Summary of Protocol Logic

February 15, 2005

1 Protocol Logic

1.1 Cord calculus syntax

(names)	N	::=	$\hat{X}$	name
(session id)	S	::=	η	session id
(thread)	P	::=	$\langle N, S \rangle$	thread
(keys)	K	::=	K	key
			name(N)	name
(nonces)	n	::=	x	nonce
(terms)	t	::=	x	variable term
			nonce(n)	nonce
			na(N)	name
			thread(P)	thread
			key(K)	key
			$t, \ldots, t$	tuple of terms
			$ENC[K](t)$	term encrypted with key K
			$SIG[K](t)$	term signed with key $\overline{K}$
(actions)	a	::=	`noop`	the null action
			`send` t	send a term t
			`receive` x	receive term into variable x
			`new` x	generate new term x
			`match` t/t	match a term to a pattern
(strands)	S	::=	$a; S \mid a$	
(cords)	C	::=	$N[S]$	

Example 1.1 *One-way Challenge response:*

$$\textbf{Init} = X(X, \hat{Y})[\texttt{new}\ x;\ \texttt{send}\ \hat{X}, \hat{Y}, x;\ \texttt{receive}\ \hat{Y}, \hat{X}, y, z;\ \texttt{match}\ z/SIG[\hat{Y}](y, x, \hat{X})]$$

$$\textbf{Resp} = Y(Y)[\texttt{receive}\ \hat{X}, \hat{Y}, x;\ \texttt{new}\ y;\ \texttt{send}\ \hat{Y}, \hat{X}, y, SIG[\hat{Y}](y, x, \hat{X})]$$

1.2 Logic syntax

Action formulas
$\mathsf{a} ::= \mathsf{Send}(P,t) \mid \mathsf{Receive}(P,t) \mid \mathsf{New}(P,t) \mid \mathsf{Decrypt}(P,t) \mid \mathsf{Verify}(P,t)$
Formulas
$\phi ::= \mathsf{a} \mid \mathsf{a} < \mathsf{a} \mid \mathsf{Has}(P,t) \mid \mathsf{Fresh}(P,t) \mid \mathsf{FirstSend}(P,t,t') \mid$
$\mathsf{Honest}(N) \mid \mathsf{Contains}(t_1,t_2) \mid \phi \wedge \phi \mid \neg\phi \mid \exists x.\phi \mid \mathsf{Start}(P)$
Modal formulas
$\Psi ::= \{\phi,\ \rho,\ \phi\}$

2 Proof System

2.1 Axioms for Protocol Actions

Axioms for protocol actions state properties that hold in the state as a result of (not) executing certain actions. Note that the a in axioms is any one of the 5 actions and a is the corresponding predicate in the logic.

AA1 $\phi[a]_X\ \mathsf{a}$

AA2 $\mathsf{Start}(X)[\,]_X\ \neg\mathsf{a}(\mathsf{X})$

AA3 $\neg\mathsf{Send}(X,t)[b]_X\neg\mathsf{Send}(X,t)$ if $\sigma\mathsf{Send}(X,t) \neq \sigma\mathsf{b}$ for all substitutions σ
(side condition for AA3 states that b cannot be unified with $\mathsf{Send}(\mathsf{X},\mathsf{t})$)

AA4 $\phi[a_1;a_2;\ldots;a_k]_X\mathsf{a}_1 < \mathsf{a}_2 \wedge \ldots \wedge \mathsf{a}_{k-1} < \mathsf{a}_k$

AA5 $\neg(\mathsf{Send}(X,m) \wedge \mathsf{Contains}(m,t))[b]_X\neg(\mathsf{Send}(X,m) \wedge \mathsf{Contains}(m,t))$
if $\sigma\mathsf{Send}(X,t) \neq \sigma\mathsf{b}$ for all substitutions σ

AN2 $\phi[\mathsf{new}\ \ x]_X\ \mathsf{Has}(Y,x) \supset (Y = X)$

AN3 $\phi[\mathsf{new}\ \ x]_X\ \mathsf{Fresh}(X,x)$

ARP $\mathsf{Receive}(X,p(x))[\mathtt{match}\ \ q(x)/q(t)]_X\ \mathsf{Receive}(X,p(t))$

2.2 Possession Axioms

The possession axioms characterize the terms that a principal can derive if it possesses certain other terms.

ORIG $\mathsf{New}(X,x) \supset \mathsf{Has}(X,x)$

REC $\mathsf{Receive}(X,x) \supset \mathsf{Has}(X,x)$

TUP $\mathsf{Has}(X,x) \wedge \mathsf{Has}(X,y) \supset \mathsf{Has}(X,(x,y))$

ENC $\mathsf{Has}(X,x) \wedge \mathsf{Has}(X,K) \supset \mathsf{Has}(X,ENC[K](x))$

PROJ $\mathsf{Has}(X,(x,y)) \supset \mathsf{Has}(X,x) \wedge \mathsf{Has}(X,y)$

DEC $\mathsf{Has}(X,ENC[K](x)) \wedge \mathsf{Has}(X,K) \supset \mathsf{Has}(X,x)$

2.3 Encryption and Signature

The next two axioms are aimed at capturing the black-box model of encryption and signature. **VER** refers to the unforgeability of signatures while **SEC** stipulates the need to possess the private key in order to decrypt a message encrypted with the corresponding public key.

SEC $\mathsf{Honest}(\hat{X}) \wedge \mathsf{Decrypt}(Y, ENC[\hat{X}](x)) \supset (\hat{Y} = \hat{X})$

VER $\mathsf{Honest}(\hat{X}) \wedge \mathsf{Verify}(Y, SIG[\hat{X}](x)) \wedge \hat{X} \neq \hat{Y} \supset \exists m.\exists s.\mathsf{Send}(\langle \hat{X}, s\rangle, m) \wedge \mathsf{Contains}(m, SIG[\hat{X}](x))$

2.4 Generic Rules

$$\frac{\theta[P]_X\phi \quad \theta[P]_X\psi}{\theta[P]_X\phi \wedge \psi}\ \mathbf{G1} \qquad \frac{\theta[P]_X\psi \quad \phi[P]_X\psi}{\theta \wedge \phi[P]_X\psi}\ \mathbf{G2} \qquad \frac{\theta[P]_X\phi \quad \theta' \supset \theta \quad \phi \supset \phi'}{\theta'[P]_X\phi'}\ \mathbf{G3} \qquad \frac{\phi}{\theta[P]_X\phi}\ \mathbf{G4}$$

2.5 Sequencing rule

Sequencing rule gives us a way of sequentially composing two cords P and P' when post-condition of P, matches the pre-condition of P'.

$$\frac{\phi_1[P]_A\phi_2 \qquad \phi_2[P']_A\phi_3}{\phi_1[PP']_A\phi_3}\ \mathbf{S1}$$

2.6 Preservation Axioms

For $\mathsf{Persist} \in \{\mathsf{Has}, \mathsf{FirstSend}, \mathsf{a} < \mathsf{b}, \mathsf{a}\}$:

P1 $\mathsf{Persist}(X, t)[a]_X\mathsf{Persist}(X, t)$

P2 $\mathsf{Fresh}(X, t)[a]_X\mathsf{Fresh}(X, t)$, where $t \not\subseteq a$ or $a \neq \langle m\rangle$ (ie, it's still fresh if you didn't send it)

P3 $\mathsf{HasAlone}(X, n)[a]_X\mathsf{HasAlone}(X, n)$, where $n \not\subseteq_v a$ or $a \neq \langle m\rangle$
(P3 will not be implemented in the first pass - it's needed for Diffie-Hellmen)

$$\mathsf{HasAlone}(X, t) \equiv \mathsf{Has}(X, t) \wedge (\mathsf{Has}(Y, t) \supset X = Y)$$

2.7 Axioms and Rules for Temporal Ordering

The next two axioms give us a way of deducing temporal ordering between actions of different threads. Informally, $\mathsf{FirstSend}(X, t, t')$ says that a thread X generated a fresh term t and sent it out first in message t'. This refers to the first such send event.

FS1 $\mathsf{Fresh}(X, t)[\mathsf{send}\ t']_X\mathsf{FirstSend}(X, t, t')$, where $t \subseteq t'$.

FS2 $\mathsf{FirstSend}(X, t, t') \wedge \mathsf{a}(Y, t") \supset \mathsf{Send}(X, t') < \mathsf{a}(Y, t")$, where $X \neq Y$ and $t \subseteq t"$.

2.8 Honesty Rule

$$\frac{\mathsf{Start}(X)[\,]_X\ \phi \qquad \forall \rho \in \mathcal{Q}.\forall P \in BS(\rho).\ \phi\ [P]_X\ \phi}{\mathsf{Honest}(\hat{X}) \supset \phi}\ \mathbf{HON_Q}$$

no free variable in ϕ except X bound in $[P\,]_X$

Example 2.1

$$\begin{aligned}\mathbf{Init} &= (X,\hat{Y})[\texttt{new}\ x;\ \texttt{send}\ \hat{X},\hat{Y},x;\ \texttt{receive}\ \hat{Y},\hat{X},y,z;\ \texttt{match}\ z/SIG_{\hat{Y}}(y,x,\hat{X})]_X\\ \mathbf{Resp} &= (Y)[\texttt{receive}\ \hat{X},\hat{Y},x;\ \texttt{new}\ y;\ \texttt{send}\ \hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X})]_Y\end{aligned}$$

AA1	ϕ
	$[\texttt{match}\ z/SIG_{\hat{Y}}(y,x,\hat{X})]_X$
	$\mathsf{Verify}(X, SIG_{\hat{Y}}(y,x,\hat{X}))$ (1)
$(1), \mathbf{VER}$	ϕ
	$[\texttt{match}\ z/SIG_{\hat{Y}}(y,x,\hat{X})]_X$
	$\mathsf{Honest}(\hat{Y}) \wedge (\hat{X} \neq \hat{Y}) \supset \exists Y.\exists m.\mathsf{Send}(Y,m) \wedge \mathsf{Contains}(m, SIG_{\hat{Y}}(y,x,\hat{X}))$ (2)
ϕ_{HON1}	$\mathsf{Honest}(\hat{Y}) \wedge \mathsf{Send}(Y,m) \wedge \mathsf{Contains}(m, SIG_{\hat{Y}}(y,x,\hat{X})) \supset$
	$\quad \mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}))$ (3)
$(2), (3)$	ϕ
	$[\texttt{match}\ z/SIG_{\hat{Y}}(y,x,\hat{X})]_X$
	$\mathsf{Honest}(\hat{Y}) \wedge (\hat{X} \neq \hat{Y}) \supset \exists Y.$
	$\quad (\mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X})))$ (4)
S1	$\mathsf{Fresh}(X,x)$
	$[\texttt{send}\ \hat{X},\hat{Y},x;\ \texttt{receive}\ \hat{Y},\hat{X},y,z;\ \texttt{match}\ z/SIG_{\hat{Y}}(y,x,\hat{X})]_X$
	$\mathsf{Honest}(\hat{Y}) \wedge (\hat{X} \neq \hat{Y}) \supset \exists Y.$
	$\quad (\mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}))$ (5)
$\mathbf{FS1}, \mathbf{P1}$	$\mathsf{Fresh}(X,x)$
	$[\texttt{send}\ \hat{X},\hat{Y},x;\ \texttt{receive}\ \hat{Y},\hat{X},y,z;\ \texttt{match}\ z/SIG_{\hat{Y}}(y,x,\hat{X})]_X$
	$\mathsf{FirstSend}(X,x,\hat{X},\hat{Y},x)$ (6)
$(5), (6), \mathbf{FS2}$	$\mathsf{Fresh}(X,x)$
	$[\texttt{send}\ \hat{X},\hat{Y},x;\ \texttt{receive}\ \hat{Y},\hat{X},y,z;\ \texttt{match}\ z/SIG_{\hat{Y}}(y,x,\hat{X})]_X$
	$\mathsf{Honest}(\hat{Y}) \wedge (\hat{X} \neq \hat{Y}) \supset \exists Y.(\mathsf{Send}(X,\hat{X},\hat{Y},x) < \mathsf{Receive}(Y,\hat{X},\hat{Y},x) \wedge$
	$\quad \mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}))$ (7)
AN3	ϕ
	$[\texttt{new}\ x]_X$
	$\mathsf{Fresh}(X,x)$ (8)
$(7), (8), \mathbf{S1}$	ϕ

$[\texttt{new}\ x;\ \texttt{send}\ \hat{X},\hat{Y},x;\ \texttt{receive}\ \hat{Y},\hat{X},y,z;\ \texttt{match}\ z/SIG_{\hat{Y}}(y,x,\hat{X})]_X$

$\mathsf{Honest}(\hat{Y}) \wedge (\hat{X} \neq \hat{Y}) \supset \exists Y.(\mathsf{Send}(X,\hat{X},\hat{Y},x) < \mathsf{Receive}(Y,\hat{X},\hat{Y},x) \wedge$
$\mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}))$ (9)

ϕ_{HON2} $\mathsf{Honest}(\hat{Y}) \wedge \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}) \supset$
$\mathsf{FirstSend}(Y,y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}))$ (10)

$(9),(10),\mathbf{FS2}$ ϕ

$[\texttt{new}\ x;\ \texttt{send}\ \hat{X},\hat{Y},x;\ \texttt{receive}\ \hat{Y},\hat{X},y,z;\ \texttt{match}\ z/SIG_{\hat{Y}}(y,x,\hat{X})]_X$

$\mathsf{Honest}(\hat{Y}) \wedge (\hat{X} \neq \hat{Y}) \supset \exists Y.($
$\mathsf{Send}(X,\hat{X},\hat{Y},x) < \mathsf{Receive}(Y,\hat{X},\hat{Y},x) \wedge$
$\mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X})) \wedge$
$\mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X})) < \mathsf{Receive}(X,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}))$
$)$ (11)

Honesty rule. All basic sequences (written as executed by $\hat{Y}$) are:

$$
\begin{aligned}
BS_0 &= [\,] \\
BS_1 &= [\texttt{new}\ y;\ \texttt{send}\ \hat{Y},\hat{X},y;\ ] \\
BS_2 &= [\texttt{receive}\ \hat{X},\hat{Y},x,z;\ \texttt{match}\ z/SIG_{\hat{X}}(x,y,\hat{Y})] \\
BS_3 &= [\texttt{receive}\ \hat{X},\hat{Y},x;\texttt{new}\ y;\ \texttt{send}\ \hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X})]
\end{aligned}
$$

$$
\begin{aligned}
\phi_{HON1} &\equiv \mathsf{Honest}(\hat{Y}) \wedge \mathsf{Send}(Y,m) \wedge \mathsf{Contains}(m,SIG_{\hat{Y}}(y,x,\hat{X})) \supset \\
&\quad \mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X})) \\
\psi &\equiv \mathsf{Send}(Y,m) \wedge \mathsf{Contains}(m,SIG_{\hat{Y}}(y,x,\hat{X})) \supset \\
&\quad \mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}))
\end{aligned}
$$

$\mathbf{AA2}$ $\mathsf{Start}(Y)$
$[\,]_Y$
$\neg\mathsf{Send}(Y,m)$ (12)

(12) $\mathsf{Start}(Y)$
$[\,]_Y$
ψ (13)

$\mathbf{P1}$ $\mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}))$
$[\texttt{new}\ y;\ \texttt{send}\ \hat{Y},\hat{X},y;\ ]_Y$
$\mathsf{Receive}(Y,\hat{X},\hat{Y},x) < \mathsf{Send}(Y,\hat{Y},\hat{X},y,SIG_{\hat{Y}}(y,x,\hat{X}))$ (14)

$\mathbf{AA5}$ $\neg(\mathsf{Send}(Y,m) \wedge \mathsf{Contains}(m,SIG_{\hat{Y}}(y,x,\hat{X})))$
$[\texttt{new}\ y;\ \texttt{send}\ \hat{Y},\hat{X},y;\ ]_Y$
$\neg(\mathsf{Send}(Y,m) \wedge \mathsf{Contains}(m,SIG_{\hat{Y}}(y,x,\hat{X})))$ (15)

$(14), (15), \mathbf{G4}$ ψ

$[\texttt{new}\ y;\ \texttt{send}\ \hat{Y}, \hat{X}, y;\ ]_Y$

ψ (16)

$\mathbf{AA2}, \mathbf{AA3}$ ψ

$[\texttt{receive}\ \hat{X}, \hat{Y}, x, z;\ \texttt{match}\ z/SIG_{\hat{X}}(x, y, \hat{Y})]_Y$

$\neg \mathsf{Send}(Y, m)$ (17)

(13) ψ

$[\texttt{receive}\ \hat{X}, \hat{Y}, x, z;\ \texttt{match}\ z/SIG_{\hat{X}}(x, y, \hat{Y})]_Y$

ψ (18)

$\mathbf{AA4}$ ψ

$[\texttt{receive}\ \hat{X}, \hat{Y}, x; \texttt{new}\ y;\ \texttt{send}\ \hat{Y}, \hat{X}, y, SIG_{\hat{Y}}(y, x, \hat{X})]_Y$

$\mathsf{Receive}(Y, \hat{X}, \hat{Y}, x) < \mathsf{Send}(Y, \hat{Y}, \hat{X}, y, SIG_{\hat{Y}}(y, x, \hat{X}))$ (19)

(15) ψ

$[\texttt{receive}\ \hat{X}, \hat{Y}, x; \texttt{new}\ y;\ \texttt{send}\ \hat{Y}, \hat{X}, y, SIG_{\hat{Y}}(y, x, \hat{X})]_Y$

ψ (20)