Our old protocol . . .

$X \longrightarrow Y$: $\{M\}_{K_Y}, X$

$Y \longrightarrow X$: $\{M\}_{K_X}$

Our old protocol . . .

$X \rightarrow Y$: $\{M\}_{K_Y}, X$

$Y \rightarrow X$: $\{M\}_{K_X}$

. . . and the familiar attack

h_1 sends $\{M\}_{K_{h_2}}, h_1$

h_2 gets $\{M\}_{K_{h_2}}, d$

h_2 sends $\{M\}_{K_d}$

Our old protocol ...

$X \to Y$: $\{M\}_{K_Y}, X$

$Y \to X$: $\{M\}_{K_X}$

an attack with 4 agents ...

h_1 sends $\{M\}_{K_{h_2}}, h_1$

h_2 gets $\{M\}_{K_{h_2}}, h_3$

h_2 sends $\{M\}_{K_{h_3}}$

h_3 gets $\{M\}_{K_{h_3}}, d$

h_3 sends $\{M\}_{K_d}$

... and the familiar attack

h_1 sends $\{M\}_{K_{h_2}}, h_1$

h_2 gets $\{M\}_{K_{h_2}}, d$

h_2 sends $\{M\}_{K_d}$

Our old protocol . . .

$X \to Y$: $\{M\}_{K_Y}, X$

$Y \to X$: $\{M\}_{K_X}$

. . . and the familiar attack

h_1 sends $\{M\}_{K_{h_2}}, h_1$

h_2 gets $\{M\}_{K_{h_2}}, d$

h_2 sends $\{M\}_{K_d}$

an attack with 4 agents . . .

h_1 sends $\{M\}_{K_{h_2}}, h_1$

h_2 gets $\{M\}_{K_{h_2}}, h_3$

h_2 sends $\{M\}_{K_{h_3}}$

h_3 gets $\{M\}_{K_{h_3}}, d$

h_3 sends $\{M\}_{K_d}$

. . . after projection

h_1 sends $\{M\}_{K_{h_2}}, h_1$

h_2 gets $\{M\}_{K_{h_2}}, d$

h_2 sends $\{M\}_{K_d}$

d gets $\{M\}_{K_d}, d$

d sends $\{M\}_{K_d}$

The protocol is described as follows.

With three honest and one dishonest agent:

$\rightsquigarrow Ha(h_1)$ $\rightsquigarrow Ha(h_2)$ $\rightsquigarrow Ha(h_3)$ $\rightsquigarrow Da(d)$

The protocol is described as follows.

With three honest and one dishonest agent:

$\rightsquigarrow Ha(h_1)$ $\rightsquigarrow Ha(h_2)$ $\rightsquigarrow Ha(h_3)$ $\rightsquigarrow Da(d)$

$Ha(x) \rightsquigarrow Agent(x), Ha(x)$ $Da(x) \rightsquigarrow Agent(x), Da(x)$

The protocol is described as follows.

With three honest and one dishonest agent:

$\rightsquigarrow Ha(h_1)$ $\quad$ $\rightsquigarrow Ha(h_2)$ $\quad$ $\rightsquigarrow Ha(h_3)$ $\quad$ $\rightsquigarrow Da(d)$

$Ha(x) \rightsquigarrow Agent(x), Ha(x)$ $\quad$ $Da(x) \rightsquigarrow Agent(x), Da(x)$

$Agent(x) \rightsquigarrow I(x), I(pub(x)), Agent(x)$ $\quad$ $Da(x) \rightsquigarrow I(prv(x)), Da(x)$

The protocol is described as follows.

With three honest and one dishonest agent:

$\rightsquigarrow Ha(h_1)$ $\quad\rightsquigarrow Ha(h_2)$ $\quad\rightsquigarrow Ha(h_3)$ $\quad\rightsquigarrow Da(d)$

$Ha(x) \rightsquigarrow Agent(x), Ha(x)$ $\qquad Da(x) \rightsquigarrow Agent(x), Da(x)$

$Agent(x) \rightsquigarrow I(x), I(pub(x)), Agent(x)$ $\qquad Da(x) \rightsquigarrow I(prv(x)), Da(x)$

$\rightsquigarrow Distinct(h_1, h_2)$ $\qquad \rightsquigarrow Distinct(h_1, d)$

$\rightsquigarrow Distinct(h_2, h_1)$ $\qquad \rightsquigarrow Distinct(d, h_1)$ $\qquad \rightsquigarrow Distinct(d, d)$

$\rightsquigarrow Distinct(h_2, h_3)$ $\qquad \rightsquigarrow Distinct(h_2, d)$

$\ldots$ $\qquad \ldots$

The usual rules for intruder actions

$$
\begin{aligned}
I(x), I(y) &\leadsto I(\langle x, y\rangle), I(x), I(y) \\
I(\langle x, y\rangle) &\leadsto I(x), I(y), I(\langle x, y\rangle) \\
I(x), I(y) &\leadsto I(\{x\}_y), I(x), I(y) \\
I(\{x\}_{pub(y)}), I(prv(y)) &\leadsto I(x), I(\{x\}_{pub(y)}), I(prv(y)) \\
I(\{x\}_{prv(y)}), I(pub(y)) &\leadsto I(x), I(\{x\}_{prv(y)}), I(pub(y)) \\
&\leadsto \exists n \cdot I(n)
\end{aligned}
$$

And the protocol specific rules

$$\begin{array}{lcl} Agent(x), Agent(y), & \rightsquigarrow & A_0(x,y), B_0(x,y), Agent(x), Agent(y), \\ Distinct(x,y) & & Distinct(x,y) \end{array}$$

And the protocol specific rules

$$\begin{aligned} Agent(x), Agent(y), \\ Distinct(x, y) \end{aligned} \quad \rightsquigarrow \quad \begin{aligned} A_0(x, y), B_0(x, y), Agent(x), Agent(y), \\ Distinct(x, y) \end{aligned}$$

$$A_0(x, y) \quad \rightsquigarrow \exists z \cdot A_1(x, y, z), I(\langle \{z\}_{pub(y)}, x \rangle)$$

And the protocol specific rules

$$
\begin{aligned}
Agent(x), Agent(y), \quad & \leadsto \quad A_0(x,y), B_0(x,y), Agent(x), Agent(y), \\
Distinct(x,y) \quad & \quad Distinct(x,y) \\
A_0(x,y) \quad & \leadsto \exists z \cdot A_1(x,y,z), I(\langle \{z\}_{pub(y)}, x\rangle) \\
B_0(x,y), I(\langle \{z\}_{pub(y)}, x\rangle) \quad & \leadsto B_1(x,y,z), I(\{z\}_{pub(x)}), I(\langle \{z\}_{pub(y)}, x\rangle)
\end{aligned}
$$

And the protocol specific rules

$$
\begin{aligned}
Agent(x), Agent(y), \quad & \\
Distinct(x, y) \quad & \rightsquigarrow \quad A_0(x, y), B_0(x, y), Agent(x), Agent(y), \\
& \qquad Distinct(x, y) \\
A_0(x, y) \quad & \rightsquigarrow \exists z \cdot A_1(x, y, z), I(\langle \{z\}_{pub(y)}, x \rangle) \\
B_0(x, y), I(\langle \{z\}_{pub(y)}, x \rangle) \quad & \rightsquigarrow B_1(x, y, z), I(\{z\}_{pub(x)}), I(\langle \{z\}_{pub(y)}, x \rangle) \\
A_1(x, y, z), I(\{z\}_{pub(x)}) \quad & \rightsquigarrow A_2(x, y, z), I(\{z\}_{pub(x)})
\end{aligned}
$$

And the protocol specific rules

$$
\begin{array}{rcl}
Agent(x), Agent(y), & & A_0(x,y), B_0(x,y), Agent(x), Agent(y), \\
& \rightsquigarrow & \\
Distinct(x,y) & & Distinct(x,y) \\
A_0(x,y) & \rightsquigarrow & \exists z \cdot A_1(x,y,z), I(\langle\{z\}_{pub(y)}, x\rangle) \\
B_0(x,y), I(\langle\{z\}_{pub(y)}, x\rangle) & \rightsquigarrow & B_1(x,y,z), I(\{z\}_{pub(x)}), I(\langle\{z\}_{pub(y)}, x\rangle) \\
A_1(x,y,z), I(\{z\}_{pub(x)}) & \rightsquigarrow & A_2(x,y,z), I(\{z\}_{pub(x)})
\end{array}
$$

Security question: is a protocol state reachable containing the pattern

$$Ha(x), Ha(y), A_2(x,y,z), I(z)$$

We can apply these rules to get a protocol state of the form

$$
\begin{aligned}
& Ha(h_1), Ha(h_2), Ha(h_3), Da(d), Agent(h_1), Agent(h_2), Agent(h_3), \\
& \quad Agent(d), Distinct(h_1, h_2), Distinct(h_3, h_2), Distinct(d, h_3), \\
& \quad A_2(h_1, h_2, m), B_0(h_1, h_2), \quad A_0(h_3, h_2), B_1(h_3, h_2, m), \\
& \quad A_0(d, h_3), B_1(d, h_3, m), \quad I(\{m\}_{pub(h_2)}, h_1),\ I(\{m\}_{pub(h_2)}, h_3), \\
& \ I(\{m\}_{pub(h_3)}),\ I(\{m\}_{pub(h_3)}, d),\ I(\{m\}_{pub(d)}),\ I(m),\ I(\{m\}_{pub(h_1)}) \\
& \ I(\ldots) \ldots I(\ldots)
\end{aligned}
$$

We can apply these rules to get a protocol state of the form

$$Ha(h_1), Ha(h_2), Ha(h_3), Da(d), Agent(h_1), Agent(h_2), Agent(h_3),$$
$$Agent(d), Distinct(h_1, h_2), Distinct(h_3, h_2), Distinct(d, h_3),$$
$$A_2(h_1, h_2, m), B_0(h_1, h_2), \quad A_0(h_3, h_2), B_1(h_3, h_2, m),$$
$$A_0(d, h_3), B_1(d, h_3, m), \quad I(\{m\}_{pub(h_2)}, h_1), \; I(\{m\}_{pub(h_2)}, h_3),$$
$$I(\{m\}_{pub(h_3)}), \; I(\{m\}_{pub(h_3)}, d), \; I(\{m\}_{pub(d)}), \; I(m), \; I(\{m\}_{pub(h_1)})$$
$$I(\ldots) \ldots I(\ldots)$$

We get the following without using the rules involving h_3 (apply `proj`)

$$Ha(h_1), Ha(h_2), Da(d), Da(d), Agent(h_1), Agent(h_2), Agent(d),$$
$$Agent(d), Distinct(h_1, h_2), Distinct(d, h_2), Distinct(d, d),$$
$$A_2(h_1, h_2, m), B_0(h_1, h_2), \quad A_0(d, h_2), B_1(d, h_2, m),$$
$$A_0(d, d), B_1(d, d, m), \quad I(\{m\}_{pub(h_2)}, h_1), \quad I(\{m\}_{pub(h_2)}, d),$$
$$I(\{m\}_{pub(d)}), \; I(\{m\}_{pub(d)}, d), \; I(\{m\}_{pub(d)}), \; I(m), \; I(\{m\}_{pub(h_1)}) \ldots$$

$k+1$ is a tight bound

A toy variant of the Needham-Schroeder public key protocol:

$$\begin{aligned}
A_1 \longrightarrow A_2 :& \quad \{A_1, A_2, \ldots, A_k, N_{A_1}\}_{K_{A_2}} \\
A_2 \longrightarrow A_1 :& \quad \{N_{A_1}, N_{A_2}\}_{K_{A_1}} \\
A_1 \longrightarrow A_2 :& \quad \{N_{A_2}\}_{K_{A_2}}
\end{aligned}$$

Other steps involving $A_2, A_3, \ldots$ could be added to make it more realistic.

This is modeled using similar rules as before. The agents $A_1, \ldots, A_k$ are required to be distinct.

There is a standard attack involving $k+1$ agents.

k honest agents are required for the two nonces to be generated, and a dishonest agent for decryption of messages.

For $k = 3$ we have the following rules.

$$\begin{aligned}&Agent(x_1), Agent(x_2), Agent(x_3), Distinct(x_1, x_2), Distinct(x_2, x_3),\\ &Distinct(x_1, x_3) \rightsquigarrow A_{1,0}(x_1, x_2, x_3), A_{2,0}(x_1, x_2, x_3), Agent(x_1), Agent(x_2),\\ &\qquad Agent(x_3), Distinct(x_1, x_2), Distinct(x_2, x_3), Distinct(x_1, x_3)\end{aligned}$$

$$A_{1,0}(x_1, x_2, x_3) \rightsquigarrow \exists z \cdot A_{1,1}(x_1, x_2, x_3, z), I(\{x_1, x_2, x_3, z\}_{pub(x_2)})$$

$$\begin{aligned}&A_{2,0}(x_1, x_2, x_3), I(\{x_1, x_2, x_3, z\}_{pub(x_2)}) \rightsquigarrow\\ &\qquad \exists w \cdot A_{2,1}(x_1, x_2, x_3, z, w), I(\{z, w\}_{pub(x_1)}), I(\{x_1, x_2, x_3, z\}_{pub(x_2)})\end{aligned}$$

$$\begin{aligned}&A_{1,1}(x_1, x_2, x_3, z), I(\{z, w\}_{pub(x_1)}) \rightsquigarrow\\ &\qquad A_{1,2}(x_1, x_2, x_3, z, w), I(\{w\}_{pub(x_2)}), I(\{z, w\}_{pub(x_1)})\end{aligned}$$

$$A_{2,1}(x_1, x_2, x_3, z, w), I(\{w\}_{pub(x_2)}) \rightsquigarrow A_{2,2}(x_1, x_2, x_3, z, w), I(\{w\}_{pub(x_2)})$$

Security questions: can a protocol state be reached which contains

- $Ha(x_1), Ha(x_2), Ha(x_3), A_{1,2}(x_1, x_2, x_3, z, w), I(z)$.
- $Ha(x_1), Ha(x_2), Ha(x_3), A_{1,2}(x_1, x_2, x_3, z, w), I(w)$.
- $Ha(x_1), Ha(x_2), Ha(x_3), A_{2,2}(x_1, x_2, x_3, z, w), I(z)$.
- $Ha(x_1), Ha(x_2), Ha(x_3), A_{2,2}(x_1, x_2, x_3, z, w), I(w)$.

The first two represent the security questions about nonces N_{A_1} and N_{A_2} respectively from the point of view of A_1.

The last two represent the security questions about nonces N_{A_1} and N_{A_2} respectively from the point of view of A_2.

The standard man-in-the-middle attack.

We use honest agents A_1, A_2, A_3 and dishonest agent C ($k = 3$)

$$\begin{aligned}
A_1 \longrightarrow C : &\quad \{A_1, C, A_3, \ldots, A_k, N_{A_1}\}_{K_C} \\
C(A_1) \longrightarrow A_2 : &\quad \{A_1, A_2, A_3, \ldots, A_k, N_{A_1}\}_{K_{A_2}} \\
A_2 \longrightarrow A_1 : &\quad \{N_{A_1}, N_{A_2}\}_{K_{A_1}} \\
A_1 \longrightarrow C : &\quad \{N_{A_2}\}_{K_C} \\
C(A_1) \longrightarrow A_2 : &\quad \{N_{A_2}\}_{K_{A_2}}
\end{aligned}$$

The standard man-in-the-middle attack.

We use honest agents A_1, A_2, A_3 and dishonest agent C ($k = 3$)

$$
\begin{array}{ll}
A_1 \rightarrow C: & \{A_1, C, A_3, \ldots, A_k, N_{A_1}\}_{K_C} \\
C(A_1) \rightarrow A_2: & \{A_1, A_2, A_3, \ldots, A_k, N_{A_1}\}_{K_{A_2}} \\
A_2 \rightarrow A_1: & \{N_{A_1}, N_{A_2}\}_{K_{A_1}} \\
A_1 \rightarrow C: & \{N_{A_2}\}_{K_C} \\
C(A_1) \rightarrow A_2: & \{N_{A_2}\}_{K_{A_2}}
\end{array}
$$

Using our rules, we get a protocol state of the form

$Ha(a_1), Ha(a_2), Ha(a_3), Da(d),$
$A_{1,2}(a_1, d, a_3, n, m), A_{2,2}(a_1, a_2, a_3, n, m), I(n), I(m), \ldots$

Hence both security questions from the point of view of A_2 are violated.

Also, a protocol state containing $A_{2,2}(x_1, x_2, x_3, z, w)$ can be reached only if x_1, x_2, x_3 are mutually distinct.

The conditions $Ha(x_1), Ha(x_2), Ha(x_3)$ in the security property mean that these three agents should be honest.

Hence we require at least 3 honest agents for an attack.

In the absence of a dishonest agent, messages containing w known to the intruder always encrypted with public keys of honest agents.

Hence w can never be known to the intruder.

Hence an attack against the fourth security property requires at least 4 agents ($k + 1$ agents in general).

Sometimes certain special names can be used in protocol: e.g. servers.

These are not counted in the number of agents required for an attack.

$$\begin{aligned}
A \to B: &\quad A, N_a \\
B \to S: &\quad B, \{A, N_a, N_b\}_{K_{bs}} \\
S \to A: &\quad \{B, K_{ab}, N_a, N_b\}_{K_{as}}, \{A, K_{ab}\}_{K_{bs}} \\
A \to B: &\quad \{A, K_{ab}\}_{K_{bs}}, \{N_b\}_{K_{ab}}
\end{aligned}$$

This is the Yahalom protocol.

We use a special agent name $server$ and the rule

$$\rightsquigarrow Agent(server)$$

No rules of the form $Ha(server)$ or $Da(server)$.

No rules to state whether $server$ is distinct from other agents.

Protocol rules may involve these special names.

$$Agent(x), Agent(y), Distinct(x,y) \rightsquigarrow A_0(x,y,server), B_0(x,y,server), S_0(x,y,server), Agent(x), Agent(y), Distinct(x,y)$$

Security properties are of the form

$$Ha(x), Ha(y), A_2(x,y,server,z,u,v), I(v)$$

In this example we have $k = 2$ ($server$ is not counted).

An attack requires $k + 1 = 3$ agents besides the $server$.

Without the $Distinct$ predicates, an attack requires 2 agents besides the $server$.

- Two agents suffice for detecting attacks when agents involved in a session need not all be distinct.
- Otherwise $k+1$ agents suffice where k is the number of honest agents involved in the security property.
- The protocols must be independent of agent names.
- Security properties must be independent of agent names.
- Security properties must be reachability properties.
- Still this does not give us a method to check these security properties.

An example of protocol analysis 'by hand'

Our familiar ping-pong protocol

$$X \to Y\text{:} \quad \{M, X\}_{K_Y}$$

$$Y \to X\text{:} \quad \{M\}_{K_X}$$

We need to show that the protocol is secure.

For simplicity we work with the following rules for intruder's knowledge.

Intruder knows $E_B(i_A(M))$.

If intruder knows $E_Y(i_A(x))$ then intruder knows $E_X(x)$.

(Besides we have computation abilities of the intruder.)

For general protocols, we need to use multiset rewriting rules.

As usual we have two honest agents A, B and a dishonest agent C.

Idea: we look at the shape of messages that may be known to the intruder.

Messages involved are of the form $w \cdot M$ where w is a string of symbols $E_A, E_B, E_C, i_A, i_B, i_C$.

E.g. the message $E_B(i_A(M))$ is written as $E_B \cdot i_A \cdot M$.

Claim: every message known to the intruder is of one of the following two forms

1. $w \cdot E_B \cdot i_A \cdot M$ for some string w
2. $w \cdot E_A \cdot M$ for some string w

The first message $E_B \cdot i_A \cdot M$ known to the intruder is clearly of this form. (Here w is the empty string.)

Now we consider a protocol step.

The intruder already knows $E_Y \cdot i_X \cdot x$ using which he learns $E_X \cdot x$.

(1) Suppose $E_Y \cdot i_X \cdot x$ is of the form $w \cdot E_B \cdot i_A \cdot M$ for some string w.

Cases:

Now we consider a protocol step.

The intruder already knows $E_Y \cdot i_X \cdot x$ using which he learns $E_X \cdot x$.

(1) Suppose $E_Y \cdot i_X \cdot x$ is of the form $w \cdot E_B \cdot i_A \cdot M$ for some string w.

Cases:

- $|x| \geq 3$. Then x is of the form $w' \cdot E_B \cdot i_A \cdot M$ and $w = E_Y \cdot i_X \cdot w'$. Hence $E_X \cdot x$ is of the form $E_X \cdot w' \cdot E_B \cdot i_A \cdot M$.

Now we consider a protocol step.

The intruder already knows $E_Y \cdot i_X \cdot x$ using which he learns $E_X \cdot x$.

(1) Suppose $E_Y \cdot i_X \cdot x$ is of the form $w \cdot E_B \cdot i_A \cdot M$ for some string w.

Cases:

- $|x| \geq 3$. Then x is of the form $w' \cdot E_B \cdot i_A \cdot M$ and $w = E_Y \cdot i_X \cdot w'$. Hence $E_X \cdot x$ is of the form $E_X \cdot w' \cdot E_B \cdot i_A \cdot M$.
- $|x| = 2$. We must have $x = i_A \cdot M$ and $i_X = E_B$, which is impossible.

Now we consider a protocol step.

The intruder already knows $E_Y \cdot i_X \cdot x$ using which he learns $E_X \cdot x$.

(1) Suppose $E_Y \cdot i_X \cdot x$ is of the form $w \cdot E_B \cdot i_A \cdot M$ for some string w.

Cases:

- $|x| \geq 3$. Then x is of the form $w' \cdot E_B \cdot i_A \cdot M$ and $w = E_Y \cdot i_X \cdot w'$. Hence $E_X \cdot x$ is of the form $E_X \cdot w' \cdot E_B \cdot i_A \cdot M$.
- $|x| = 2$. We must have $x = i_A \cdot M$ and $i_X = E_B$, which is impossible.
- $|x| = 1$. We have $x = M$, $Y = B$ and $X = A$. The new message $E_X \cdot M = E_A \cdot M$ is of the required form.

Now we consider a protocol step.

The intruder already knows $E_Y \cdot i_X \cdot x$ using which he learns $E_X \cdot x$.

(1) Suppose $E_Y \cdot i_X \cdot x$ is of the form $w \cdot E_B \cdot i_A \cdot M$ for some string w.

Cases:

- $|x| \geq 3$. Then x is of the form $w' \cdot E_B \cdot i_A \cdot M$ and $w = E_Y \cdot i_X \cdot w'$. Hence $E_X \cdot x$ is of the form $E_X \cdot w' \cdot E_B \cdot i_A \cdot M$.
- $|x| = 2$. We must have $x = i_A \cdot M$ and $i_X = E_B$, which is impossible.
- $|x| = 1$. We have $x = M$, $Y = B$ and $X = A$. The new message $E_X \cdot M = E_A \cdot M$ is of the required form.
- $|x| = 0$. We must have $i_X = M$ which is impossible.

Now we consider a protocol step.

The intruder already knows $E_Y \cdot i_X \cdot x$ using which he learns $E_X \cdot x$.

(2) Suppose $E_Y \cdot i_X \cdot x$ is of the form $w \cdot E_A \cdot M$ for some string w.

Cases:

Now we consider a protocol step.

The intruder already knows $E_Y \cdot i_X \cdot x$ using which he learns $E_X \cdot x$.

(2) Suppose $E_Y \cdot i_X \cdot x$ is of the form $w \cdot E_A \cdot M$ for some string w.

Cases:

- $|x| \geq 2$. Then x is of the form $w' \cdot E_A \cdot M$ and $w = E_Y \cdot i_X \cdot w'$. Hence $E_X \cdot x$ is of the form $E_X \cdot w' \cdot E_A \cdot M$.

Now we consider a protocol step.

The intruder already knows $E_Y \cdot i_X \cdot x$ using which he learns $E_X \cdot x$.

(2) Suppose $E_Y \cdot i_X \cdot x$ is of the form $w \cdot E_A \cdot M$ for some string w.

Cases:

- $|x| \geq 2$. Then x is of the form $w' \cdot E_A \cdot M$ and $w = E_Y \cdot i_X \cdot w'$. Hence $E_X \cdot x$ is of the form $E_X \cdot w' \cdot E_A \cdot M$.
- $|x| = 1$. We must have $x = M$ and $i_X = E_A$, which is impossible.

Now we consider a protocol step.

The intruder already knows $E_Y \cdot i_X \cdot x$ using which he learns $E_X \cdot x$.

(2) Suppose $E_Y \cdot i_X \cdot x$ is of the form $w \cdot E_A \cdot M$ for some string w.

Cases:

- $|x| \geq 2$. Then x is of the form $w' \cdot E_A \cdot M$ and $w = E_Y \cdot i_X \cdot w'$. Hence $E_X \cdot x$ is of the form $E_X \cdot w' \cdot E_A \cdot M$.
- $|x| = 1$. We must have $x = M$ and $i_X = E_A$, which is impossible.
- $|x| = 0$. We must have $i_X = M$ which is impossible.

Finally we consider intruder computations.

The intruder knows a message w_1 of the form

1. $w \cdot E_B \cdot i_A \cdot M$ for some string w
2. or $w \cdot E_A \cdot M$ for some string w

Finally we consider intruder computations.

The intruder knows a message w_1 of the form

1. $w \cdot E_B \cdot i_A \cdot M$ for some string w
2. or $w \cdot E_A \cdot M$ for some string w

- If the intruder computes $E_X \cdot w_1$ or $i_X \cdot w_1$ (pushing a new symbol) then this new message is of the required form.

Finally we consider intruder computations.

The intruder knows a message w_1 of the form

1. $w \cdot E_B \cdot i_A \cdot M$ for some string w
2. or $w \cdot E_A \cdot M$ for some string w

- If the intruder computes $E_X \cdot w_1$ or $i_X \cdot w_1$ (pushing a new symbol) then this new message is of the required form.
- Now suppose the intruder pops a symbol i_X. This is possible only if $w = i_X \cdot w'$. Hence the new message is of the required form.

Finally we consider intruder computations.

The intruder knows a message w_1 of the form

1. $w \cdot E_B \cdot i_A \cdot M$ for some string w
2. or $w \cdot E_A \cdot M$ for some string w

- If the intruder computes $E_X \cdot w_1$ or $i_X \cdot w_1$ (pushing a new symbol) then this new message is of the required form.
- Now suppose the intruder pops a symbol i_X. This is possible only if $w = i_X \cdot w'$. Hence the new message is of the required form.
- Now suppose the intruder pops a symbol E_C. This is possible only if $w = E_C \cdot w'$. Hence the new message is of the required form.

Finally we consider intruder computations.

The intruder knows a message w_1 of the form

1. $w \cdot E_B \cdot i_A \cdot M$ for some string w
2. or $w \cdot E_A \cdot M$ for some string w

- If the intruder computes $E_X \cdot w_1$ or $i_X \cdot w_1$ (pushing a new symbol) then this new message is of the required form.
- Now suppose the intruder pops a symbol i_X. This is possible only if $w = i_X \cdot w'$. Hence the new message is of the required form.
- Now suppose the intruder pops a symbol E_C. This is possible only if $w = E_C \cdot w'$. Hence the new message is of the required form.

Hence the protocol is secure :-)

Some Key Distribution Protocols

Diffie-Hellman secret-key exchange protocol

Due to Diffie and Hellman (1976).

Two parties A and B have no symmetric or asymmetric keys, and want to agree on a common key to be used for symmetric encryption.

Fix a prime number p.

$$\mathbb{Z}_p^* = \{x \mid 0 < x < p, gcd(x, p) = 1\}$$

As p is prime, $\mathbb{Z}_p^* = \{1, \ldots, p-1\}$.

For every prime p there is some $g \in \mathbb{Z}_p^*$ such that

$$\mathbb{Z}_p^* = \{g^0 \bmod p, \ldots, g^{p-2} \bmod p\}$$

g is called the generator of $\mathbb{Z}_p^*$.

The protocol

The prime p and the generator g are known to everybody.

- A randomly chooses $0 \le N_a \le p-2$ and sends $X = g^{N_a} \mod p$ to B.
- B randomly chooses $0 \le N_b \le p-2$ and sends $Y = g^{N_b} \mod p$ to A.
- A computes Y^{N_a} as the secret key.
- B computes X^{N_b} as the secret key.

$$X^{N_b} = (g^{N_a})^{N_b} = g^{N_a N_b} = (g^{N_b})^{N_a} = Y^{N_a} \pmod p$$