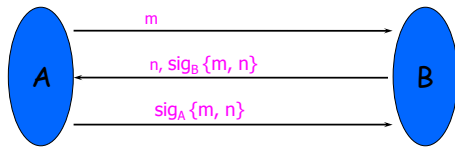
**AM1**  $(A \ B \ n) \vdash_{\text{ax}} \text{Has}(A, A, n) \wedge \text{Has}(A, B, n)$   
**ANB**  $\{0\} \vdash_{\text{ax}} \text{Fresh}(A, m, n)$   
**AA1**  $\{A, B, m\} \vdash_{\text{ax}} \diamond \text{Send}(A, \{A, B, m\}, n)$   
**AA1**  $\{B, A, n, \{m, n, A\}\} \vdash_{\text{ax}}$   
 $\diamond \text{Receive}(A, \{B, A, n, \{m, n, A\}\}, n)$   
**AA1**  $\{[m, n, A]\} \vdash_{\text{ax}} \diamond \text{Verify}(A, \{m, n, A\}, n)$   
**AA1**  $\{A, B, \{m, n, A\}\} \vdash_{\text{ax}} \diamond \text{Send}(A, \{A, B, \{m, n, A\}\}, n)$   
**AA1**  $(A \ B \ n) \vdash_{\text{ax}} \{A, B, m\} \vdash (x/B, A, n, \{m, n, A\})$   
**AF1, AF2**  $\{[m, n, A]\} \vdash_{\text{ax}} \{A, B, \{m, n, A\}\} \vdash_{\text{ax}}$   
 $\text{ActionOrder}(\text{Send}(A, \{A, B, m\}, n), \text{Receive}(A, \{B, A, n, \{m, n, A\}\}, n),$   
 $\text{Send}(A, \{A, B, \{m, n, A\}\}, n))$   
**N1**  $\diamond \text{New}(A, m, n) \supset \diamond \text{New}(B, m, n')$   
**5, VER**  $\text{Honest}(B) \wedge \diamond \text{Verify}(A, \{m, n, A\}, n) \supset$   
 $\exists m'. \text{Send} \wedge \diamond \text{Send}(B, m', n') \wedge \{[m, n, A]\} \subseteq m'$   
**HON**  $\text{Honest}(B) \supset \exists m'. \text{Send} \wedge (\diamond \text{Send}(B, m', n') \wedge$   
 $\{m, n, A\} \subseteq m' \wedge \neg \diamond \text{New}(B, m, n')) \supset$   
 $(m' = \{B, A, \{n, \{m, n, A\}\}\} \wedge \diamond \text{Receive}(B, \{A, B, m\}, n')) \wedge$   
 $\text{ActionOrder}(\text{Receive}(B, \{A, B, m\}, n'), \text{New}(B, m, n'),$   
 $\text{Send}(B, \{B, A, \{n, \{m, n, A\}\}, n'))$   
**2, 3, 11, AF3**  $\text{Honest}(B) \supset \text{Auth}(\text{Send}(A, \{A, B, m\}, n),$   
 $\text{Receive}(B, \{A, B, m\}, n'))$   
**11, AF2**  $\text{Honest}(B) \supset \text{Auth}(\text{Receive}(B, \{A, B, m\}, n'),$   
 $\text{Send}(B, \{B, A, \{n, \{m, n, A\}\}, n'))$   
**11, 4, AF3**  $\text{Honest}(B) \supset \text{Auth}(\text{Send}(B, \{B, A, \{n, \{m, n, A\}\}, n'),$   
 $\text{Receive}(A, \{B, A, \{n, \{m, n, A\}\}, n))$   
**10 - 13, AF2**  $\text{Honest}(B) \supset \exists m'. (\text{ActionOrder}(\text{Send}(A, \{A, B, m\}, n),$   
 $\text{Receive}(B, \{A, B, m\}, n'), \text{Send}(B, \{B, A, \{n, \{m, n, A\}\}, n'),$   
 $\text{Receive}(A, \{B, A, \{n, \{m, n, A\}\}, n))$

Table 8. Deductions of  $A$  executing  $\text{Init}$  role of CR

## What does proof tell us?

- ◆ Soundness Theorem:
  - If  $Q \vdash \phi$  then  $Q \models \phi$
  - If  $\phi$  is provable about protocol  $Q$ , then  $\phi$  is true about protocol  $Q$ .
- ◆  $\phi$  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 its own actions

WCR |- [ InitWCR(A, B) ]<sub>A</sub>  
 ◇ Verify(A, sig<sub>B</sub> {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) ]<sub>A</sub> Honest(B) ⊃  
 ∃ m' (◇ Send(B, m') ∧ Contains(m', sig<sub>B</sub> {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<sub>x</sub> {y, x}) ∧ ¬ ◇ New(X, y) ⊃  
 m = X, Z, {x, sig<sub>z</sub> {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) ]<sub>A</sub> 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<sub>B</sub>(m, n)
- X(B) → A n, sig<sub>B</sub>(m, n)

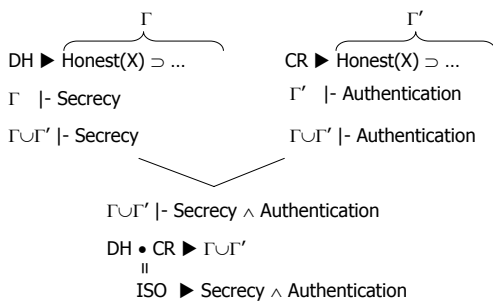
## Extensions

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

## Composition Rules

- ◆ Prove assertions from invariants
 
$$\Gamma \vdash \varphi \dots P \ \psi$$
- ◆ Invariant weakening rule
 
$$\frac{\Gamma \vdash \varphi \dots P \ \psi}{\Gamma \cup \Gamma' \vdash \varphi \dots P \ \psi} \quad \text{If combining protocols, extend assertions to combined invariants}$$
- ◆ Prove invariants from protocol
 
$$\frac{Q \triangleright \Gamma \quad Q' \triangleright \Gamma'}{Q \bullet Q' \triangleright \Gamma} \quad \text{Use honesty (invariant) rule to show that both protocols preserve assumed invariants}$$

## Combining protocols



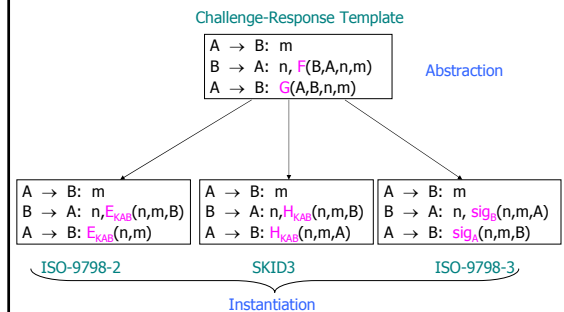
## Protocol Templates

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

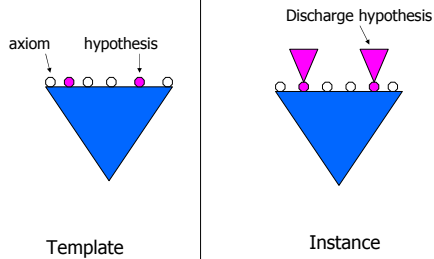
## Extending Formalism

- ◆ 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  $\sigma$  eliminating all function variables
- ◆ Soundness Theorem
  - Every provable formula is valid

## Example



## 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

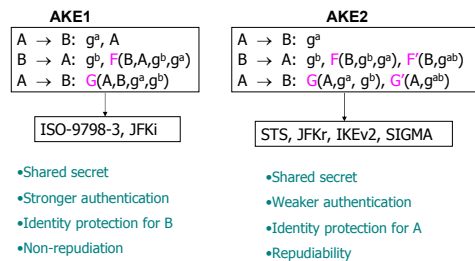
### Encrypt Signatures

```

A → B:  $g^a, A$ 
B → A:  $g^b, E_K \{ \text{sig}_B \{ g^a, g^b, A \} \}$ 
A → B:  $E_K \{ \text{sig}_A \{ g^a, g^b, B \} \}$ 
    
```

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

## Authenticated key exchange



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