Formal semantics

Let $fn(M)$ and $fn(P)$ be the set of free names in term M and process P respectively. Let $fv(M)$ and $fv(P)$ be the set of free variables in term M and process P respectively.

Closed processes are processes without any free variables.

Reaction relation:

A process is like a chemical solution of molecules waiting to react.

$$\overline{m}\langle N\rangle.P \mid m(x).Q \to P \mid Q[N/x]$$

Reduction relation $>$ on closed processes:

$$
\begin{aligned}
!P &> P \mid !P \\
[M \textit{ is } M]P &> P \\
\textit{let } (x, y) = (M, N) \textit{ in } P &> P[M/x][N/y] \\
\textit{case } 0 \textit{ of } 0 : P \ \textit{suc}(x) : Q &> P \\
\textit{case } \ \textit{suc}(M) \textit{ of } 0 : P \ \textit{suc}(x) : Q &> Q[M/x] \\
\textit{case } \{M\}_N \textit{ of } \{x\}_N \textit{ in } P &> P[M/x]
\end{aligned}
$$

Structural equivalence on closed processes:

$$\begin{aligned}
P \mid \mathbf{0} &\equiv P \\
P \mid Q &\equiv Q \mid P \\
P \mid (Q \mid R) &\equiv (P \mid Q) \mid R \\
(\nu m)(\nu n)P &\equiv (\nu n)(\nu m)P \\
(\nu n)\mathbf{0} &\equiv \mathbf{0} \\
(\nu n)(P \mid Q) &\equiv P \mid (\nu n)Q \text{ if } n \notin fn(P)
\end{aligned}$$

$$\frac{P > Q}{P \equiv Q}$$

$$\frac{}{P \equiv P} \qquad \frac{P \equiv Q}{Q \equiv P} \qquad \frac{P \equiv Q \quad Q \equiv R}{P \equiv R}$$

$$\frac{P \equiv P'}{P \mid Q \equiv P' \mid Q} \qquad \frac{P \equiv P'}{(\nu m)P \equiv (\nu m)P'}$$

The complete reaction rules:

$$\overline{m}\langle N\rangle.P \mid m(x).Q \rightarrow P \mid Q[N/x]$$

$$\frac{P \equiv P' \quad P' \rightarrow Q' \quad Q' \equiv Q}{P \rightarrow Q}$$

$$\frac{P \rightarrow P'}{P \mid Q \rightarrow P' \mid Q} \qquad \frac{P \rightarrow P'}{(\nu n)P \rightarrow (\nu n)P'}$$

$$A \longrightarrow S : \{K_{AB}\}_{K_{AS}} \text{ on } c_{AS}$$
$$S \longrightarrow B : \{K_{AB}\}_{K_{SB}} \text{ on } c_{SB}$$
$$A \longrightarrow B : \{M\}_{K_{AB}} \text{ on } c_{AB}$$

$$\begin{array}{rcl}
A(M) & \triangleq & (\nu K_{AB})(\overline{c_{AS}}\langle\{K_{AB}\}_{K_{AS}}\rangle \\
 & & .\overline{c_{AB}}\langle\{M\}_{K_{AB}}\rangle.\mathbf{0} \\
S & \triangleq & c_{AS}(x).case\ x\ of\ \{y\}_{K_{AS}}\ in\ \overline{c_{SB}}\langle\{y\}_{K_{SB}}\rangle.\mathbf{0} \\
B & \triangleq & c_{SB}(x).case\ x\ of\ \{y\}_{K_{SB}}\ in \\
 & & c_{AB}(z).case\ z\ of\ \{w\}_{y}\ in\ F(w) \\
Inst(M) & \triangleq & (\nu K_{AS})(\nu K_{SB})(A(M) \mid S \mid B)
\end{array}$$

$$
\begin{array}{rcl}
Inst(M) & \equiv & (\nu K_{AS})(\nu K_{SB})(A(M) \mid S \mid B) \\
& \longrightarrow & (\nu K_{AS})(\nu K_{SB})(\nu K_{AB}) \\
& & (\overline{c_{AB}}\langle\{M\}_{K_{AB}}\rangle.\mathbf{0} \mid \overline{c_{SB}}\langle\{K_{AB}\}_{K_{SB}}\rangle.\mathbf{0} \mid B) \\
& \longrightarrow & (\nu K_{AS})(\nu K_{SB})(\nu K_{AB}) \\
& & (\overline{c_{AB}}\langle\{M\}_{K_{AB}}\rangle.\mathbf{0} \mid \\
& & c_{AB}(z).case\ z\ of\ \{w\}_{K_{AB}}\ in\ F(w)) \\
& \longrightarrow & (\nu K_{AS})(\nu K_{SB})(\nu K_{AB})F(M)
\end{array}
$$

Testing equivalence

For this equivalence we are interested in the channels on which a process may communicate.

A barb is an element $\beta \in \{m, \overline{m}\}$ where m is a name.

We write $P \downarrow \beta$ to say that the closed process P can input or output immediately on the barb β. We say that P exhibits barb β.

$$m(x).P \downarrow m \qquad\qquad \overline{m}\langle M\rangle.P \downarrow \overline{m}$$

$$\frac{P \downarrow \beta}{P \mid Q \downarrow \beta} \qquad \frac{P \downarrow \beta \quad \beta \notin \{m, \overline{m}\}}{(\nu m)P \downarrow \beta} \qquad \frac{P \equiv Q \quad Q \downarrow \beta}{P \downarrow \beta}$$

We write $P \Downarrow \beta$ to say that P exhibits β after some reactions.

$$\frac{P \downarrow \beta}{P \Downarrow \beta} \qquad \frac{P \rightarrow Q \quad Q \Downarrow \beta}{P \Downarrow \beta}$$

a test is a closed process R and a barb β. A closed process P passes the test iff $(P \mid R) \Downarrow \beta$.

The testing equivalence is defined as:

$P \simeq Q \triangleq$ for any test (R, β), $(P \mid R) \Downarrow \beta$ iff $(Q \mid R) \Downarrow \beta$.

$$A \longrightarrow B : M \qquad \text{on } c_{AB}$$

$$\begin{aligned} A(M) &\triangleq \overline{c_{AB}}\langle M \rangle.\mathbf{0} \\ B &\triangleq c_{AB}(x).\mathbf{0} \\ Inst(M) &\triangleq (\nu c_{AB})(A(M) \mid B) \end{aligned}$$

Secrecy property: $Inst(M) \simeq Inst(M')$ for all M, M'.

i.e., for any process R and barb β,

$(Inst(M) \mid R) \Downarrow \beta$ iff $(Inst(M') \mid R) \Downarrow \beta$.

Actually the only barbs exhibited are those by the process R.

But if A and B communicate on unrestricted channels:

$$\begin{aligned} A(M) &\triangleq \overline{c_{AB}}\langle M\rangle.\mathbf{0} \\ B &\triangleq c_{AB}(x).\mathbf{0} \\ Inst(M) &\triangleq A(M) \mid B \end{aligned}$$

Let m be some message supposed to be secret. Then consider the process

$$R \triangleq c_{AB}(x).[x\ is\ m]\overline{d}\langle x\rangle.\mathbf{0}$$

We have $(Inst(m) \mid R) \Downarrow d$
but not $(Inst(M) \mid R) \Downarrow d$ for $m \neq M$.

$$
\begin{aligned}
A(M) &\triangleq \overline{c_{AB}}\langle M\rangle.\mathbf{0}\\
B &\triangleq c_{AB}(x).F(x)\\
Inst(M) &\triangleq (\nu c_{AB})(A(M) \mid B)
\end{aligned}
$$

For any process R, the barbs exhibited by the process $Inst(M) \mid R$ are exactly those exhibited by the process $F(M) \mid R$, hence we have the required secrecy property:

If $F(M) \simeq F(M')$ for all M, M'
then $Inst(M) \simeq Inst(M')$ for all M, M'

Authenticity:

$$\begin{aligned} A(M) &\triangleq \overline{c_{AB}}\langle M\rangle.\mathbf{0} \\ B_{spec}(M) &\triangleq c_{AB}(x).F(M) \\ Inst_{spec}(M) &\triangleq (\nu c_{AB})(A(M) \mid B_{spec}(M)) \end{aligned}$$

The barbs exhibited by the process $Inst_{spec}(M) \mid R$ are exactly those exhibited by the process $F(M) \mid R$, hence we have the required authenticity property:

$$Inst(M) \simeq Inst_{spec}(M) \text{ for all } M$$

In case of encryption:

$$P(M) \triangleq (\nu K)\overline{c}\langle\{M\}_K\rangle.\mathbf{0}$$

Secrecy is again preserved: $P(M) \simeq P(M')$ for all M, M'.

This is because the key K is restricted. No other process R may decrypt using this key. Hence whatever actions he may take using the message $\{M\}_K$ he may also take similar actions using the message $\{M'\}_K$.

Some approximation techniques for protocol analysis

The protocol security problem is undecidable.

Hence we do approximate analysis of protocols.

- Unsafe approximation: detect a few attacks, but not necessarily all. Useful while developing protocols.
  - bounding the number of sessions
  - bounding the size of messages
  - ...
- Safe approximation: detect all attacks. Sometimes false attacks may be detected. Useful for certifying protocols.

Some approximation techniques for protocol analysis

The protocol security problem is undecidable.

Hence we do approximate analysis of protocols.

- Unsafe approximation: detect a few attacks, but not necessarily all. Useful while developing protocols.
  - bounding the number of sessions
  - bounding the size of messages
  - ...
- Safe approximation: detect all attacks. Sometimes false attacks may be detected. Useful for certifying protocols.
  For the secrecy problem: over-approximate the intruder's knowledge

A common approximation is to let nonces be non-fresh in our modeling of the protocol.

Insecure protocol remains insecure after this abstraction.

Proof idea: take an attack in the multiset rewriting notation.
Show that systematically replacing a nonce by some other term produces a valid attack (except for the freshness condition on nonces).

Hence we choose a small number of nonces which are used repeatedly in several sessions.

Typical approximation: use only a finitely many nonces.

The public key Needham-Schroeder example:

$$\begin{array}{ll} 1. & A \longrightarrow B : \{A, N_a\}_{K_b} \\ 2. & B \longrightarrow A : \{N_a, N_b\}_{K_a} \\ 3. & A \longrightarrow B : \{N_b\}_{K_b} \end{array}$$

Following the results on reduction of number of agents, we first fix two honest agents A, B and one dishonest agent C.

Choose a finite set of nonces $n_{xy}^1, n_{xy}^2, n_{yx}^1, n_{yx}^2$ for distinct agents x and y.

Choose three keys K_a, K_b, K_c.

We have rules to define (an over-approximation of) the set of messages known to the intruder.

1. $A \longrightarrow B : \{A, N_a\}_{K_b}$
2. $B \longrightarrow A : \{N_a, N_b\}_{K_a}$
3. $A \longrightarrow B : \{N_b\}_{K_b}$

$$I(\{A, n_{ab}^1\}_{K_b})$$

$$I(\{A, n_{ac}^1\}_{K_c})$$

$$I(\{B, n_{ba}^1\}_{K_a})$$

$$I(\{B, n_{bc}^1\}_{K_c})$$

$$I(\{C, n_{ca}^1\}_{K_a})$$

$$I(\{C, n_{cb}^1\}_{K_b})$$

These can be written as push and pop rules.

The rule $I(\{A, n_{ab}^1\}_{K_b})$ is written as

$$
\begin{aligned}
&\longrightarrow q_1(A) \\
&\longrightarrow q_2(n_{ab}^1) \\
&\longrightarrow q_3(K_b) \\
q_1(x), q_2(y) &\longrightarrow q_4(\langle x, y \rangle) \\
q_4(x), q_3(y) &\longrightarrow I(\{x\}_y)
\end{aligned}
$$

for fresh states q_1, q_2, q_3, q_4.

The above are all push rules.

1. $A \longrightarrow B : \{A, N_a\}_{K_b}$

2. $B \longrightarrow A : \{N_a, N_b\}_{K_a}$

3. $A \longrightarrow B : \{N_b\}_{K_b}$

$$
\begin{aligned}
I(\{A, x\}_{K_b}) &\rightarrow I(\{x, n^2_{ab}\}_{K_a}) \\
I(\{A, x\}_{K_c}) &\rightarrow I(\{x, n^2_{ac}\}_{K_a}) \\
I(\{B, x\}_{K_a}) &\rightarrow I(\{x, n^2_{ba}\}_{K_b}) \\
I(\{B, x\}_{K_c}) &\rightarrow I(\{x, n^2_{bc}\}_{K_b}) \\
I(\{C, x\}_{K_a}) &\rightarrow I(\{x, n^2_{ca}\}_{K_c}) \\
I(\{C, x\}_{K_b}) &\rightarrow I(\{x, n^2_{cb}\}_{K_c})
\end{aligned}
$$

The rule $I(\{A, x\}_{K_b}) \longrightarrow I(\{x, n_{ab}^2\}_{K_a})$ can be written as:

$$
\begin{array}{rlr}
 & \longrightarrow p_1(K_b) & \text{(push)} \\
I(\{y\}_z), p_1(z) & \longrightarrow p_2(y) & \text{(pop)} \\
 & \longrightarrow p_3(A) & \text{(push)} \\
p_2(\langle y', x\rangle), p_3(y') & \longrightarrow p_4(x) & \text{(pop)} \\
 & \longrightarrow p_5(n_{ab}^2) & \text{(push)} \\
p_4(x), p_5(x') & \longrightarrow p_6(\langle x, x'\rangle) & \text{(pop)} \\
 & \longrightarrow p_7(K_a) & \text{(push)} \\
p_6(x''), p_7(x''') & \longrightarrow I(\{x''\}_{x'''}) & \text{(pop)}
\end{array}
$$

1. $A \longrightarrow B : \{A, N_a\}_{K_b}$

2. $B \longrightarrow A : \{N_a, N_b\}_{K_a}$

3. $A \longrightarrow B : \{N_b\}_{K_b}$

$$I(\{n^1_{ab}, x\}_{K_a}) \rightarrow I(\{x\}_{K_b})$$

$$I(\{n^1_{ac}, x\}_{K_a}) \rightarrow I(\{x\}_{K_c})$$

$$I(\{n^1_{ba}, x\}_{K_b}) \rightarrow I(\{x\}_{K_c})$$

$$I(\{n^1_{bc}, x\}_{K_b}) \rightarrow I(\{x\}_{K_a})$$

$$I(\{n^1_{ca}, x\}_{K_c}) \rightarrow I(\{x\}_{K_a})$$

$$I(\{n^1_{cb}, x\}_{K_c}) \rightarrow I(\{x\}_{K_b})$$

As usual we have the intruder's initial knowledge:

$I(n^1_{ca})$ $\quad$ $I(n^1_{cb})$ $\quad$ $I(n^2_{ac})$ $\quad$ $I(n^2_{bc})$ $\quad$ $I(K_a)$ $\quad$ $I(K_b)$

$I(K_c)$ $\quad$ $I(K_c^{-1})$ $\quad$ $I(A)$ $\quad$ $I(B)$ $\quad$ $I(C)$

The rules for intruder actions:

$$\begin{aligned} I(x), I(y) &\rightarrow I(\{x\}_y) \\ I(\{x\}_k), I(k^{-1}) &\rightarrow I(x) \\ &\ldots \end{aligned}$$

Example secrecy question : is nonce n^1_{ab} accepted at state I?

Security of this abstract protocol implies security of the real protocol.

Finally all protocols need not be translatable to push and pop rules.

E.g. rules like $q_1(f(x,x)), q_2(x) \longrightarrow q(x)$ are not pop rules.

Indeed the secrecy question is undecidable even for protocols without nonces.

Less severe abstractions are also possible.

E.g. the nonces may be a function not just of user names, but also of previous messages exchanged.

$$
\begin{array}{ll}
1. & A \longrightarrow B : \{A, N_a\}_{K_b} \\
2. & B \longrightarrow A : \{N_a, N_b\}_{K_a} \\
3. & A \longrightarrow B : \{N_b\}_{K_b}
\end{array}
$$

$$I(\{A, x\}_{K_b}) \quad \to I(\{x, n^2_{ab}(x)\}_{K_a})$$

$$\ldots$$

The second nonce $n^2_{ab}(x)$ depends on message x received.

Nonces may be non-fresh, but infinitely many of them may be used.

Extending the Dolev-Yao model with equations

Sometimes an accurate analysis of protocols requires considering special properties of underlying operations.

E.g. for the Diffie-Hellman key exchange we required special properties of modular exponentiation.

$$(g^x)^y = (g^y)^x$$

Other operations that are often used are e.g. XOR.

Sometimes there are attacks against protocols based on these special properties of the underlying operations.

Group key agreement protocols

Generalization of the Diffie-Hellman key exchange to several participants.

A

B

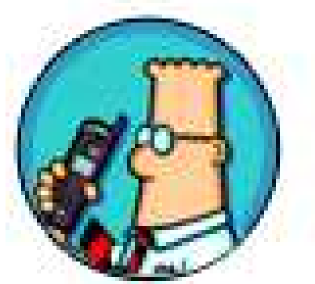
C

Group key agreement protocols

Generalization of the Diffie-Hellman key exchange to several participants.

A

$$\xrightarrow{\alpha^{Na}}$$

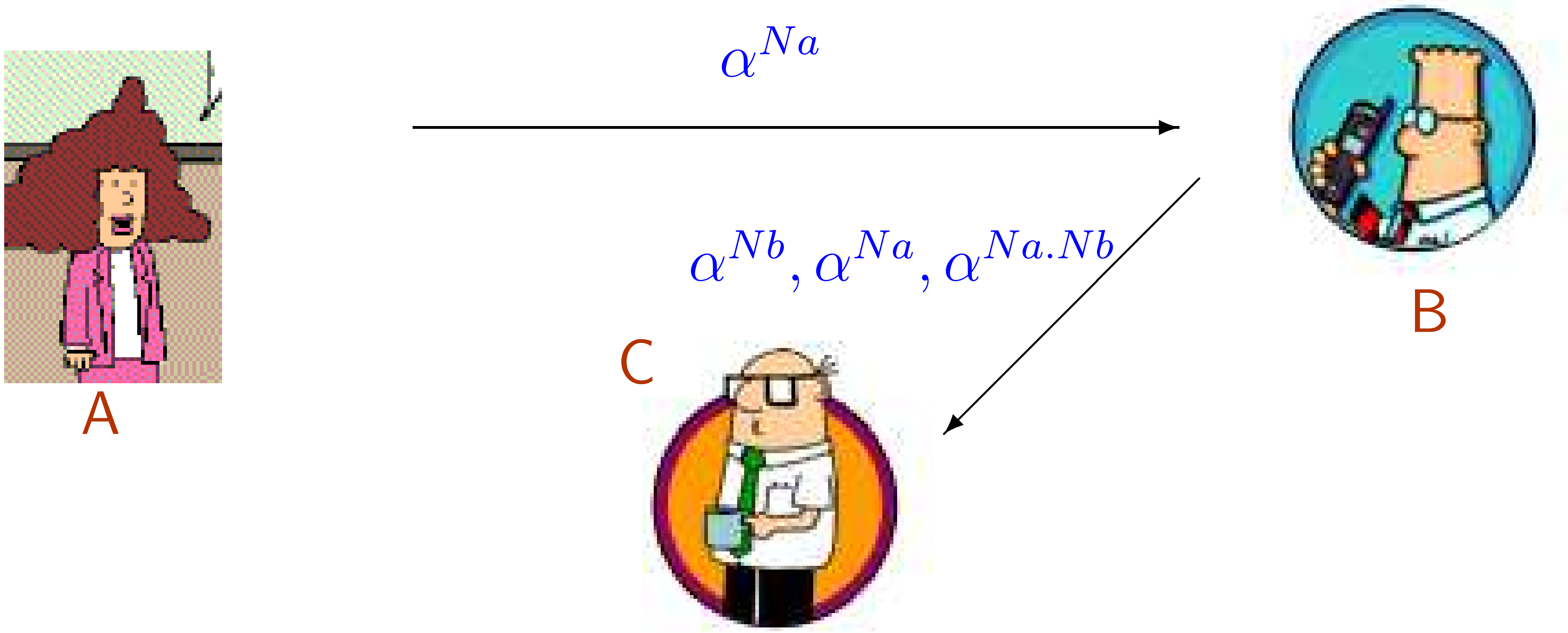


B

C

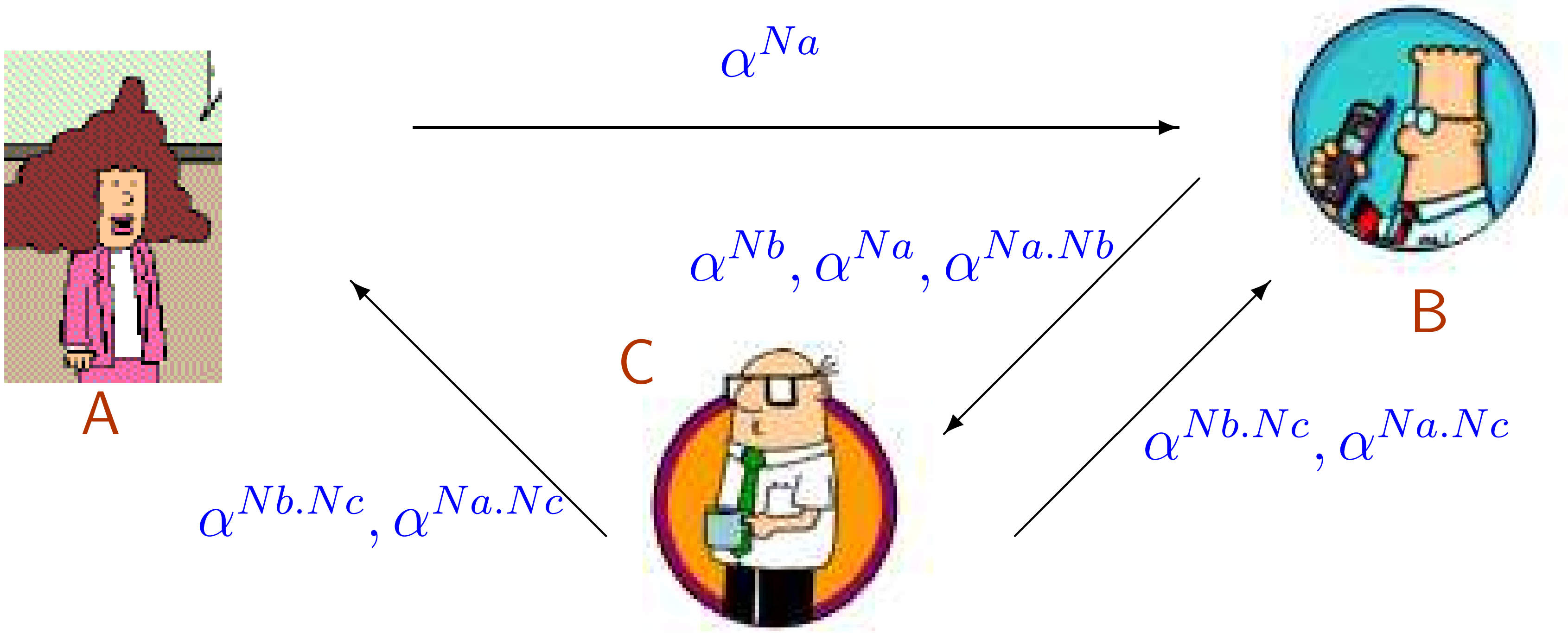
Group key agreement protocols

Generalization of the Diffie-Hellman key exchange to several participants.



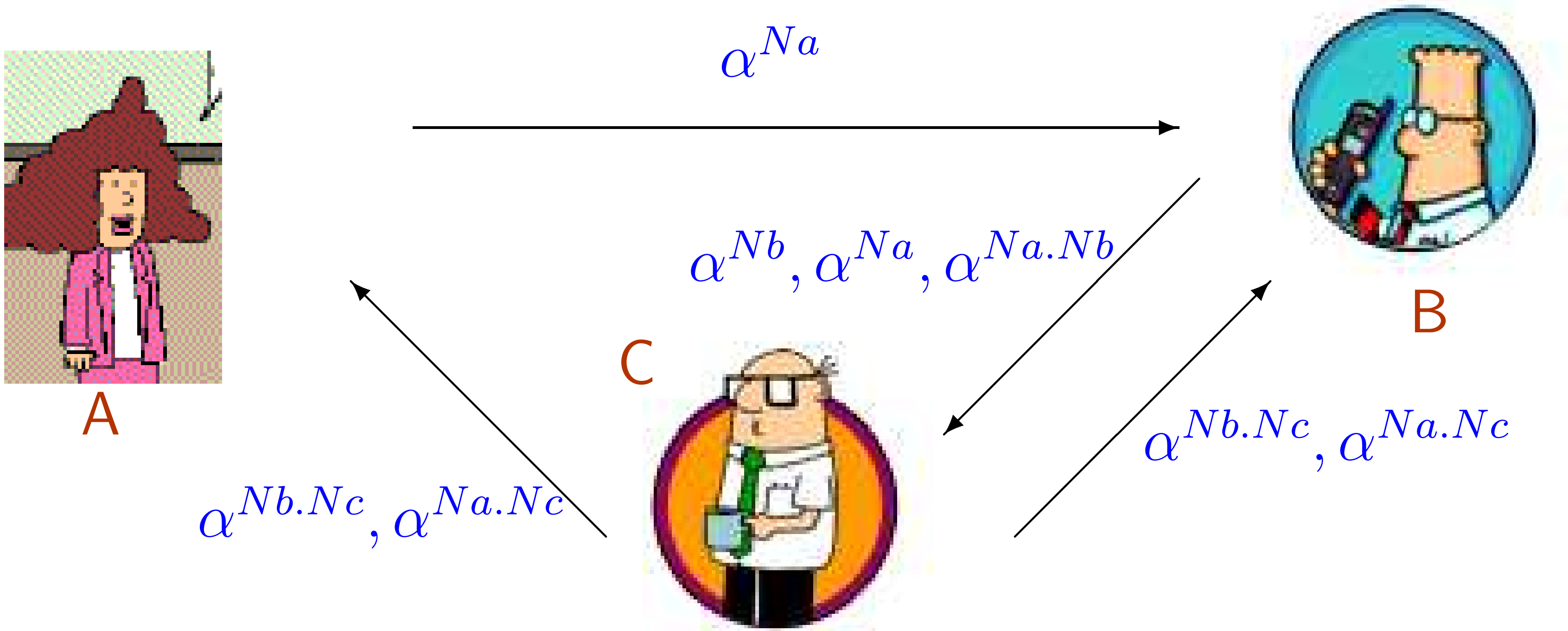
Group key agreement protocols

Generalization of the Diffie-Hellman key exchange to several participants.



Group key agreement protocols

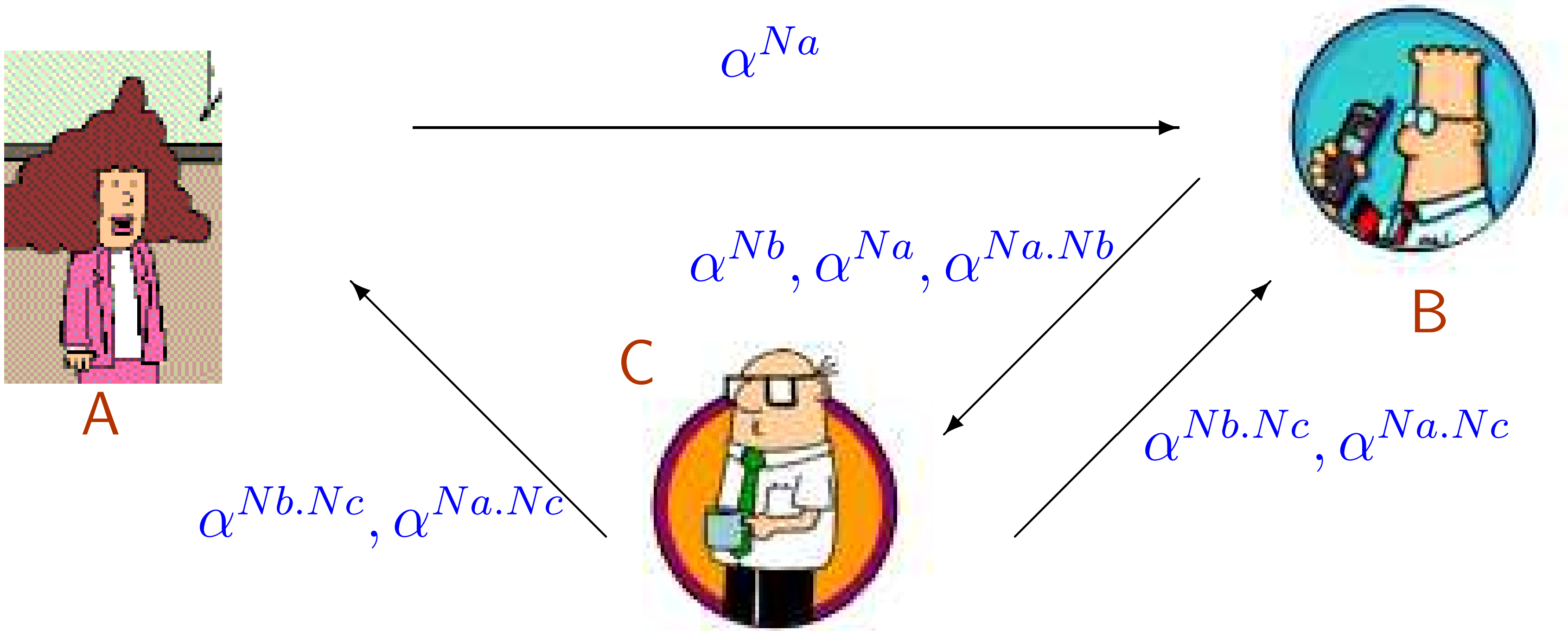
Generalization of the Diffie-Hellman key exchange to several participants.



Group key $\alpha^{Na.Nb.Nc}$ is then computed by each participant

Group key agreement protocols

Generalization of the Diffie-Hellman key exchange to several participants.



Group key $\alpha^{Na.Nb.Nc}$ is then computed by each participant

Code messages $\alpha^{x_1 \ldots x_n}$ by terms $e(x_1 + \ldots + x_n)$

$\Rightarrow +$ is ACU

The ACU theory

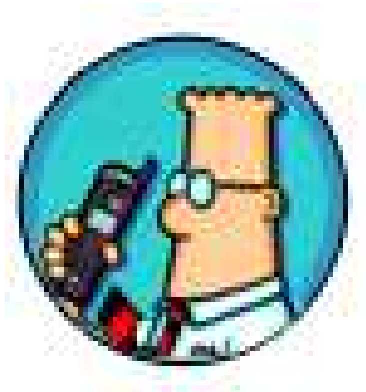
$x + (y + z) = (x + y) + z$	Associativity
$x + y = y + x$	Commutativity
$x + 0 = x$	Unit

A protocol using XOR

An example protocol using XOR:

$+$ is XOR

Alice



Bob

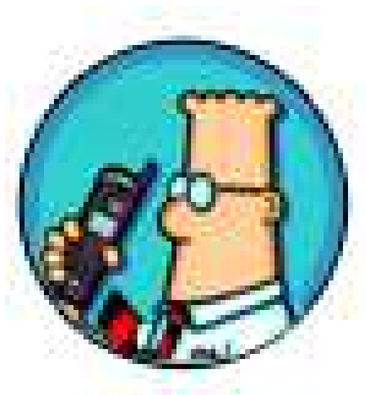
A protocol using XOR

An example protocol using XOR:

$+$ is XOR

Alice

$$\xrightarrow{\quad N_a + K_{ab} \quad}$$



Bob

A protocol using XOR

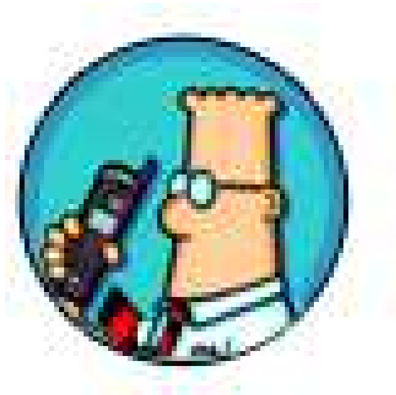
An example protocol using XOR:

$+$ is XOR

Alice

$$N_a + K_{ab} \quad (\text{Alice} \rightarrow \text{Bob})$$

$$N_b + N_a \quad (\text{Bob} \rightarrow \text{Alice})$$

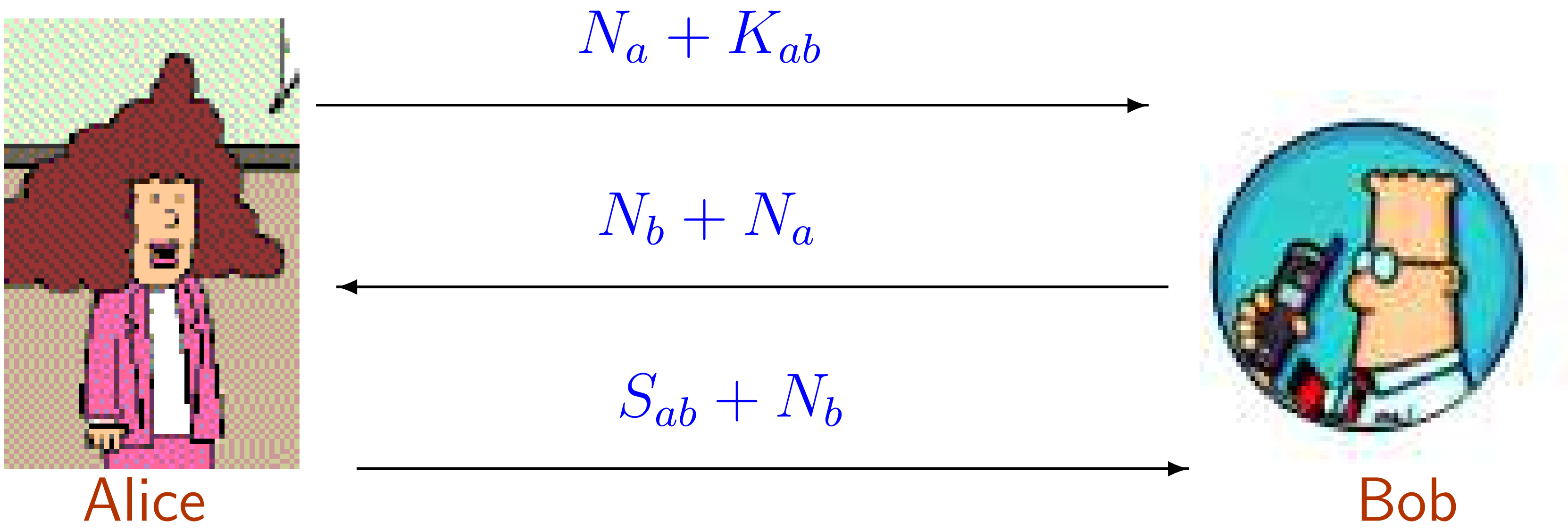


Bob

A protocol using XOR

An example protocol using XOR:

$+$ is XOR



A protocol using XOR

An example protocol using XOR:

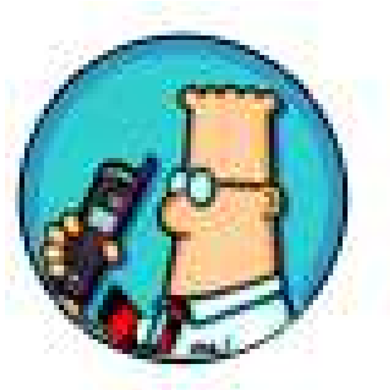
$+$ is XOR

Alice

$$N_a + K_{ab}$$

$$N_b + N_a$$

$$S_{ab} + N_b$$



Bob

Requires the XOR theory for modeling.

The XOR theory

The ACU theory, together with the equation

$$x + x = 0 \quad \text{Nilpotence}$$

Another example: the Abelian Groups theory

The ACU theory, together with the equation

$$x + (-x) = 0 \quad \text{Cancellation}$$

Typical equational theories that occur often in protocols are the ACU theory and its variants.