Protocoles cryptographiques et Vérification logique

Thomas Genet

IRISA

École des chercheurs de l'IRISA 2008

28-30 Janvier 2008

D'où vient cet exposé ?

- **Cours** de protocoles cryptographique de M1
- **Recherche** passée sur la vérification formelle de ces protocoles
- **Valorisation** industrielle des outils associés

Contexte

- Qu'est-ce qu'un protocole cryptographique ?
  - Protocole crypto= communication chiffrée entre 2 (ou +) agents

Contexte

- Qu'est-ce qu'un protocole cryptographique ?
  - Protocole crypto= communication chiffrée entre 2 (ou +) agents
  - Exemples : https, SSH, GSM, Paiement par carte bancaire

Contexte

- Qu'est-ce qu'un protocole cryptographique ?
  - Protocole crypto= communication chiffrée entre 2 (ou +) agents
  - Exemples : https, SSH, GSM, Paiement par carte bancaire
- Qu'entend-on par vérification formelle ?

Contexte

- Qu'est-ce qu'un protocole cryptographique ?
  - Protocole crypto= communication chiffrée entre 2 (ou +) agents
  - Exemples : https, SSH, GSM, Paiement par carte bancaire
- Qu'entend-on par vérification formelle ?
  - preuve (semi)-automatique de la validité d'un programme

Contexte

- Qu'est-ce qu'un protocole cryptographique ?
  - Protocole crypto= communication chiffrée entre 2 (ou +) agents
  - Exemples : https, SSH, GSM, Paiement par carte bancaire
- Qu'entend-on par vérification formelle ?
  - preuve (semi)-automatique de la validité d'un programme
- Pourquoi s'intéresser à leur vérification par des méthodes formelles ?
  - protocoles sont trop combinatoires pour une vérification manuelle !

Contexte

- Qu'est-ce qu'un protocole cryptographique ?
  - Protocole crypto= communication chiffrée entre 2 (ou +) agents
  - Exemples : https, SSH, GSM, Paiement par carte bancaire
- Qu'entend-on par vérification formelle ?
  - preuve (semi)-automatique de la validité d'un programme
- Pourquoi s'intéresser à leur vérification par des méthodes formelles ?
  - protocoles sont trop combinatoires pour une vérification manuelle !
  - 1995 Lowe : détection automatique de failles...

Contexte

- Qu'est-ce qu'un protocole cryptographique ?
  - Protocole crypto= communication chiffrée entre 2 (ou +) agents
  - Exemples : https, SSH, GSM, Paiement par carte bancaire
- Qu'entend-on par vérification formelle ?
  - preuve (semi)-automatique de la validité d'un programme
- Pourquoi s'intéresser à leur vérification par des méthodes formelles ?
  - protocoles sont trop combinatoires pour une vérification manuelle !
  - 1995 Lowe : détection automatique de failles...

...dans des protocoles vérifiés à la main.

Contexte

- Qu'est-ce qu'un protocole cryptographique ?
  - Protocole crypto= communication chiffrée entre 2 (ou +) agents
  - Exemples : https, SSH, GSM, Paiement par carte bancaire
- Qu'entend-on par vérification formelle ?
  - preuve (semi)-automatique de la validité d'un programme
- Pourquoi s'intéresser à leur vérification par des méthodes formelles ?
  - protocoles sont trop combinatoires pour une vérification manuelle !
  - 1995 Lowe : détection automatique de failles...

    ...dans des protocoles vérifiés à la main.
  - 1997 Bolignano, Paulson : preuves semi-automatiques

Plan

Plan

Vue symbolique de la cryptographie

Soient :

- m, m_1, m_2 des messages
- K une clé
- A et B des agents

Vue symbolique de la cryptographie

Soient :

- m, m_1, m_2 des messages
- K une clé
- A et B des agents

On note :

- m_1, m_2 le message constitué de m_1 et m_2

Vue symbolique de la cryptographie

Soient :

- m, m_1, m_2 des messages
- K une clé
- A et B des agents

On note :

- m_1, m_2 le message constitué de m_1 et m_2
- $\{m\}_K$ le message m crypté avec K

Vue symbolique de la cryptographie

Soient :

- m, m_1, m_2 des messages
- K une clé
- A et B des agents

On note :

- m_1, m_2 le message constitué de m_1 et m_2
- $\{m\}_K$ le message m crypté avec K
- $A \hookrightarrow B : m$ l'envoi par A d'un message m à B.

Vue symbolique de la cryptographie

Soient :

- m, m_1, m_2 des messages
- K une clé
- A et B des agents

On note :

- m_1, m_2 le message constitué de m_1 et m_2
- $\{m\}_K$ le message m crypté avec K
- $A \hookrightarrow B : m$ l'envoi par A d'un message m à B.
- $I(A) \hookrightarrow B : m$ l'envoi d'un message m par I se faisant passer pour A

Vue symbolique de la cryptographie

- Chiffrement à clé asymétrique (RSA, PGP, . . .) :

Vue symbolique de la cryptographie

- Chiffrement à clé asymétrique (RSA, PGP, . . .) :
  - la clé publique (K_A) donnée à tous les acteurs

Vue symbolique de la cryptographie

- Chiffrement à clé asymétrique (RSA, PGP, . . .) :
  - la clé publique (K_A) donnée à tous les acteurs
  - la clé privée (K_A^{-1}) connue de A seul

Vue symbolique de la cryptographie

- Chiffrement à clé asymétrique (RSA, PGP, ...) :
  - la clé publique (K_A) donnée à tous les acteurs
  - la clé privée (K_A^{-1}) connue de A seul
  - $\{\{m\}_{K_A}\}_{K_A^{-1}} = m = \{\{m\}_{K_A^{-1}}\}_{K_A}$

Vue symbolique de la cryptographie

- Chiffrement à clé asymétrique (RSA, PGP, . . .) :
  - la clé publique (K_A) donnée à tous les acteurs
  - la clé privée (K_A^{-1}) connue de A seul
  - $\{\{m\}_{K_A}\}_{K_A^{-1}} = m = \{\{m\}_{K_A^{-1}}\}_{K_A}$
  - inutilisable pour crypter de gros volumes de données

Vue symbolique de la cryptographie

- Chiffrement à clé asymétrique (RSA, PGP, . . .) :
  - la clé publique (K_A) donnée à tous les acteurs
  - la clé privée (K_A^{-1}) connue de A seul
  - $\{\{m\}_{K_A}\}_{K_A^{-1}} = m = \{\{m\}_{K_A^{-1}}\}_{K_A}$
  - inutilisable pour crypter de gros volumes de données
- Chiffrement à clé symétrique (Ex. DES) :

Vue symbolique de la cryptographie

- Chiffrement à clé asymétrique (RSA, PGP, ...) :
  - la clé publique (K_A) donnée à tous les acteurs
  - la clé privée (K_A^{-1}) connue de A seul
  - $\{\{m\}_{K_A}\}_{K_A^{-1}} = m = \{\{m\}_{K_A^{-1}}\}_{K_A}$
  - inutilisable pour crypter de gros volumes de données
- Chiffrement à clé symétrique (Ex. DES) :
  - $K_{AB} \equiv K_{AB}^{-1}$

Vue symbolique de la cryptographie

- Chiffrement à clé asymétrique (RSA, PGP, . . .) :
  - la clé publique (K_A) donnée à tous les acteurs
  - la clé privée (K_A^{-1}) connue de A seul
  - $\{\{m\}_{K_A}\}_{K_A^{-1}} = m = \{\{m\}_{K_A^{-1}}\}_{K_A}$
  - inutilisable pour crypter de gros volumes de données
- Chiffrement à clé symétrique (Ex. DES) :
  - $K_{AB} \equiv K_{AB}^{-1}$
  - bon rapport $\dfrac{\text{volume de données à crypter}}{\text{temps de chiffrement}}$

Vue symbolique de la cryptographie

- Fonctions de hachage (Ex. SHA, MD4, MD5, . . .)

Vue symbolique de la cryptographie

- Fonctions de hachage (Ex. SHA, MD4, MD5, . . .)
  - $| \mathit{hash}(d) | < | d |$

Vue symbolique de la cryptographie

- Fonctions de hachage (Ex. SHA, MD4, MD5, . . .)
  - $\mid hash(d) \mid < \mid d \mid$

et statistiquement :

  - $\forall d_1, d_2 : hash(d_1) \neq hash(d_2)$ si $d_1 \neq d_2$

Vue symbolique de la cryptographie

- Fonctions de hachage (Ex. SHA, MD4, MD5, . . .)
  - $\mid hash(d) \mid < \mid d \mid$

et statistiquement :

  - $\forall d_1, d_2 : hash(d_1) \neq hash(d_2)$ si $d_1 \neq d_2$
  - non inversible : $hash^{-1}$ n'existe pas

Propriétés associées à la cryptographie

- Confidentialité
  - $\{\text{“4976 0974 2373 7788”}\}_{K_B}$
  - $\{\texttt{recette\_teurgoule.ps}\}_{K_{AB}}$

Propriétés associées à la cryptographie

- Confidentialité
  - $\{$ “4976 0974 2373 7788” $\}_{K_B}$
  - $\{$`recette_teurgoule.ps`$\}_{K_{AB}}$
- Authentification (Signature électronique)
  - “genet@irisa.fr”, $\{$ “genet@irisa.fr” $\}_{K_{genet}^{-1}}$
  - `toto.gif`, $\{$ *hash*(`toto.gif`) $\}_{K_{genet}^{-1}}$

Propriétés associées à la cryptographie

- Confidentialité
  - $\{$ “4976 0974 2373 7788” $\}_{K_B}$
  - $\{$`recette_teurgoule.ps`$\}_{K_{AB}}$
- Authentification (Signature électronique)
  - “genet@irisa.fr”, $\{$ “genet@irisa.fr” $\}_{K_{genet}^{-1}}$
  - `toto.gif`, $\{$ *hash*(`toto.gif`) $\}_{K_{genet}^{-1}}$
- Intégrité
  - Le contenu m d'un message crypté $\{m\}_K$ ne peut être modifié sans K

Paiement par carte bancaire (vue externe)

Acteurs (ou agents) :

- (A)lice
- (C)arte bancaire (détenue par A)
- (T)erminal du commerçant
- (B)anque (banque émettrice de la carte)

Paiement par carte bancaire (vue externe)

Protocole de transaction :

- A introduit sa carte C dans T

Paiement par carte bancaire (vue externe)

Protocole de transaction :

- A introduit sa carte C dans T
- le commerçant saisit le montant m sur T

Paiement par carte bancaire (vue externe)

Protocole de transaction :

- A introduit sa carte C dans T
- le commerçant saisit le montant m sur T
- T authentifie C `''Authentification''`

Paiement par carte bancaire (vue externe)

Protocole de transaction :

- A introduit sa carte C dans T
- le commerçant saisit le montant m sur T
- T authentifie C `''Authentification''`
- A donne son code (3456) à C `''Code?''`

Paiement par carte bancaire (vue externe)

Protocole de transaction :

- A introduit sa carte C dans T
- le commerçant saisit le montant m sur T
- T authentifie C `''Authentification''`
- A donne son code (3456) à C `''Code?''`

Si m dépasse 100 euros (et dans 20% des cas)

Paiement par carte bancaire (vue externe)

Protocole de transaction :

- A introduit sa carte C dans T
- le commerçant saisit le montant m sur T
- T authentifie C `''Authentification''`
- A donne son code (3456) à C `''Code?''`

Si m dépasse 100 euros (et dans 20% des cas)

- T demande l'autorisation à B pour C
- B donne l'autorisation

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède un code secret : 3456

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède un code secret : 3456
- (B)anque possède
  - une clé publique : K_B (RSA)
  - une clé privée : K_B^{-1}

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède un code secret : 3456
- (B)anque possède
  - une clé publique : K_B (RSA)
  - une clé privée : K_B^{-1}
  - une clé symétrique partagée avec C : K_{CB} (DES)

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède un code secret : 3456
- (B)anque possède
  - une clé publique : K_B (RSA)
  - une clé privée : K_B^{-1}
  - une clé symétrique partagée avec C : K_{CB} (DES)
- (C)arte possède des informations **publiques**

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède un code secret : 3456
- (B)anque possède
  - une clé publique : K_B (RSA)
  - une clé privée : K_B^{-1}
  - une clé symétrique partagée avec C : K_{CB} (DES)
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède un code secret : 3456
- (B)anque possède
  - une clé publique : K_B (RSA)
  - une clé privée : K_B^{-1}
  - une clé symétrique partagée avec C : K_{CB} (DES)
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède un code secret : 3456
- (B)anque possède
  - une clé publique : K_B (RSA)
  - une clé privée : K_B^{-1}
  - une clé symétrique partagée avec C : K_{CB} (DES)
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$

  et la clé **secrète** K_{CB}

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède un code secret : 3456
- (B)anque possède
  - une clé publique : K_B (RSA)
  - une clé privée : K_B^{-1}
  - une clé symétrique partagée avec C : K_{CB} (DES)
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$

  et la clé **secrète** K_{CB}
- (T)erminal possède
  - $hash$
  - une clé publique K_B

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède son code : 3456
- (C)arte possède $Data$ et $\{hash(Data)\}_{K_B^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_B

Phase hors ligne de la transaction :

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède son code : 3456
- (C)arte possède $Data$ et $\{hash(Data)\}_{K_B^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_B

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T$: $Data, \quad \{hash(Data)\}_{K_B^{-1}}$

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède son code : 3456
- (C)arte possède $Data$ et $\{hash(Data)\}_{K_B^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_B

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

     $\overset{hash}{\downarrow}$

     $hash(Data)$

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède son code : 3456
- (C)arte possède $Data$ et $\{hash(Data)\}_{K_B^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_B

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T$: $Data$, $\{hash(Data)\}_{K_B^{-1}}$

$$\begin{array}{ccc} & \overset{hash}{\downarrow} & \overset{K_B}{\downarrow} \\ hash(Data) & = & hash(Data) \end{array}$$

Paiement par carte bancaire (vue interne, avant 2006)

- (A)lice possède son code : 3456
- (C)arte possède $Data$ et $\{hash(Data)\}_{K_B^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_B

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T$: $\underset{\overset{hash}{\downarrow}}{Data}, \quad \underset{\overset{K_B}{\downarrow}}{\{hash(Data)\}_{K_B^{-1}}}$
     $hash(Data) \quad = hash(Data)$
- A donne son code à C (C authentifie A)
  2. $T \hookrightarrow A$: code ?
  3. $A \hookrightarrow T$: 3456
  4. $T \hookrightarrow C$: 3456
  5. $C \hookrightarrow T$: ok

Paiement par carte bancaire (vue interne, avant 2006)

Si le montant est supérieur à 100 euros

- T demande l'autorisation à B pour C
  6. $T \hookrightarrow B$: Demande d'authentification

Paiement par carte bancaire (vue interne, avant 2006)

Si le montant est supérieur à 100 euros

- T demande l'autorisation à B pour C
  6. $T \hookrightarrow B$: Demande d'authentification
- B réalise l'authentification en ligne de C
  7. $B \hookrightarrow T : N_B$

Paiement par carte bancaire (vue interne, avant 2006)

Si le montant est supérieur à 100 euros

- T demande l'autorisation à B pour C
  6. $T \hookrightarrow B$: Demande d'authentification
- B réalise l'authentification en ligne de C
  7. $B \hookrightarrow T : N_B$
  8. $T \hookrightarrow C : N_B$

Paiement par carte bancaire (vue interne, avant 2006)

Si le montant est supérieur à 100 euros

- T demande l'autorisation à B pour C
  6. $T \hookrightarrow B$: Demande d'authentification
- B réalise l'authentification en ligne de C
  7. $B \hookrightarrow T : N_B$
  8. $T \hookrightarrow C : N_B$
  9. $C \hookrightarrow T : A, \{N_B\}_{K_{CB}}$

Paiement par carte bancaire (vue interne, avant 2006)

Si le montant est supérieur à 100 euros

- T demande l'autorisation à B pour C
  6. $T \hookrightarrow B$: Demande d'authentification
- B réalise l'authentification en ligne de C
  7. $B \hookrightarrow T : N_B$
  8. $T \hookrightarrow C : N_B$
  9. $C \hookrightarrow T : A, \{N_B\}_{K_{CB}}$
  10. $T \hookrightarrow B : A, \{N_B\}_{K_{CB}}$

Paiement par carte bancaire (vue interne, avant 2006)

Si le montant est supérieur à 100 euros

- T demande l'autorisation à B pour C
  6. $T \hookrightarrow B$: Demande d'authentification
- B réalise l'authentification en ligne de C
  7. $B \hookrightarrow T : N_B$
  8. $T \hookrightarrow C : N_B$
  9. $C \hookrightarrow T : A, \{N_B\}_{K_{CB}}$
  10. $T \hookrightarrow B : A, \{N_B\}_{K_{CB}}$

$$\{N_B\}_{K_{CB}} \downarrow N_B$$

Paiement par carte bancaire (vue interne, avant 2006)

Si le montant est supérieur à 100 euros

- T demande l'autorisation à B pour C
  6. $T \hookrightarrow B$: Demande d'authentification
- B réalise l'authentification en ligne de C
  7. $B \hookrightarrow T : \boxed{N_B}$
  8. $T \hookrightarrow C : N_B$
  9. $C \hookrightarrow T : A, \{N_B\}_{K_{CB}}$
  10. $T \hookrightarrow B : A, \{N_B\}_{K_{CB}}$

$$\{N_B\}_{K_{CB}} \downarrow$$

$$\boxed{N_B}$$

Paiement par carte bancaire (vue interne, avant 2006)

Si le montant est supérieur à 100 euros

- T demande l'autorisation à B pour C
  6. $T \hookrightarrow B$: Demande d'authentification
- B réalise l'authentification en ligne de C
  7. $B \hookrightarrow T : \boxed{N_B}$
  8. $T \hookrightarrow C : N_B$
  9. $C \hookrightarrow T : A, \{N_B\}_{K_{CB}}$
  10. $T \hookrightarrow B : A, \{N_B\}_{K_{CB}}$

$$\{N_B\}_{K_{CB}} \downarrow$$

$$\boxed{N_B}$$

- B donne l'autorisation
  10. $B \hookrightarrow T$: ok

Quelques faiblesses de la carte bancaire

Initialement la sécurité de la carte reposait beaucoup sur :

- la non réplicabilité de la carte
- le secret autour des clés employées, du protocole, ...

Quelques faiblesses de la carte bancaire

Initialement la sécurité de la carte reposait beaucoup sur :

- la non réplicabilité de la carte
- le secret autour des clés employées, du protocole, ...

Mais

Quelques faiblesses de la carte bancaire

Initialement la sécurité de la carte reposait beaucoup sur :

- la non réplicabilité de la carte
- le secret autour des clés employées, du protocole, ...

Mais

- Faiblesse cryptographique :
  clés RSA 320 bits ne sont plus sûres (1988)

Quelques faiblesses de la carte bancaire

Initialement la sécurité de la carte reposait beaucoup sur :

- la non réplicabilité de la carte
- le secret autour des clés employées, du protocole, ...

Mais

- Faiblesse cryptographique :
  clés RSA 320 bits ne sont plus sûres (1988)
- Faiblesse logique du protocole :
  pas de lien "code à 4 chiffres" et authentification !

Quelques faiblesses de la carte bancaire

Initialement la sécurité de la carte reposait beaucoup sur :

- la non réplicabilité de la carte
- le secret autour des clés employées, du protocole, ...

Mais

- Faiblesse cryptographique :
  clés RSA 320 bits ne sont plus sûres (1988)
- Faiblesse logique du protocole :
  pas de lien "code à 4 chiffres" et authentification !
- Faiblesse physique :
  réplicabilité $\Rightarrow$ Yescard ! (Humpich 1998)

Quelques faiblesses (suite)

Faiblesse logique du protocole :

1. $C \hookrightarrow T : Data, \{hash(Data)\}_{K_B^{-1}}$
2. $T \hookrightarrow A$: code ?
3. $A \hookrightarrow C$: 3456
4. $C \hookrightarrow T$: ok

Quelques faiblesses (suite)

Faiblesse logique du protocole :

1. $C \hookrightarrow T : Data, \{hash(Data)\}_{K_B^{-1}}$
2. $T \hookrightarrow A$: code ?
3. $A \hookrightarrow C$: 3456
4. $C \hookrightarrow T$: ok

$A \hookrightarrow C' : 7575$

$C' \hookrightarrow T$: ok

Quelques faiblesses (suite)

Faiblesse logique du protocole :

1. $C \hookrightarrow T : Data, \{hash(Data)\}_{K_B^{-1}}$
2. $T \hookrightarrow A$: code ?
3. $A \hookrightarrow C'$: 7575
4. $C' \hookrightarrow T$: ok

Quelques faiblesses (suite)

Faiblesse logique du protocole :

1. $C \hookrightarrow T : Data, \{hash(Data)\}_{K_B^{-1}}$
2. $T \hookrightarrow A$: code ?
3. $A \hookrightarrow C'$: 7575
4. $C' \hookrightarrow T$: ok

Implantée dans une Yescard :

1. $C \hookrightarrow T : Data, \{hash(Data)\}_{K_B^{-1}}$
2. $T \hookrightarrow A$: code ?
3. $A \hookrightarrow C$: 0000
4. $C \hookrightarrow T$: ok

Quelques faiblesses (suite)

Faiblesse logique du protocole :

1. $C \hookrightarrow T : Data, \{hash(Data)\}_{K_B^{-1}}$
2. $T \hookrightarrow A$: code ?
3. $A \hookrightarrow C' : 7575$
4. $C' \hookrightarrow T$: ok

Implantée dans une Yescard (et fausse signature RSA)

1. $C \hookrightarrow T : ZZZ, \{hash(ZZZ)\}_{K_B^{-1}}$
2. $T \hookrightarrow A$: code ?
3. $A \hookrightarrow C$: 0000
4. $C \hookrightarrow T$: ok

Corrections appliquées par EMVCo (2004)

Europay, MasterCard et Visa ont produit *EMV* pour les cartes bancaires :

- non pas un mais 3 protocoles : SDA, DDA, CDA

Corrections appliquées par EMVCo (2004)

Europay, MasterCard et Visa ont produit *EMV* pour les cartes bancaires :

- non pas un mais 3 protocoles : SDA, DDA, CDA
- Conçu en 2004 et déployé depuis environ 2006

Corrections appliquées par EMVCo (2004)

Europay, MasterCard et Visa ont produit *EMV* pour les cartes bancaires :

- non pas un mais 3 protocoles : SDA, DDA, CDA
- Conçu en 2004 et déployé depuis environ 2006
- spécifications disponibles sur le web !

Corrections appliquées par EMVCo (2004)

Europay, MasterCard et Visa ont produit *EMV* pour les cartes bancaires :

- non pas un mais 3 protocoles : SDA, DDA, CDA
- Conçu en 2004 et déployé depuis environ 2006
- spécifications disponibles sur le web !
- multi-applications

Paiement par carte bancaire (vue interne, SDA)

SDA=Static Data Authentication

- (A)lice possède un code secret : 3456

Paiement par carte bancaire (vue interne, SDA)

SDA=Static Data Authentication

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **publiques**

Paiement par carte bancaire (vue interne, SDA)

SDA=Static Data Authentication

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$

Paiement par carte bancaire (vue interne, SDA)

SDA=Static Data Authentication

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$
  - Certificat $\{K_B\}_{K_S^{-1}}$ de la clé de la banque

Paiement par carte bancaire (vue interne, SDA)

SDA=Static Data Authentication

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$
  - Certificat $\{K_B\}_{K_S^{-1}}$ de la clé de la banque
- (T)erminal possède
  - *hash*
  - une clé publique K_S

Paiement par carte bancaire (vue interne, SDA)

- (A)lice possède son code : 3456
- (C)arte possède $Data$, $\{hash(Data)\}_{K_B^{-1}}$, $\{K_B\}_{K_S^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_S

Phase hors ligne de la transaction :

Paiement par carte bancaire (vue interne, SDA)

- (A)lice possède son code : 3456
- (C)arte possède $Data$, $\{hash(Data)\}_{K_B^{-1}}$, $\{K_B\}_{K_S^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_S

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

Paiement par carte bancaire (vue interne, SDA)

- (A)lice possède son code : 3456
- (C)arte possède $Data$, $\{hash(Data)\}_{K_B^{-1}}$, $\{K_B\}_{K_S^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_S

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

     $K_S \downarrow$

     $K_B,$

Paiement par carte bancaire (vue interne, SDA)

- (A)lice possède son code : 3456
- (C)arte possède $Data$, $\{hash(Data)\}_{K_B^{-1}}$, $\{K_B\}_{K_S^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_S

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

     $K_S \downarrow$ $\qquad$ $hash \downarrow$

     $K_B, \quad hash(Data)$

Paiement par carte bancaire (vue interne, SDA)

- (A)lice possède son code : 3456
- (C)arte possède $Data$, $\{hash(Data)\}_{K_B^{-1}}$, $\{K_B\}_{K_S^{-1}}$
- (T)erminal possède $hash$ et la clé publique K_S

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \underset{K_S \downarrow}{\{K_B\}_{K_S^{-1}}}, \quad \underset{hash \downarrow}{Data}, \quad \underset{K_B \downarrow}{\{hash(Data)\}_{K_B^{-1}}}$

     $K_B, \quad hash(Data) \quad = hash(Data)$

Paiement par carte bancaire (vue interne, SDA)

- (A)lice possède son code : 3456
- (C)arte possède *Data*, $\{hash(Data)\}_{K_B^{-1}}$, $\{K_B\}_{K_S^{-1}}$
- (T)erminal possède *hash* et la clé publique K_S

Phase hors ligne de la transaction :

- T authentifie C

$$1.\quad C \hookrightarrow T : \underset{K_B,}{\{K_B\}_{K_S^{-1}} \ {\scriptstyle K_S}\!\downarrow},\quad \underset{hash(Data)}{Data, \ {\scriptstyle hash}\!\downarrow}\quad \underset{=\ hash(Data)}{\{hash(Data)\}_{K_B^{-1}} \ {\scriptstyle K_B}\!\downarrow}$$

- A donne son code à C (C authentifie A)

2. $T \hookrightarrow A$: code ?
3. $A \hookrightarrow C$: 3456
4. $C \hookrightarrow T$: ok

Paiement par carte bancaire (vue interne, DDA)

DDA=Dynamic Data Authentication (Déployé depuis mai 2007)

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **privées**
  - Clé K_C^{-1} propre à la (C)arte

Paiement par carte bancaire (vue interne, DDA)

DDA=Dynamic Data Authentication (Déployé depuis mai 2007)

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **privées**
  - Clé K_C^{-1} propre à la (C)arte
- (C)arte possède des informations **publiques**

Paiement par carte bancaire (vue interne, DDA)

DDA=Dynamic Data Authentication (Déployé depuis mai 2007)

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **privées**
  - Clé K_C^{-1} propre à la (C)arte
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$

Paiement par carte bancaire (vue interne, DDA)

DDA=Dynamic Data Authentication (Déployé depuis mai 2007)

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **privées**
  - Clé K_C^{-1} propre à la (C)arte
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$
  - Certificat $\{K_B\}_{K_S^{-1}}$ de la clé de la (B)anque

Paiement par carte bancaire (vue interne, DDA)

DDA=Dynamic Data Authentication (Déployé depuis mai 2007)

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **privées**
  - Clé K_C^{-1} propre à la (C)arte
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$
  - Certificat $\{K_B\}_{K_S^{-1}}$ de la clé de la (B)anque
  - Certificat $\{K_C\}_{K_B^{-1}}$ de la clé de la (C)arte

Paiement par carte bancaire (vue interne, DDA)

DDA=Dynamic Data Authentication (Déployé depuis mai 2007)

- (A)lice possède un code secret : 3456
- (C)arte possède des informations **privées**
  - Clé K_C^{-1} propre à la (C)arte
- (C)arte possède des informations **publiques**
  - $Data$= nom, prénom, numéro carte, date de validité
  - Valeur de Signature $VS = \{hash(Data)\}_{K_B^{-1}}$
  - Certificat $\{K_B\}_{K_S^{-1}}$ de la clé de la (B)anque
  - Certificat $\{K_C\}_{K_B^{-1}}$ de la clé de la (C)arte
- (T)erminal possède
  - *hash*
  - une clé publique K_S

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \{K_C\}_{K_B^{-1}} \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \{K_C\}_{K_B^{-1}} \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

$K_S \downarrow$

$K_B,$

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \underset{K_S \downarrow}{\{K_B\}_{K_S^{-1}}}, \underset{K_B \downarrow}{\{K_C\}_{K_B^{-1}}} \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

     $K_B, \qquad K_C,$

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \{K_C\}_{K_B^{-1}} \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

     $K_S \downarrow \qquad K_B \downarrow \qquad hash \downarrow$

     $K_B, \qquad K_C, \quad hash(Data)$

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \underset{K_S \downarrow}{\{K_B\}_{K_S^{-1}}}, \underset{K_B \downarrow}{\{K_C\}_{K_B^{-1}}} \quad \underset{hash \downarrow}{Data}, \quad \underset{K_B \downarrow}{\{hash(Data)\}_{K_B^{-1}}}$

     $K_B, \quad K_C, \quad hash(Data) \quad = hash(Data)$

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \{K_C\}_{K_B^{-1}} \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

     $\downarrow_{K_S} \quad \downarrow_{K_B} \quad \downarrow_{hash} \quad \downarrow_{K_B}$

     $K_B, \quad K_C, \quad hash(Data) \quad = hash(Data)$

  2. $T \hookrightarrow C : N_T$

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \{K_C\}_{K_B^{-1}} \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

     $K_S \downarrow \quad K_B \downarrow \quad hash \downarrow \quad K_B \downarrow$

     $K_B, \quad K_C, \quad hash(Data) \quad = hash(Data)$

  2. $T \hookrightarrow C : N_T$
  3. $C \hookrightarrow T : \{N_T\}_{K_C^{-1}}$

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \{K_C\}_{K_B^{-1}} \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

     $K_S \downarrow \quad K_B \downarrow \quad hash \downarrow \quad K_B \downarrow$

     $K_B, \quad K_C, \quad hash(Data) \quad = hash(Data)$

  2. $T \hookrightarrow C : N_T$
  3. $C \hookrightarrow T : \{N_T\}_{K_C^{-1}}$

     $K_C \downarrow$

     N_T

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \underset{K_S \downarrow}{\{K_B\}_{K_S^{-1}}}, \underset{K_B \downarrow}{\{K_C\}_{K_B^{-1}}} \quad \underset{hash \downarrow}{Data}, \quad \underset{K_B \downarrow}{\{hash(Data)\}_{K_B^{-1}}}$

     $K_B, \qquad K_C, \quad hash(Data) \quad = hash(Data)$

  2. $T \hookrightarrow C : \boxed{N_T}$
  3. $C \hookrightarrow T : \underset{K_C \downarrow}{\{N_T\}_{K_C^{-1}}}$

     $\boxed{N_T}$

Paiement par carte bancaire (vue interne, CDA)

Phase hors ligne de la transaction :

- T authentifie C
  1. $C \hookrightarrow T : \{K_B\}_{K_S^{-1}}, \{K_C\}_{K_B^{-1}} \quad Data, \quad \{hash(Data)\}_{K_B^{-1}}$

     $\downarrow_{K_S} \quad \downarrow_{K_B} \quad \downarrow_{hash} \quad \downarrow_{K_B}$

     $K_B, \quad K_C, \quad hash(Data) \quad = hash(Data)$

  2. $T \hookrightarrow C : \boxed{N_T}$
  3. $C \hookrightarrow T : \{N_T\}_{K_C^{-1}}$

     $\downarrow_{K_C}$

     $\boxed{N_T}$

- A donne son code à C (C authentifie A)
  4. $T \hookrightarrow A$: code ?
  5. $A \hookrightarrow T$: 3456
  6. $T \hookrightarrow C : \{3456\}_{K_C}$
  7. $C \hookrightarrow T$: ok

Plan

Pourquoi vérifier les protocoles cryptographiques ?

Beaucoup de protocoles de la littérature ont des failles !

Pourquoi vérifier les protocoles cryptographiques ?

Beaucoup de protocoles de la littérature ont des failles !

Il s'agit de failles logiques :

- liées à l'enchaînement des messages
- pas liées à l'implantation
- ne nécessitent pas de cryptanalyse

Pourquoi vérifier les protocoles cryptographiques ?

Beaucoup de protocoles de la littérature ont des failles !

Il s'agit de failles logiques :

- liées à l'enchaînement des messages
- pas liées à l'implantation
- ne nécessitent pas de cryptanalyse

protocoles crypto.= cas d'étude idéal pour les méthodes formelles

- relativement abstraits
- de taille réduite
- vérification impraticable à la main

Pourquoi vérifier les protocoles cryptographiques ?

Beaucoup de protocoles de la littérature ont des failles !

Il s'agit de failles logiques :

- liées à l'enchaînement des messages
- pas liées à l'implantation
- ne nécessitent pas de cryptanalyse

protocoles crypto.= cas d'étude idéal pour les méthodes formelles

- relativement abstraits
- de taille réduite
- vérification impraticable à la main

Il faut formaliser les capacités de l'intrus= modèle de Dolev-Yao (1983)

Les capacités de l'intrus – Modèle de Dolev Yao

① L'intrus ne fait pas de cryptanalyse $\Rightarrow$ suppose les clés incassables

Les capacités de l'intrus – Modèle de Dolev Yao

① L'intrus ne fait pas de cryptanalyse ⇒ suppose les clés incassables

Hypothèse *raisonnable* en pratique :

- utiliser des clés de longueurs suffisantes
- suivre les résultats des compétitions de factorisation de clés

Les capacités de l'intrus – Modèle de Dolev Yao

① L'intrus ne fait pas de cryptanalyse ⇒ suppose les clés incassables

Hypothèse *raisonnable* en pratique :

- utiliser des clés de longueurs suffisantes
- suivre les résultats des compétitions de factorisation de clés

---

Exemple : RSA record à 663 bits (2005)

Carte bancaires *doivent être* à RSA 1024 bits...

Compétition de factorisation RSA

RSA-155	155	512		22 août 1999	Herman te Riele *et al.*
RSA-160	160	530		1er avril 2003	Jens Franke *et al.*, Université de Bonn
RSA-170	170	563			*ouvert*
RSA-576	174	576	10 000 $ USD	3 décembre, 2003	Jens Franke *et al.*, Université de Bonn
RSA-180	180	596			*ouvert*
RSA-190	190	629			*ouvert*
RSA-640	193	640	20 000 $ USD	2 novembre 2005	Jens Franke *et al.*, Université de Bonn
RSA-200	200	663		9 mai 2005	Jens Franke *et al.*, Université de Bonn
RSA-210	210	696			*ouvert*
RSA-704	212	704	30 000 $ USD		*ouvert*

Les capacités de l'intrus – Modèle de Dolev-Yao

Version simplifiée de SSH en mode dégradé et basé sur Diffie-Hellman

1. $A \hookrightarrow B : g^{N_A}$

Les capacités de l'intrus – Modèle de Dolev-Yao

Version simplifiée de SSH en mode dégradé et basé sur Diffie-Hellman

1. $A \hookrightarrow B : g^{N_A}$
2. $B \hookrightarrow A : g^{N_B}$

$$K = (g^{N_A})^{N_B} = (g^{N_B})^{N_A} = g^{N_A.N_B}$$

Les capacités de l'intrus – Modèle de Dolev-Yao

Version simplifiée de SSH en mode dégradé et basé sur Diffie-Hellman

1. $A \hookrightarrow B : g^{N_A}$
2. $B \hookrightarrow A : g^{N_B}$
3. $B \hookrightarrow A : \{login :\}_K$

$$K = (g^{N_A})^{N_B} = (g^{N_B})^{N_A} = g^{N_A.N_B}$$

Les capacités de l'intrus – Modèle de Dolev-Yao

Version simplifiée de SSH en mode dégradé et basé sur Diffie-Hellman

1. $A \hookrightarrow B : g^{N_A}$
2. $B \hookrightarrow A : g^{N_B}$
3. $B \hookrightarrow A : \{login :\}_K$
4. $A \hookrightarrow B : \{A\}_K$

$$K = (g^{N_A})^{N_B} = (g^{N_B})^{N_A} = g^{N_A.N_B}$$

Les capacités de l'intrus – Modèle de Dolev-Yao

Version simplifiée de SSH en mode dégradé et basé sur Diffie-Hellman

1. $A \hookrightarrow B : g^{N_A}$
2. $B \hookrightarrow A : g^{N_B}$
3. $B \hookrightarrow A : \{login :\}_K$
4. $A \hookrightarrow B : \{A\}_K$
5. $B \hookrightarrow A : \{passwd :\}_K$

$$K = (g^{N_A})^{N_B} = (g^{N_B})^{N_A} = g^{N_A.N_B}$$

Les capacités de l'intrus – Modèle de Dolev-Yao

Version simplifiée de SSH en mode dégradé et basé sur Diffie-Hellman

1. $A \hookrightarrow B : g^{N_A}$
2. $B \hookrightarrow A : g^{N_B}$
3. $B \hookrightarrow A : \{login :\}_K$
4. $A \hookrightarrow B : \{A\}_K$
5. $B \hookrightarrow A : \{passwd :\}_K$
6. $A \hookrightarrow B : \{P\}_K$

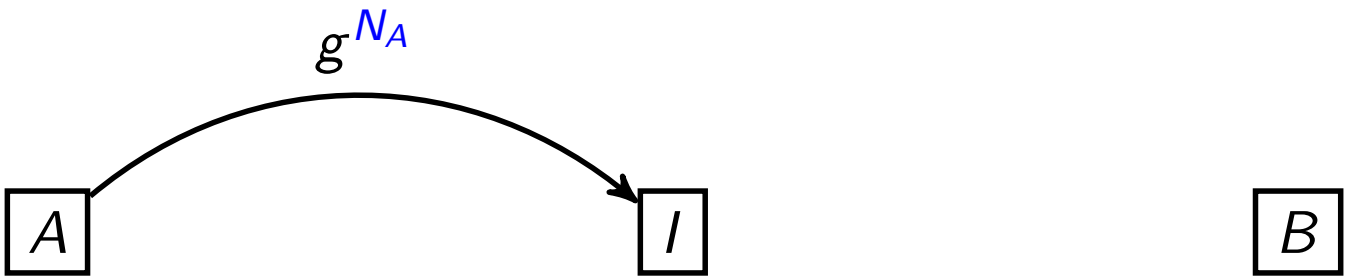
$$K = (g^{N_A})^{N_B} = (g^{N_B})^{N_A} = g^{N_A.N_B}$$

Les capacités de l'intrus – Modèle de Dolev-Yao

Attaque de type "Man in the middle" :

Les capacités de l'intrus – Modèle de Dolev-Yao

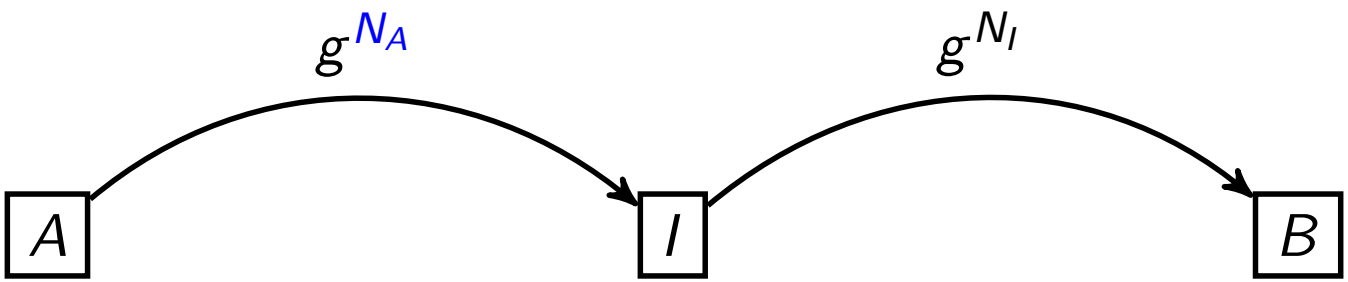
Attaque de type "Man in the middle" :



② L'intrus peut intercepter/bloquer tous les messages

Les capacités de l'intrus – Modèle de Dolev-Yao

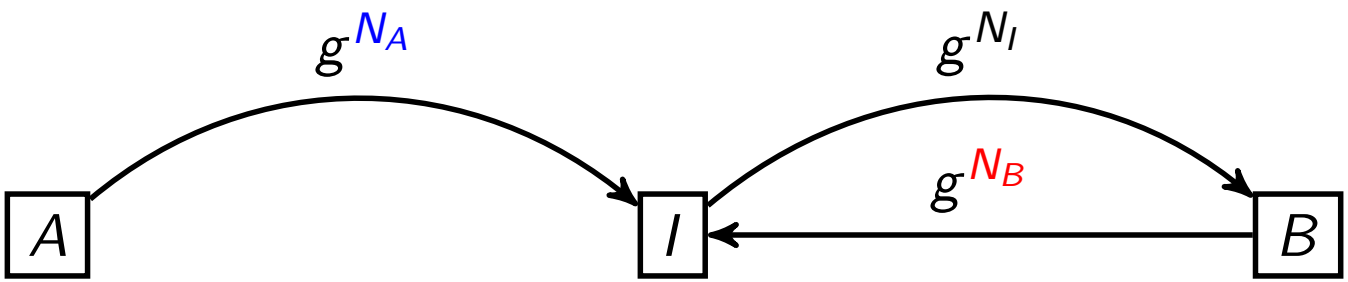
Attaque de type "Man in the middle" :



③ L'intrus peut générer des valeurs aléatoires

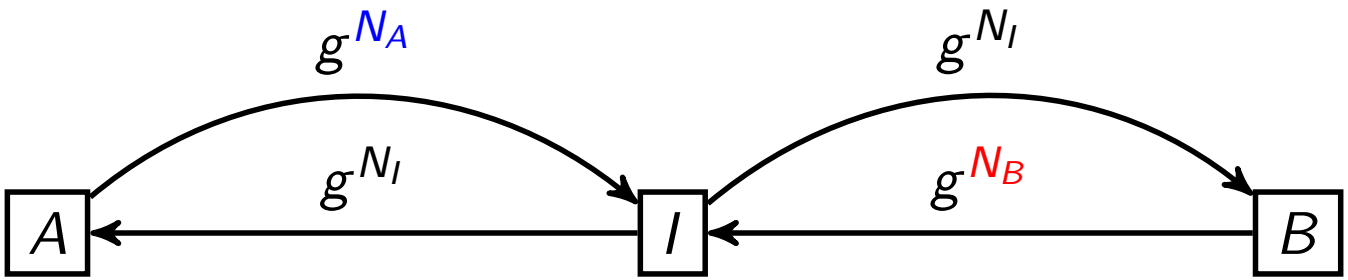
Les capacités de l'intrus – Modèle de Dolev-Yao

Attaque de type "Man in the middle" :



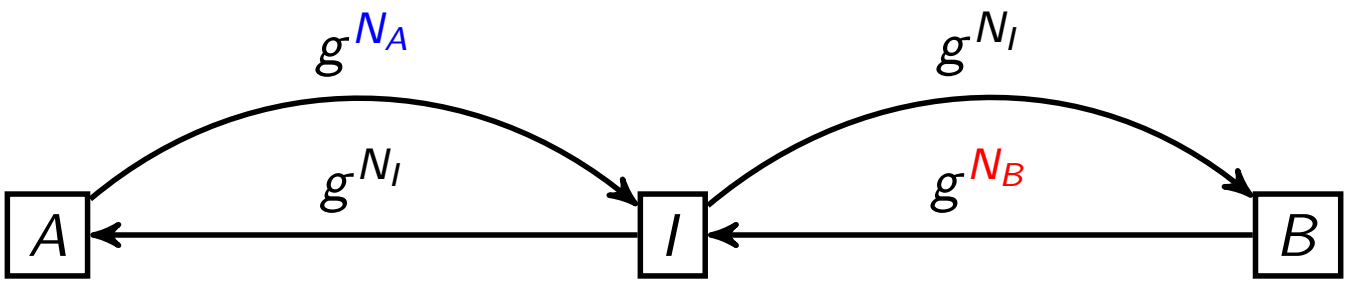
Les capacités de l'intrus – Modèle de Dolev-Yao

Attaque de type "Man in the middle" :



Les capacités de l'intrus – Modèle de Dolev-Yao

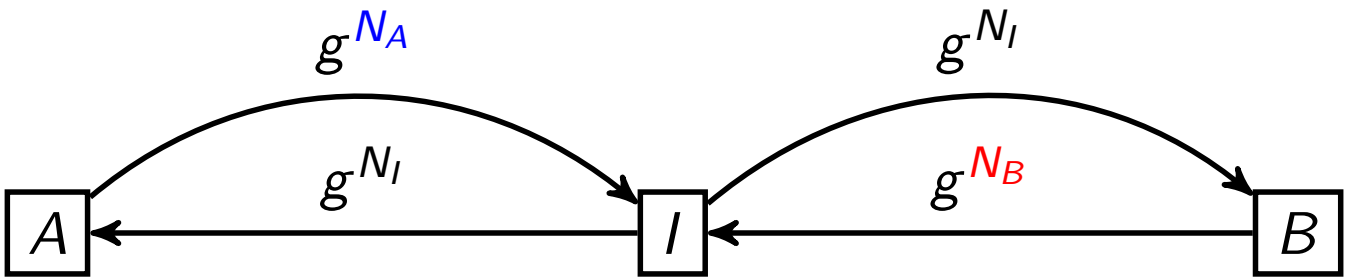
Attaque de type "Man in the middle" :



$$K_1 = (g^{N_A})^{N_I}$$

Les capacités de l'intrus – Modèle de Dolev-Yao

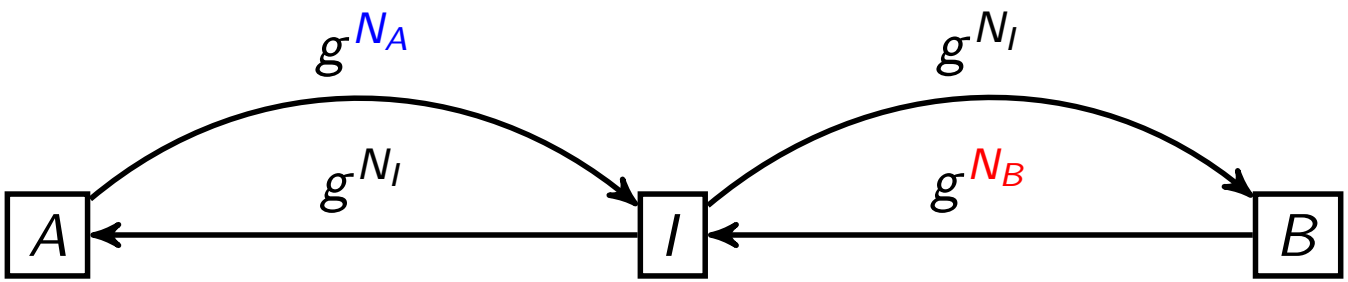
Attaque de type "Man in the middle" :



$K_1 = (g^{N_A})^{N_I}$ $K_2 = (g^{N_B})^{N_I}$

Les capacités de l'intrus – Modèle de Dolev-Yao

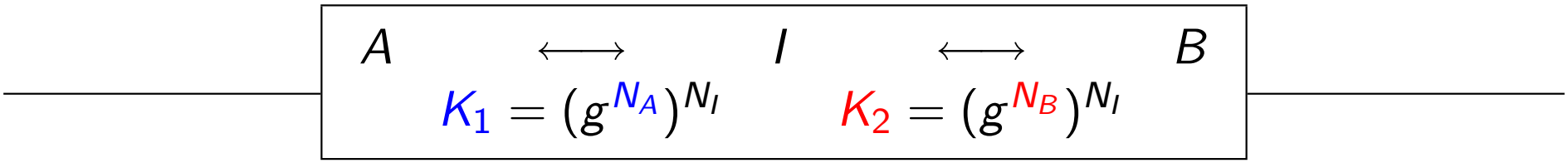
Attaque de type "Man in the middle" :



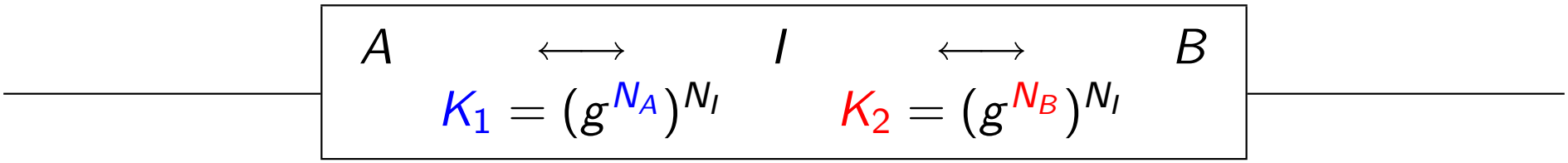
$K_1 = (g^{N_A})^{N_I}$ $K_2 = (g^{N_B})^{N_I}$

④ L'intrus peut exécuter plusieurs sessions entrelacées d'un même protocole

Les capacités de l'intrus – Modèle de Dolev-Yao



Les capacités de l'intrus – Modèle de Dolev-Yao

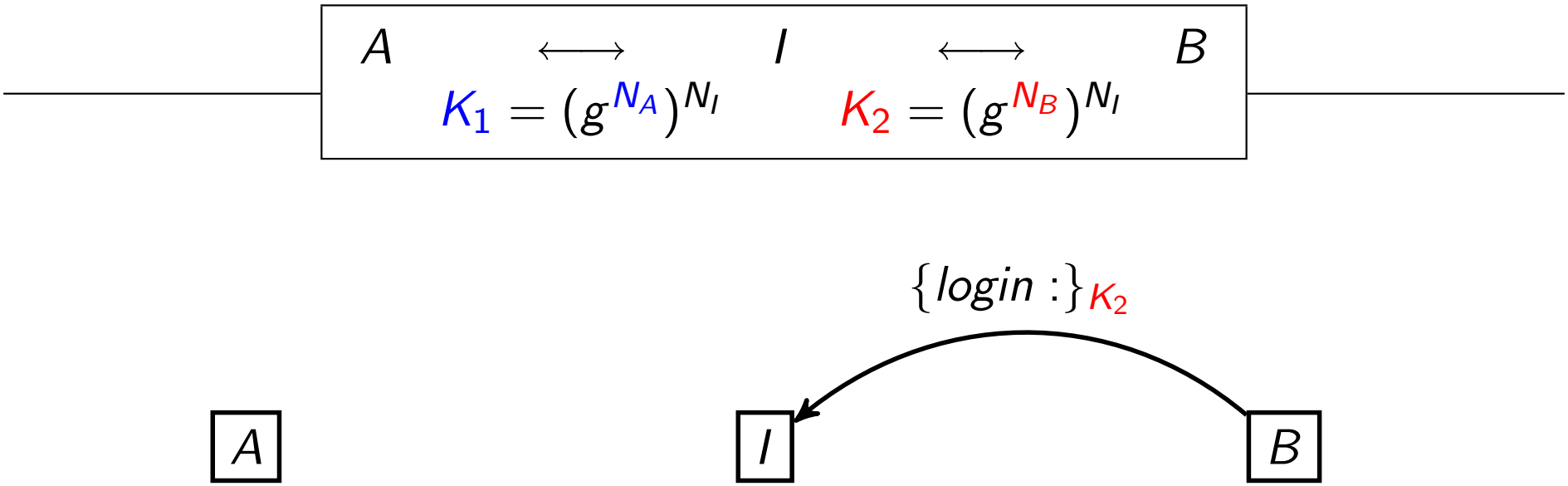


Les capacités de l'intrus – Modèle de Dolev-Yao

$A \longleftrightarrow I \longleftrightarrow B$

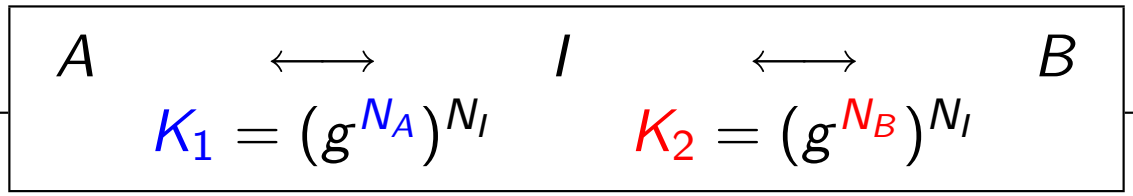
$K_1 = (g^{N_A})^{N_I}$ $\quad$ $K_2 = (g^{N_B})^{N_I}$

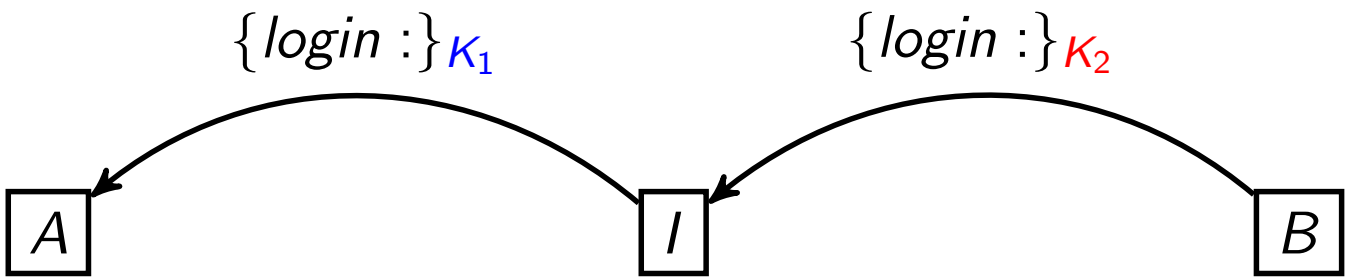
Les capacités de l'intrus – Modèle de Dolev-Yao



⑤ L'intrus peut déchiffrer un message chiffré s'il a la clé inverse

Les capacités de l'intrus – Modèle de Dolev-Yao



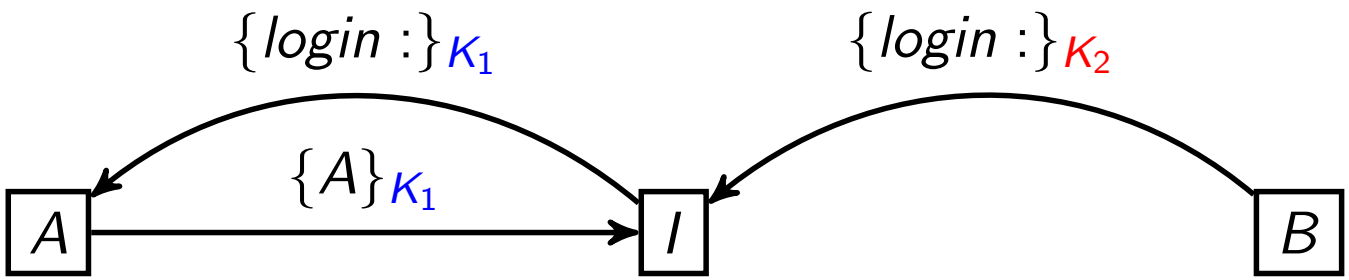


⑥ L'intrus peut *chiffrer* s'il a la clé

Les capacités de l'intrus – Modèle de Dolev-Yao

$A \longleftrightarrow I \longleftrightarrow B$

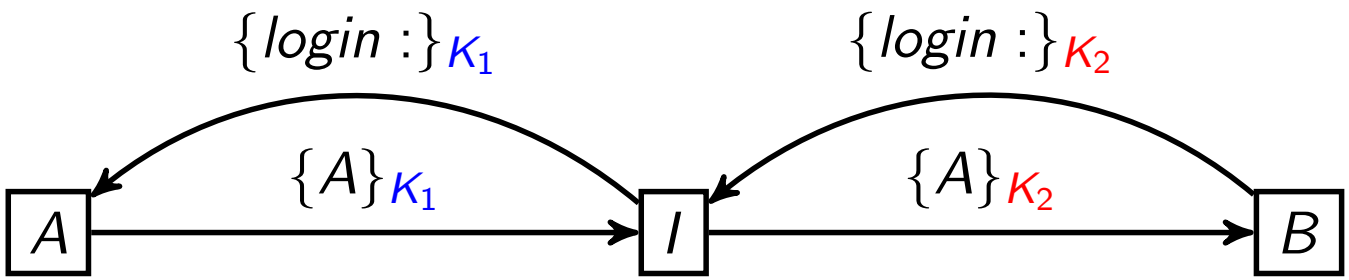
$K_1 = (g^{N_A})^{N_I}$ $\quad$ $K_2 = (g^{N_B})^{N_I}$



Les capacités de l'intrus – Modèle de Dolev-Yao

$A \longleftrightarrow I \longleftrightarrow B$

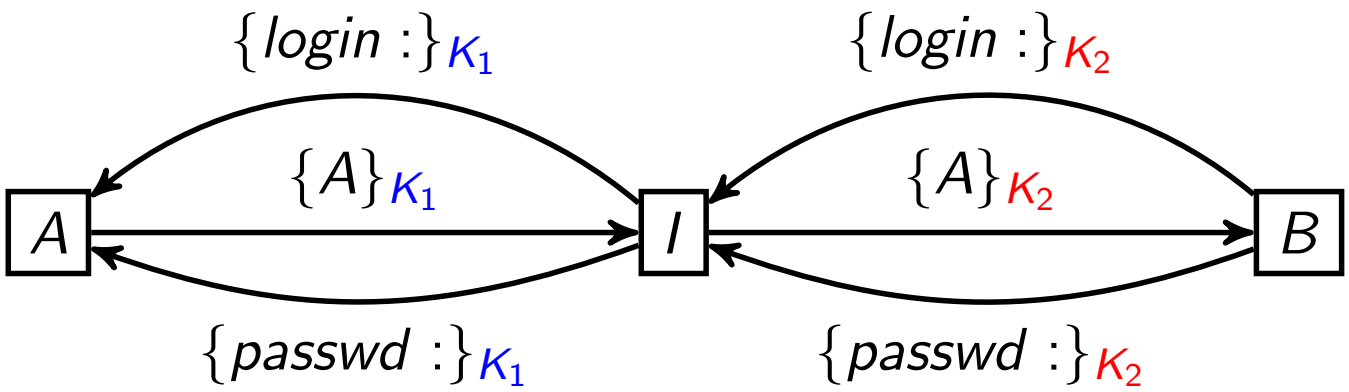
$K_1 = (g^{N_A})^{N_I}$ $\quad$ $K_2 = (g^{N_B})^{N_I}$



Les capacités de l'intrus – Modèle de Dolev-Yao

$A \longleftrightarrow I \longleftrightarrow B$

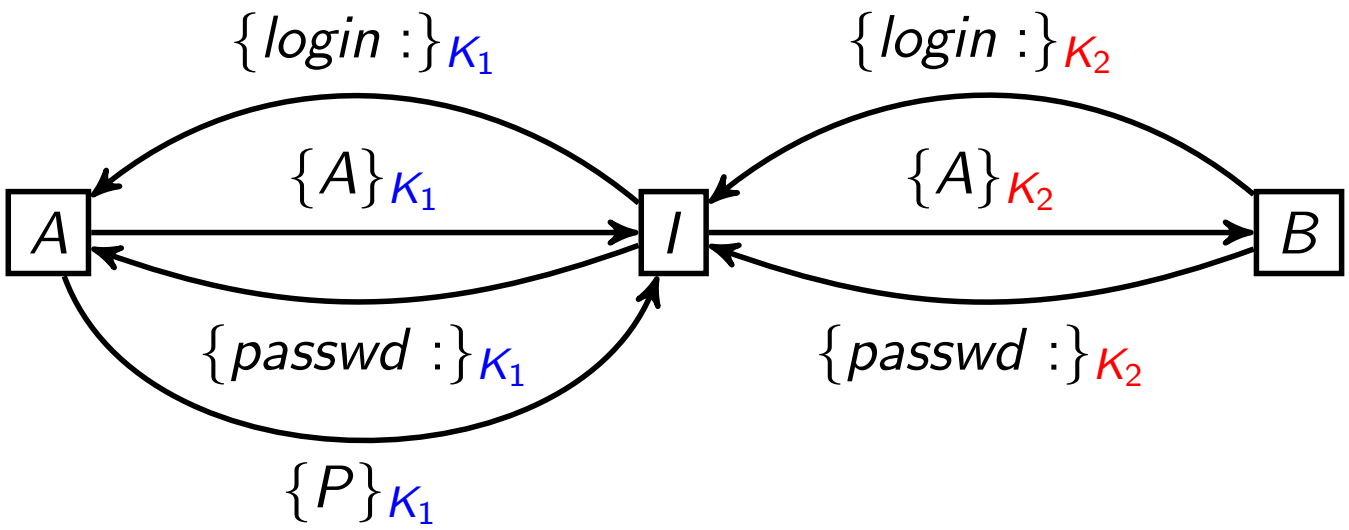
$K_1 = (g^{N_A})^{N_I}$ $\quad$ $K_2 = (g^{N_B})^{N_I}$



Les capacités de l'intrus – Modèle de Dolev-Yao

$A \longleftrightarrow I \longleftrightarrow B$

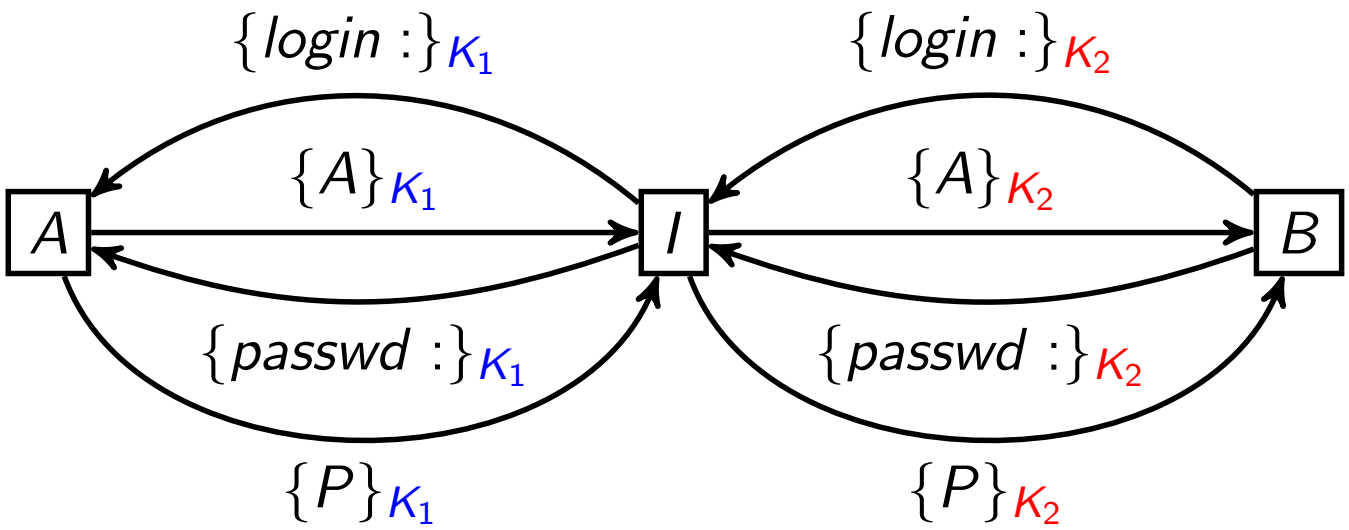
$K_1 = (g^{N_A})^{N_I}$ $\quad$ $K_2 = (g^{N_B})^{N_I}$



Les capacités de l'intrus – Modèle de Dolev-Yao

$A \longleftrightarrow I \longleftrightarrow B$

$K_1 = (g^{N_A})^{N_I}$ $\quad$ $K_2 = (g^{N_B})^{N_I}$



⑦ L'intrus peut envoyer n'importe quel message construit à partir des connaissances qu'il a accumulé

Les capacités de l'intrus – Modèle de Dolev-Yao

① L'intrus ne fait pas de cryptanalyse

② L'intrus peut intercepter/bloquer tous les messages

③ L'intrus peut générer des valeurs aléatoires

④ L'intrus peut exécuter plusieurs sessions entrelacées d'un même protocole

⑤ L'intrus peut déchiffrer un message chiffré s'il a la clé inverse

⑥ L'intrus peut *chiffrer* s'il a la clé

⑦ L'intrus peut envoyer n'importe quel message construit à partir des connaissances qu'il a accumulé

Les capacités de l'intrus – Modèle de Dolev-Yao

① L'intrus ne fait pas de cryptanalyse

② L'intrus peut intercepter/bloquer tous les messages

③ L'intrus peut générer des valeurs aléatoires

④ L'intrus peut exécuter plusieurs sessions entrelacées d'un même protocole

⑤ L'intrus peut déchiffrer un message chiffré s'il a la clé inverse

⑥ L'intrus peut *chiffrer* s'il a la clé

⑦ L'intrus peut envoyer n'importe quel message construit à partir des connaissances qu'il a accumulé

Modèle de Dolev-Yao= L'intrus est le réseau

Formalisation logique de Dolev-Yao

Formalisation logique de Dolev-Yao

- $I =$ connaissance initiale de l'intrus

Formalisation logique de Dolev-Yao

- I = connaissance initiale de l'intrus
  - les identités de tous les acteurs
  - toutes les clés publiques
  - des clés privées corrompues

Formalisation logique de Dolev-Yao

- I = connaissance initiale de l'intrus
  - les identités de tous les acteurs
  - toutes les clés publiques
  - des clés privées corrompues
- $\left.\begin{array}{l}\text{Manipulations de l'intrus}\\ \text{Exécution du protocole}\end{array}\right| = \text{déductions sur } I$

Formalisation logique de Dolev-Yao

- I = connaissance initiale de l'intrus
  - les identités de tous les acteurs
  - toutes les clés publiques
  - des clés privées corrompues
- $\left.\begin{array}{l}\text{Manipulations de l'intrus}\\ \text{Exécution du protocole}\end{array}\right| = \text{déductions sur } I$

Manipulations de l'intrus

Formalisation logique de Dolev-Yao

- I = connaissance initiale de l'intrus
  - les identités de tous les acteurs
  - toutes les clés publiques
  - des clés privées corrompues
- $\left.\begin{array}{l}\text{Manipulations de l'intrus}\\ \text{Exécution du protocole}\end{array}\right| = \text{déductions sur } I$

Manipulations de l'intrus

$$\frac{}{I, M \vdash M}$$

Formalisation logique de Dolev-Yao

- I= connaissance initiale de l'intrus
  - les identités de tous les acteurs
  - toutes les clés publiques
  - des clés privées corrompues
- $\left.\begin{array}{l}\text{Manipulations de l'intrus}\\ \text{Exécution du protocole}\end{array}\right| = \text{déductions sur } I$

Manipulations de l'intrus

$$\frac{}{I, M \vdash M} \qquad \frac{I \vdash \{M\}_K \quad I \vdash K^{-1}}{I \vdash M}$$

Formalisation logique de Dolev-Yao

- I = connaissance initiale de l'intrus
  - les identités de tous les acteurs
  - toutes les clés publiques
  - des clés privées corrompues
- Manipulations de l'intrus
Exécution du protocole $\Big|$ = déductions sur I

Manipulations de l'intrus

$$\frac{}{I, M \vdash M} \qquad \frac{I \vdash \{M\}_K \quad I \vdash K^{-1}}{I \vdash M} \qquad \frac{I \vdash M \quad I \vdash K}{I \vdash \{M\}_K}$$

Formalisation logique de Dolev-Yao

- I = connaissance initiale de l'intrus
  - les identités de tous les acteurs
  - toutes les clés publiques
  - des clés privées corrompues
- $\left.\begin{array}{l}\text{Manipulations de l'intrus}\\ \text{Exécution du protocole}\end{array}\right| = \text{déductions sur } I$

Manipulations de l'intrus

$$\frac{}{I, M \vdash M} \qquad \frac{I \vdash \{M\}_K \quad I \vdash K^{-1}}{I \vdash M} \qquad \frac{I \vdash M \quad I \vdash K}{I \vdash \{M\}_K}$$

$$\frac{I \vdash X, Y}{I \vdash X} \qquad \frac{I \vdash X, Y}{I \vdash Y}$$

Formalisation logique de Dolev-Yao

- I = connaissance initiale de l'intrus
  - les identités de tous les acteurs
  - toutes les clés publiques
  - des clés privées corrompues
- Manipulations de l'intrus
Exécution du protocole $\Big|$ = déductions sur I

Manipulations de l'intrus

$$\frac{}{I, M \vdash M} \qquad \frac{I \vdash \{M\}_K \quad I \vdash K^{-1}}{I \vdash M} \qquad \frac{I \vdash M \quad I \vdash K}{I \vdash \{M\}_K}$$

$$\frac{I \vdash X, Y}{I \vdash X} \qquad \frac{I \vdash X, Y}{I \vdash Y} \qquad \frac{I \vdash X \quad I \vdash Y}{I \vdash X, Y}$$

Formalisation logique de Dolev-Yao

Exécution du protocole = règles de déduction supplémentaires

Formalisation logique de Dolev-Yao

Exécution du protocole = règles de déduction supplémentaires

Exemple : $\begin{cases} 1. & A \hookrightarrow B : g^{N_A} \\ 2. & B \hookrightarrow A : g^{N_B} \\ 3. & A \hookrightarrow B : \{s\}_K \quad \text{où} \quad K = (g^{N_B})^{N_A} \end{cases}$

Formalisation logique de Dolev-Yao

Exécution du protocole = règles de déduction supplémentaires

Exemple : $\begin{cases} 1. & A \hookrightarrow B : g^{N_A} \\ 2. & B \hookrightarrow A : g^{N_B} \\ 3. & A \hookrightarrow B : \{s\}_K \quad \text{où} \quad K = (g^{N_B})^{N_A} \end{cases}$

$$1.\ \frac{I \vdash a \quad I \vdash g}{I \vdash g^{N_a}}$$

Formalisation logique de Dolev-Yao

Exécution du protocole = règles de déduction supplémentaires

Exemple : $\begin{cases} 1. & A \hookrightarrow B : g^{N_A} \\ 2. & B \hookrightarrow A : g^{N_B} \\ 3. & A \hookrightarrow B : \{s\}_K \quad \text{où} \quad K = (g^{N_B})^{N_A} \end{cases}$

$$1.\frac{I \vdash a \quad I \vdash g}{I \vdash g^{N_a}} \qquad 2.\frac{I \vdash b \quad I \vdash X}{I \vdash g^{N_b}}$$

Formalisation logique de Dolev-Yao

Exécution du protocole = règles de déduction supplémentaires

Exemple : $\begin{cases} 1. & A \hookrightarrow B : g^{N_A} \\ 2. & B \hookrightarrow A : g^{N_B} \\ 3. & A \hookrightarrow B : \{s\}_K \quad \text{où} \quad K = (g^{N_B})^{N_A} \end{cases}$

$$1.\frac{I \vdash a \quad I \vdash g}{I \vdash g^{N_a}} \qquad 2.\frac{I \vdash b \quad I \vdash X}{I \vdash g^{N_b}} \qquad 3.\frac{I \vdash a \quad I \vdash Y}{I \vdash \{s\}_{Y^{N_a}}}$$

Formalisation logique de Dolev-Yao

$$1.\ \frac{I \vdash a \quad I \vdash g}{I \vdash g^{N_a}} \qquad 2.\ \frac{I \vdash b \quad I \vdash X}{I \vdash g^{N_b}} \qquad 3.\ \frac{I \vdash a \quad I \vdash Y}{I \vdash \{s\}_{Y^{N_a}}}$$

$$\frac{}{I, M \vdash M} \qquad \frac{I \vdash \{M\}_K \quad I \vdash K^{-1}}{I \vdash M} \qquad \frac{I \vdash M \quad I \vdash K}{I \vdash \{M\}_K}$$

$$\frac{I \vdash X, Y}{I \vdash X} \qquad \frac{I \vdash X, Y}{I \vdash Y} \qquad \frac{I \vdash X \quad I \vdash Y}{I \vdash X, Y}$$

$$a, b, g \vdash \{\{s\}_{g^{N_a}}\}_g$$

Formalisation logique de Dolev-Yao

$$1.\ \frac{I \vdash a \quad I \vdash g}{I \vdash g^{N_a}} \qquad 2.\ \frac{I \vdash b \quad I \vdash X}{I \vdash g^{N_b}} \qquad 3.\ \frac{I \vdash a \quad I \vdash Y}{I \vdash \{s\}_{Y^{N_a}}}$$

$$\frac{}{I, M \vdash M} \qquad \frac{I \vdash \{M\}_K \quad I \vdash K^{-1}}{I \vdash M} \qquad \frac{I \vdash M \quad I \vdash K}{I \vdash \{M\}_K}$$

$$\frac{I \vdash X, Y}{I \vdash X} \qquad \frac{I \vdash X, Y}{I \vdash Y} \qquad \frac{I \vdash X \quad I \vdash Y}{I \vdash X, Y}$$

$$\frac{a, b, g \vdash \{s\}_{g^{N_a}} \qquad \dfrac{}{a, b, g \vdash g}}{a, b, g \vdash \{\{s\}_{g^{N_a}}\}_g}$$

Formalisation logique de Dolev-Yao

$$1.\ \frac{I \vdash a \quad I \vdash g}{I \vdash g^{N_a}} \qquad 2.\ \frac{I \vdash b \quad I \vdash X}{I \vdash g^{N_b}} \qquad 3.\ \frac{I \vdash a \quad I \vdash Y}{I \vdash \{s\}_{Y^{N_a}}}$$

$$\frac{}{I, M \vdash M} \qquad \frac{I \vdash \{M\}_K \quad I \vdash K^{-1}}{I \vdash M} \qquad \frac{I \vdash M \quad I \vdash K}{I \vdash \{M\}_K}$$

$$\frac{I \vdash X, Y}{I \vdash X} \qquad \frac{I \vdash X, Y}{I \vdash Y} \qquad \frac{I \vdash X \quad I \vdash Y}{I \vdash X, Y}$$

$$\frac{a, b, g \vdash \{s\}_{g^{N_a}} \qquad \dfrac{}{a, b, g \vdash g}}{a, b, g \vdash \{\{s\}_{g^{N_a}}\}_g}$$

Formalisation logique de Dolev-Yao

$$1.\ \frac{I \vdash a \quad I \vdash g}{I \vdash g^{N_a}} \qquad 2.\ \frac{I \vdash b \quad I \vdash X}{I \vdash g^{N_b}} \qquad 3.\ \frac{I \vdash a \quad I \vdash Y}{I \vdash \{s\}_{Y^{N_a}}}$$

$$\frac{}{I, M \vdash M} \qquad \frac{I \vdash \{M\}_K \quad I \vdash K^{-1}}{I \vdash M} \qquad \frac{I \vdash M \quad I \vdash K}{I \vdash \{M\}_K}$$

$$\frac{I \vdash X, Y}{I \vdash X} \qquad \frac{I \vdash X, Y}{I \vdash Y} \qquad \frac{I \vdash X \quad I \vdash Y}{I \vdash X, Y}$$

$$\dfrac{\dfrac{\dfrac{}{a, b, g \vdash a} \quad \dfrac{}{a, b, g \vdash g}}{a, b, g \vdash \{s\}_{g^{N_a}}} \quad \dfrac{}{a, b, g \vdash g}}{a, b, g \vdash \{\{s\}_{g^{N_a}}\}_g}$$

Formalisation logique de Dolev-Yao

$$1.\ \frac{I \vdash a \quad I \vdash g}{I \vdash g^{N_a}} \qquad 2.\ \frac{I \vdash b \quad I \vdash X}{I \vdash g^{N_b}} \qquad 3.\ \frac{I \vdash a \quad I \vdash Y}{I \vdash \{s\}_{Y^{N_a}}}$$

$$\frac{}{I, M \vdash M} \qquad \frac{I \vdash \{M\}_K \quad I \vdash K^{-1}}{I \vdash M} \qquad \frac{I \vdash M \quad I \vdash K}{I \vdash \{M\}_K}$$

$$\frac{I \vdash X, Y}{I \vdash X} \qquad \frac{I \vdash X, Y}{I \vdash Y} \qquad \frac{I \vdash X \quad I \vdash Y}{I \vdash X, Y}$$

$$\dfrac{\dfrac{\dfrac{}{a, b, g \vdash a} \quad \dfrac{}{a, b, g \vdash g}}{a, b, g \vdash \{s\}_{g^{N_a}}} \quad \dfrac{}{a, b, g \vdash g}}{a, b, g \vdash \{\{s\}_{g^{N_a}}\}_g}$$

Abrégé en $a, b, g \vdash^2 \{\{s\}_{g^{N_a}}\}_g$

Vérification logique sur Dolev-Yao

- Preuve que s est secret $= \boxed{\forall n : I \not\vdash^{n} s}$

Vérification logique sur Dolev-Yao

- Preuve que s est secret = $\boxed{\forall n : I \not\vdash^{n} s}$
- Vérification complexe : 3 dimensions non bornées

Vérification logique sur Dolev-Yao

- Preuve que s est secret $= \boxed{\forall n : I \not\vdash^n s}$
- Vérification complexe : 3 dimensions non bornées
  - Nombre d'agents exécutant le protocole en //

Vérification logique sur Dolev-Yao

- Preuve que s est secret $= \boxed{\forall n : I \not\vdash^n s}$
- Vérification complexe : 3 dimensions non bornées
  - Nombre d'agents exécutant le protocole en //
  - Nombre de sessions pour chaque agent

Vérification logique sur Dolev-Yao

- Preuve que s est secret $= \boxed{\forall n : I \not\vdash^n s}$
- Vérification complexe : 3 dimensions non bornées
  - Nombre d'agents exécutant le protocole en //
  - Nombre de sessions pour chaque agent
  - Déductions/constructions de l'intrus sur sa connaissance

Vérification logique sur Dolev-Yao

- Preuve que s est secret $= \boxed{\forall n : I \not\vdash^n s}$
- Vérification complexe : 3 dimensions non bornées
  - Nombre d'agents exécutant le protocole en //
  - Nombre de sessions pour chaque agent
  - Déductions/constructions de l'intrus sur sa connaissance
- Trois grandes catégories de travaux et d'outils :
  - Vérification automatique sur modèle fini (model-checking)

    Pour nb d'agents fixé et k fixé : $\boxed{\forall n \leq k : I \not\vdash^n s}$

Vérification logique sur Dolev-Yao

- Preuve que s est secret = $\boxed{\forall n : I \not\vdash^n s}$
- Vérification complexe : 3 dimensions non bornées
  - Nombre d'agents exécutant le protocole en //
  - Nombre de sessions pour chaque agent
  - Déductions/constructions de l'intrus sur sa connaissance
- Trois grandes catégories de travaux et d'outils :
  - Vérification automatique sur modèle fini (model-checking)

    Pour nb d'agents fixé et k fixé : $\boxed{\forall n \leq k : I \not\vdash^n s}$

    Détection d'attaques : $\boxed{\exists k : I \vdash^k s}$

Vérification logique sur Dolev-Yao

- Preuve que s est secret = $\boxed{\forall n : I \not\vdash^n s}$
- Vérification complexe : 3 dimensions non bornées
  - Nombre d'agents exécutant le protocole en //
  - Nombre de sessions pour chaque agent
  - Déductions/constructions de l'intrus sur sa connaissance
- Trois grandes catégories de travaux et d'outils :
  - Vérification automatique sur modèle fini (model-checking)

    Pour nb d'agents fixé et k fixé : $\boxed{\forall n \leq k : I \not\vdash^n s}$

    Détection d'attaques : $\boxed{\exists k : I \vdash^k s}$
  - Vérification semi-automatique (preuve assistée)

    Par induction sur nb d'agents et n : $\boxed{\forall n : I \not\vdash^n s}$

Vérification logique sur Dolev-Yao

- Preuve que s est secret = $\boxed{\forall n : I \not\vdash^n s}$
- Vérification complexe : 3 dimensions non bornées
  - Nombre d'agents exécutant le protocole en //
  - Nombre de sessions pour chaque agent
  - Déductions/constructions de l'intrus sur sa connaissance
- Trois grandes catégories de travaux et d'outils :
  - Vérification automatique sur modèle fini (model-checking)

    Pour nb d'agents fixé et k fixé : $\boxed{\forall n \leq k : I \not\vdash^n s}$

    Détection d'attaques : $\boxed{\exists k : I \vdash^k s}$
  - Vérification semi-automatique (preuve assistée)

    Par induction sur nb d'agents et n : $\boxed{\forall n : I \not\vdash^n s}$
  - Vérification automatique par approximation (abstraction)

    Approximations $I^\sharp \supseteq I$ et $\vdash^\sharp \supseteq \vdash$ telles que $\boxed{\forall n : I^\sharp \not\vdash^{\sharp n} s}$

Vérification automatique par approximation

Idée= construire le langage des termes déductibles =*Store*

Vérification automatique par approximation

Idée= construire le langage des termes déductibles =*Store*

$$\dfrac{\dfrac{\dfrac{}{M, K \vdash M} \qquad \dfrac{}{M, K \vdash K}}{M, K \vdash \{M\}_K} \qquad \dfrac{}{M, K \vdash K}}{M, K \vdash \{\{M\}_K\}_K}$$

Vérification automatique par approximation

Idée= construire le langage des termes déductibles =*Store*

$$\dfrac{\dfrac{\dfrac{}{M, K \vdash M} \qquad \dfrac{}{M, K \vdash K}}{M, K \vdash \{M\}_K} \qquad \dfrac{}{M, K \vdash K}}{M, K \vdash \{\{M\}_K\}_K}$$

$$Store = \{M, K, \{M\}_K, \{\{M\}_K\}_K, \ldots\} \;=\; \{T \mid \forall n : I \vdash^n T\}$$

Vérification automatique par approximation

Idée= construire le langage des termes déductibles =*Store*

$$\dfrac{\dfrac{\dfrac{}{M, K \vdash M} \qquad \dfrac{}{M, K \vdash K}}{M, K \vdash \{M\}_K} \qquad \dfrac{}{M, K \vdash K}}{M, K \vdash \{\{M\}_K\}_K}$$

$$Store = \{M, K, \{M\}_K, \{\{M\}_K\}_K, \ldots\} \;=\; \{T \mid \forall n : I \vdash^n T\}$$

Construction d'un *automate approximation* $\mathcal{A}$

Vérification automatique par approximation

Idée= construire le langage des termes déductibles =*Store*

$$\dfrac{\dfrac{\dfrac{}{M,K \vdash M} \qquad \dfrac{}{M,K \vdash K}}{M,K \vdash \{M\}_K} \qquad \dfrac{}{M,K \vdash K}}{M,K \vdash \{\{M\}_K\}_K}$$

$$Store = \{M, K, \{M\}_K, \{\{M\}_K\}_K, \ldots\} \;=\; \{T \mid \forall n : I \vdash^n T\}$$

Construction d'un *automate approximation* $\mathcal{A}$ t.q. $L(\mathcal{A}) \supseteq Store$

Vérification automatique par approximation

Idée= construire le langage des termes déductibles =$Store$

$$\dfrac{\dfrac{\dfrac{}{M,K \vdash M} \qquad \dfrac{}{M,K \vdash K}}{M,K \vdash \{M\}_K} \qquad \dfrac{}{M,K \vdash K}}{M,K \vdash \{\{M\}_K\}_K}$$

$$Store = \{M, K, \{M\}_K, \{\{M\}_K\}_K, \ldots\} \;=\; \{T \mid \forall n : I \vdash^n T\}$$

Construction d'un *automate approximation* $\mathcal{A}$ t.q. $L(\mathcal{A}) \supseteq Store$

Secret de $s \equiv$ $\quad s \notin L(\mathcal{A})$

Vérification automatique par approximation

Idée= construire le langage des termes déductibles =$Store$

$$\dfrac{\dfrac{\dfrac{}{M,K \vdash M} \qquad \dfrac{}{M,K \vdash K}}{M,K \vdash \{M\}_K} \qquad \dfrac{}{M,K \vdash K}}{M,K \vdash \{\{M\}_K\}_K}$$

$Store = \{M, K, \{M\}_K, \{\{M\}_K\}_K, \ldots\} \;=\; \{T \mid \forall n : I \vdash^n T\}$

Construction d'un *automate approximation* $\mathcal{A}$ t.q. $L(\mathcal{A}) \supseteq Store$

Secret de $s \equiv \quad s \notin L(\mathcal{A})$	$\Rightarrow s \notin Store$

Vérification automatique par approximation

Idée= construire le langage des termes déductibles =$Store$

$$\dfrac{\dfrac{\dfrac{}{M,K \vdash M} \qquad \dfrac{}{M,K \vdash K}}{M,K \vdash \{M\}_K} \qquad \dfrac{}{M,K \vdash K}}{M,K \vdash \{\{M\}_K\}_K}$$

$Store = \{M, K, \{M\}_K, \{\{M\}_K\}_K, \ldots\} \;=\; \{T \mid \forall n : I \vdash^n T\}$

Construction d'un *automate approximation* $\mathcal{A}$ t.q. $L(\mathcal{A}) \supseteq Store$

$\boxed{\text{Secret de } s \equiv \quad s \notin L(\mathcal{A})} \quad \Rightarrow s \notin Store \quad \Rightarrow \forall n : I \not\vdash^n s$

Méthodologie d'approximation pour les protocoles

$A \hookrightarrow B : \{N_A^1, A\}_{K_B}$	$C \hookrightarrow D : \{N_C^1, C\}_{K_D}$	$E \hookrightarrow A : \{N_E^1, E\}_{K_A}$
$B \hookrightarrow A : \{N_A^1, N_B^1\}_{K_A}$	$D \hookrightarrow C : \{N_C^1, N_D^1\}_{K_C}$	$A \hookrightarrow E : \{N_E^1, N_A^1\}_{K_E}$
$A \hookrightarrow B : \{N_B^1\}_{K_B}$	$C \hookrightarrow D : \{N_D^1\}_{K_D}$	$E \hookrightarrow A : \{N_A^1\}_{K_A}$
$A \hookrightarrow B : \{N_A^2, A\}_{K_B}$	$C \hookrightarrow D : \{N_C^2, C\}_{K_D}$	$E \hookrightarrow A : \{N_E^2, E\}_{K_A}$
$B \hookrightarrow A : \{N_A^2, N_B^2\}_{K_A}$	$D \hookrightarrow C : \{N_C^2, N_D^2\}_{K_C}$	$A \hookrightarrow E : \{N_E^2, N_A^2\}_{K_E}$
$A \hookrightarrow B : \{N_B^2\}_{K_B}$	$C \hookrightarrow D : \{N_D^2\}_{K_D}$	$E \hookrightarrow A : \{N_A^2\}_{K_A}$
$A \hookrightarrow B : \{N_A^3, A\}_{K_B}$	$C \hookrightarrow D : \{N_C^3, C\}_{K_D}$	$E \hookrightarrow A : \{N_E^3, E\}_{K_A}$
$B \hookrightarrow A : \{N_A^3, N_B^3\}_{K_A}$	$D \hookrightarrow C : \{N_C^3, N_D^3\}_{K_C}$	$A \hookrightarrow E : \{N_E^3, N_A^3\}_{K_E}$
$A \hookrightarrow B : \{N_B^3\}_{K_B}$	$C \hookrightarrow D : \{N_D^3\}_{K_D}$	$E \hookrightarrow A : \{N_A^3\}_{K_A}$
$A \hookrightarrow B : \{N_A^4, A\}_{K_B}$	$C \hookrightarrow D : \{N_C^4, C\}_{K_D}$	$E \hookrightarrow A : \{N_E^4, E\}_{K_A}$
$B \hookrightarrow A : \{N_A^4, N_B^4\}_{K_A}$	$D \hookrightarrow C : \{N_C^4, N_D^4\}_{K_C}$	$A \hookrightarrow E : \{N_E^4, N_A^4\}_{K_E}$
$A \hookrightarrow B : \{N_B^4\}_{K_B}$	$C \hookrightarrow D : \{N_D^4\}_{K_D}$	$E \hookrightarrow A : \{N_A^4\}_{K_A}$
$A \hookrightarrow B : \{N_A^5, A\}_{K_B}$	$C \hookrightarrow D : \{N_C^5, C\}_{K_D}$	$E \hookrightarrow A : \{N_E^5, E\}_{K_A}$
$B \hookrightarrow A : \{N_A^5, N_B^5\}_{K_A}$	$D \hookrightarrow C : \{N_C^5, N_D^5\}_{K_C}$	$A \hookrightarrow E : \{N_E^5, N_A^5\}_{K_E}$
$A \hookrightarrow B : \{N_B^5\}_{K_B}$	$C \hookrightarrow D : \{N_D^5\}_{K_D}$	$E \hookrightarrow A : \{N_A^5\}_{K_A}$

Méthodologie d'approximation pour les protocoles

$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, \blacksquare\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, \square\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\blacksquare}, \square\}_{K_{\square}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, N_{\square}\}_{K_{\square}}$	$\square \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\square}\}_{K_{\blacksquare}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, \blacksquare\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, \square\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\blacksquare}, \square\}_{K_{\square}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, N_{\square}\}_{K_{\square}}$	$\square \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\square}\}_{K_{\blacksquare}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, \blacksquare\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, \square\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\blacksquare}, \square\}_{K_{\square}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, N_{\square}\}_{K_{\square}}$	$\square \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\square}\}_{K_{\blacksquare}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, \blacksquare\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, \square\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\blacksquare}, \square\}_{K_{\square}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, N_{\square}\}_{K_{\square}}$	$\square \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\square}\}_{K_{\blacksquare}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, \blacksquare\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, \square\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\blacksquare}, \square\}_{K_{\square}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}, N_{\square}\}_{K_{\square}}$	$\square \hookrightarrow \blacksquare : \{N_{\blacksquare}, N_{\square}\}_{K_{\blacksquare}}$
$\blacksquare \hookrightarrow \blacksquare : \{N_{\blacksquare}\}_{K_{\blacksquare}}$	$\square \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$	$\blacksquare \hookrightarrow \square : \{N_{\square}\}_{K_{\square}}$

Pour prouver le secret de $N_{\square}$: Honnêtes $= \square$, Intrus$= \blacksquare$, $N_X^i = N_X^{i+1}$

Méthodologie d'approximation pour les protocoles

$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, \blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_A, A\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_\blacksquare, A\}_{K_A}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, N_\blacksquare\}_{K_\blacksquare}$	$B \hookrightarrow A : \{N_A, N_B\}_{K_A}$	$A \hookrightarrow \blacksquare : \{N_\blacksquare, N_A\}_{K_\blacksquare}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_B\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_A\}_{K_A}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, \blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_A, A\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_\blacksquare, A\}_{K_A}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, N_\blacksquare\}_{K_\blacksquare}$	$B \hookrightarrow A : \{N_A, N_B\}_{K_A}$	$A \hookrightarrow \blacksquare : \{N_\blacksquare, N_A\}_{K_\blacksquare}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_B\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_A\}_{K_A}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, \blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_A, A\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_\blacksquare, A\}_{K_A}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, N_\blacksquare\}_{K_\blacksquare}$	$B \hookrightarrow A : \{N_A, N_B\}_{K_A}$	$A \hookrightarrow \blacksquare : \{N_\blacksquare, N_A\}_{K_\blacksquare}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_B\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_A\}_{K_A}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, \blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_A, A\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_\blacksquare, A\}_{K_A}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, N_\blacksquare\}_{K_\blacksquare}$	$B \hookrightarrow A : \{N_A, N_B\}_{K_A}$	$A \hookrightarrow \blacksquare : \{N_\blacksquare, N_A\}_{K_\blacksquare}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_B\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_A\}_{K_A}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, \blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_A, A\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_\blacksquare, A\}_{K_A}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare, N_\blacksquare\}_{K_\blacksquare}$	$B \hookrightarrow A : \{N_A, N_B\}_{K_A}$	$A \hookrightarrow \blacksquare : \{N_\blacksquare, N_A\}_{K_\blacksquare}$
$\blacksquare \hookrightarrow \blacksquare : \{N_\blacksquare\}_{K_\blacksquare}$	$A \hookrightarrow B : \{N_B\}_{K_B}$	$\blacksquare \hookrightarrow A : \{N_A\}_{K_A}$

Pour l'authentification : A, B honnêtes, Intrus$=\blacksquare$, $N_X^i = N_X^{i+1}$

Plan

Spécification et vérification : *proposition académique*

Spécification du protocole

- Formelle
- Réalisée après la conception !

  $\Rightarrow$ Le protocole est entièrement finalisé et connu
- En vue de prouver des propriétés classiques

  Essentiellement secret et authentification

Spécification et vérification : *proposition académique*

Spécification du protocole

- Formelle
- Réalisée après la conception !

  $\Rightarrow$ Le protocole est entièrement finalisé et connu
- En vue de prouver des propriétés classiques

  Essentiellement secret et authentification

Vérification automatique du protocole – Modèle de Dolev-Yao

- Détecter des attaques invalidant les propriétés
- Prouver leur absence pour un nombre non borné :
  - d'agents
  - de sessions entrelacées
  - d'opération élémentaires de l'intrus

Les langages de spécification de protocoles

Formats proches des notations à la « Alice et Bob »

- Casper [Lowe]
- Capsl [Mitchell]
- Eva, etc. [Jacquemard, Le Métayer]

Casper	Capsl	Eva
0. ->A: B 1. A->B: {na, A}{PK(B)} 2. B->A: {na, nb}{PK(A)} 3. A->B: {nb}{PK(B)}	A->B: {A,Na}pk(B); B->A: {Na,Nb}pk(A); A->B: {Nb}pk(B);	1. A->B: {Na, A}KPb 2. B->A: {Na, Nb}KPa 3. A->B: {Nb}KPb
Secret(A, na, [B]) Secret(B, nb, [A]) Agreement(A,B,[na,nb]) Agreement(B,A,[na,nb])	SECRET Na; SECRET Nb; PRECEDES A: B | Na; PRECEDES B: A | Nb;	Claim Agreement(A,B,Na,Na) Agreement(A,B,Nb,Nb)

Les langages de spécification de protocoles (II)

Formats inspirés des langages de processus

- ProVerif [Blanchet]
- Prouvé [Kremer, Laknech, Treinen]
- AVISPA (HLPSL) [Armando, et col.]

Alice	Bob
role alice (A,B: agent, ...) local State: nat, Na,Nb: text init State:= 0 transition 0. State=0 /\ RCV(start) =|> State':= 2 /\ Na' := new() /\ SND({Na'.A}_Kb) /\ secret(Na',na,{A,B}) 2. State=2 /\ RCV({Na.Nb'}_Ka) =|> State':= 4 /\ SND({Nb'}_Kb) end role	role bob(A, B: agent, ...) local State : nat, Na,Nb: text init State:= 1 transition 1. State= 1 /\ RCV({Na'.A}_Kb) =|> State':= 3 /\ Nb' := new() /\ SND({Na'.Nb'}_Ka) /\ secret(Nb',nb,{A,B}) 3. State= 3 /\ RCV({Nb'}_Kb) =|> State':= 5 /\ end role

Spécification et vérification : *pratiques industrielles*

Spécification du protocole

- partielle : entrelacée avec la conception
- Informelle : documents de conception/discussion/brevet
- Ad-hoc : environnement de fonctionnement spécifique
- Propriétés atypiques : Ex. « authentification anonyme »

Spécification et vérification : *pratiques industrielles*

Spécification du protocole

- partielle : entrelacée avec la conception
- Informelle : documents de conception/discussion/brevet
- Ad-hoc : environnement de fonctionnement spécifique
- Propriétés atypiques : Ex. « authentification anonyme »

Vérification du protocole

- Rare car spécifier formellement c'est déjà beaucoup !

Spécification et vérification : *pratiques industrielles*

Spécification du protocole

- partielle : entrelacée avec la conception
- Informelle : documents de conception/discussion/brevet
- Ad-hoc : environnement de fonctionnement spécifique
- Propriétés atypiques : Ex. « authentification anonyme »

Vérification du protocole

- Rare car spécifier formellement c'est déjà beaucoup !
- Réalisée manuellement par un expert extérieur
  $\Rightarrow$ long, coûteux et non-reproductible

Spécification et vérification : *pratiques industrielles*

Spécification du protocole

- partielle : entrelacée avec la conception
- Informelle : documents de conception/discussion/brevet
- Ad-hoc : environnement de fonctionnement spécifique
- Propriétés atypiques : Ex. « authentification anonyme »

Vérification du protocole

- Rare car spécifier formellement c'est déjà beaucoup !
- Réalisée manuellement par un expert extérieur
  $\Rightarrow$ long, coûteux et non-reproductible
- Vérifier formellement *a posteriori* ne suffit pas
- Vérifier *pendant* la conception serait idéal

Quels outils pour spécifier les protocoles industriels ?

Quels outils pour spécifier les protocoles industriels ?

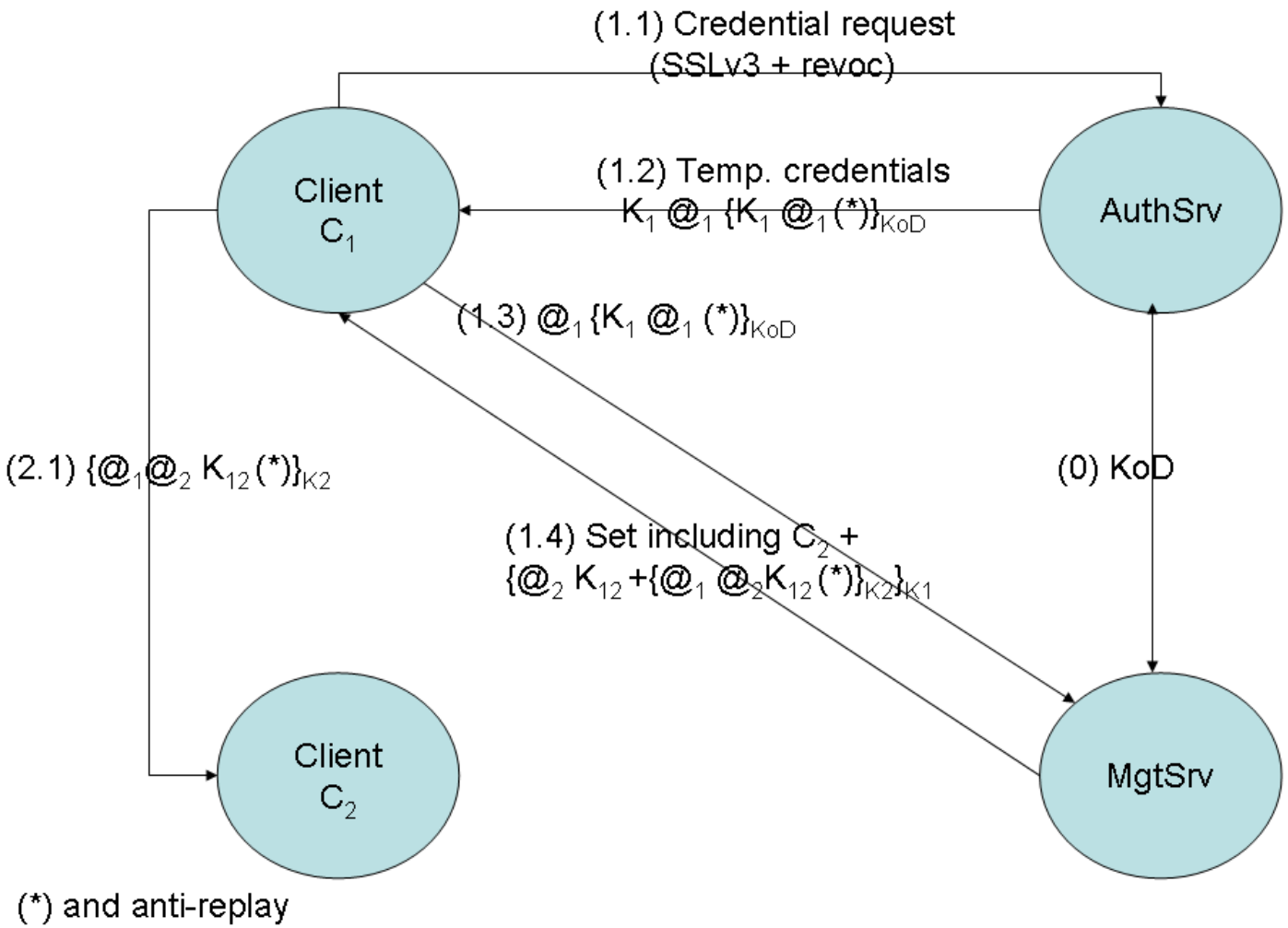
- Word pour les documents techniques/brevets

Quels outils pour spécifier les protocoles industriels ?

- Word pour les documents techniques/brevets
- Tableau blanc et Powerpoint pour la conception

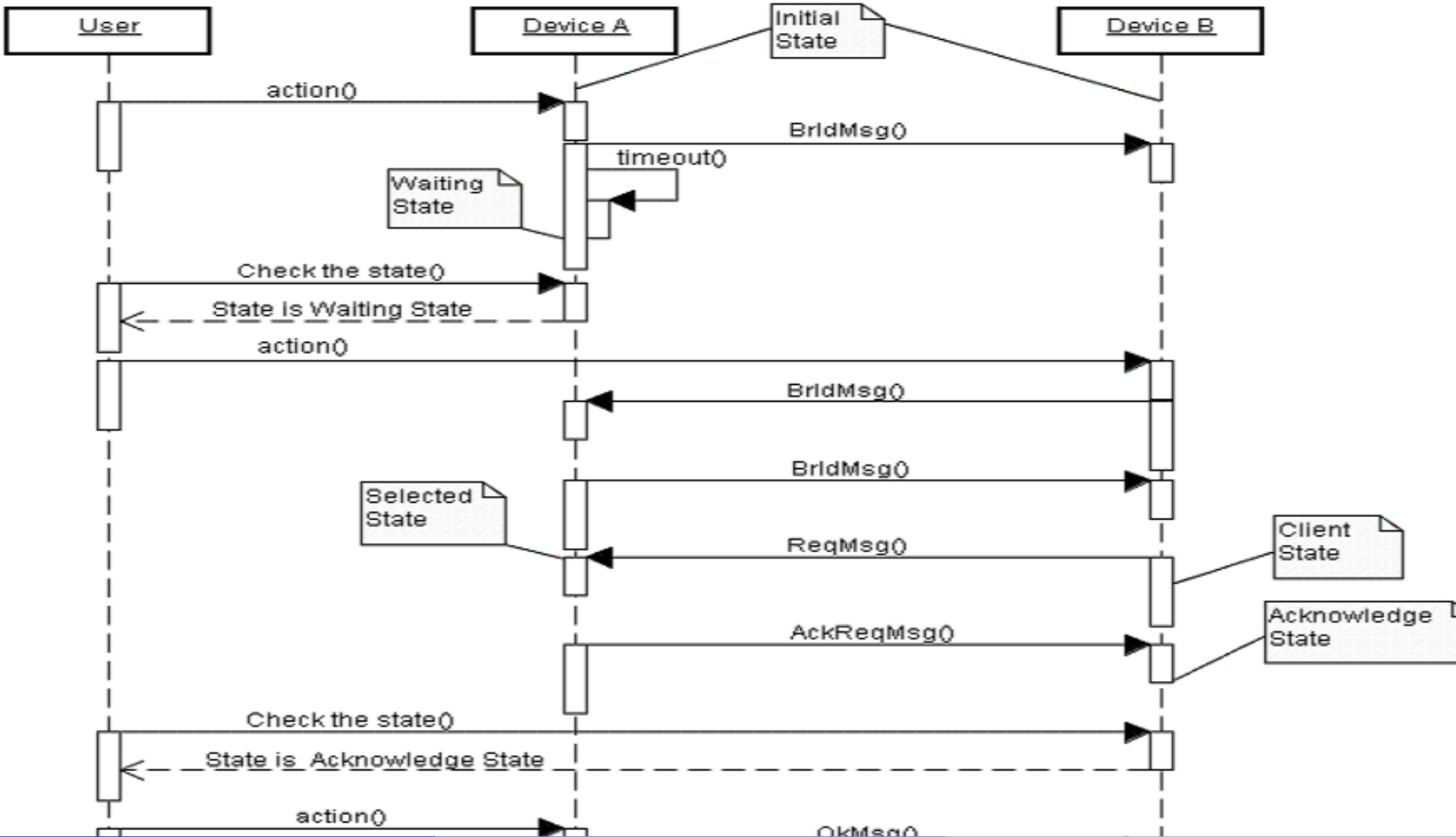
Quels outils pour spécifier les protocoles industriels ?

- Word pour les documents techniques/brevets
- Tableau blanc et Powerpoint pour la conception



Quels outils pour spécifier les protocoles industriels

- UML



Une première collaboration (2003) : SmartRight

Le protocole

- Un protocole de diffusion de contenu numérique
- Propriété : protection contre la copie/rediffusion

Une première collaboration (2003) : SmartRight

Le protocole

- Un protocole de diffusion de contenu numérique
- Propriété : protection contre la copie/rediffusion

La modélisation et la vérification

- Description du protocole dans le langage du prouveur ! (Timbuk)
- Détection « d'attaques » sur la formalisation initiale
- Correction et vérification de la propriété pour infinité de sessions

Une première collaboration (2003) : SmartRight

Le protocole

- Un protocole de diffusion de contenu numérique
- Propriété : protection contre la copie/rediffusion

La modélisation et la vérification

- Description du protocole dans le langage du prouveur ! (Timbuk)
- Détection « d'attaques » sur la formalisation initiale
- Correction et vérification de la propriété pour infinité de sessions

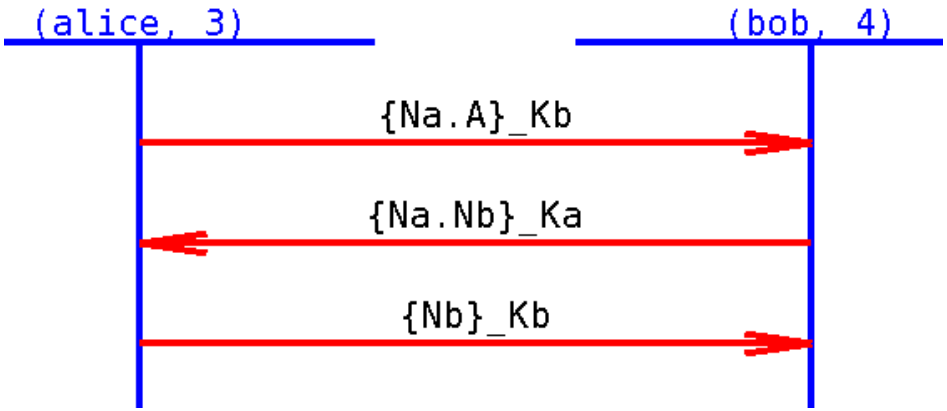
Des interrogations

- Collaboration sur la formalisation très difficile
- Interactions avec la modélisation quasi nulles
- $\Rightarrow$ A-t-on modélisé le protocole attendu ?

Concilier formalisme et intuition (2006) : SPAN

```
role alice (A,B: agent, ...)
local State: nat, Na,Nb: text
init State:= 0
transition
0. State=0 /\ RCV(start) =|>
   State':= 2 /\ Na' := new()
   /\ SND({Na'.A}_Kb)
   /\ secret(Na',na,{A,B})

2. State=2 /\ RCV({Na.Nb'}_Ka) =|>
   State':= 4 /\ SND({Nb'}_Kb)
end role
```



Développement de l'outil SPAN pour AVISPA
avec Yann Glouche

Les caractéristiques de AVISPA+SPAN

Choix de l'outil AVISPA

- Langage HLPSL assez accessible
- Vérification sur Dolev-Yao avec affaiblissement (*exp* et $\oplus$)
- Outil reconnu, maintenu et assez largement utilisé

Les caractéristiques de AVISPA+SPAN

Choix de l'outil AVISPA

- Langage HLPSL assez accessible
- Vérification sur Dolev-Yao avec affaiblissement (*exp* et $\oplus$)
- Outil reconnu, maintenu et assez largement utilisé

Apports de SPAN à AVISPA

- Construction de diagrammes de séquences
- Outil interactif d'aide à la construction d'attaques

  $\Rightarrow$ Construction d'attaques particulières non détectées par AVISPA
- Facilite la conception collaborative
  - Spécifications et traces en format texte
  - Partage électronique de scénarios/attaques

Démo

Plan

1. Spécification HLPSL de Diffie-Hellman
2. Vérification automatique par AVISPA
3. Animation par SPAN
4. Construction de l'attaque trouvée automatiquement
5. Construction de l'attaque de type « Man in the Middle »
6. Exemple sur un protocole de Thomson

Conclusion

- Début de convergence des intérêts académiques/industriels
  - Niveau de formalisme « presque acceptable » par un industriel

Conclusion

- Début de convergence des intérêts académiques/industriels
  - Niveau de formalisme « presque acceptable » par un industriel
  - Un format $\rightsquigarrow$ plusieurs utilisations :
    - ★ Cycle formalisation/simulation pour la mise au point
    - ★ Production de documents de spécification
    - ★ Recherche manuelle/automatique d'attaques
    - ★ Diffusion électronique de spécifications/scénarios

Conclusion

- Début de convergence des intérêts académiques/industriels
  - Niveau de formalisme « presque acceptable » par un industriel
  - Un format $\rightsquigarrow$ plusieurs utilisations :
    - ★ Cycle formalisation/simulation pour la mise au point
    - ★ Production de documents de spécification
    - ★ Recherche manuelle/automatique d'attaques
    - ★ Diffusion électronique de spécifications/scénarios
- Mais beaucoup reste à faire
  - Prise en compte de protocoles plus exotiques
    - ★ protocoles de groupes
    - ★ réseaux spécifiques
    - ★ canaux protégés

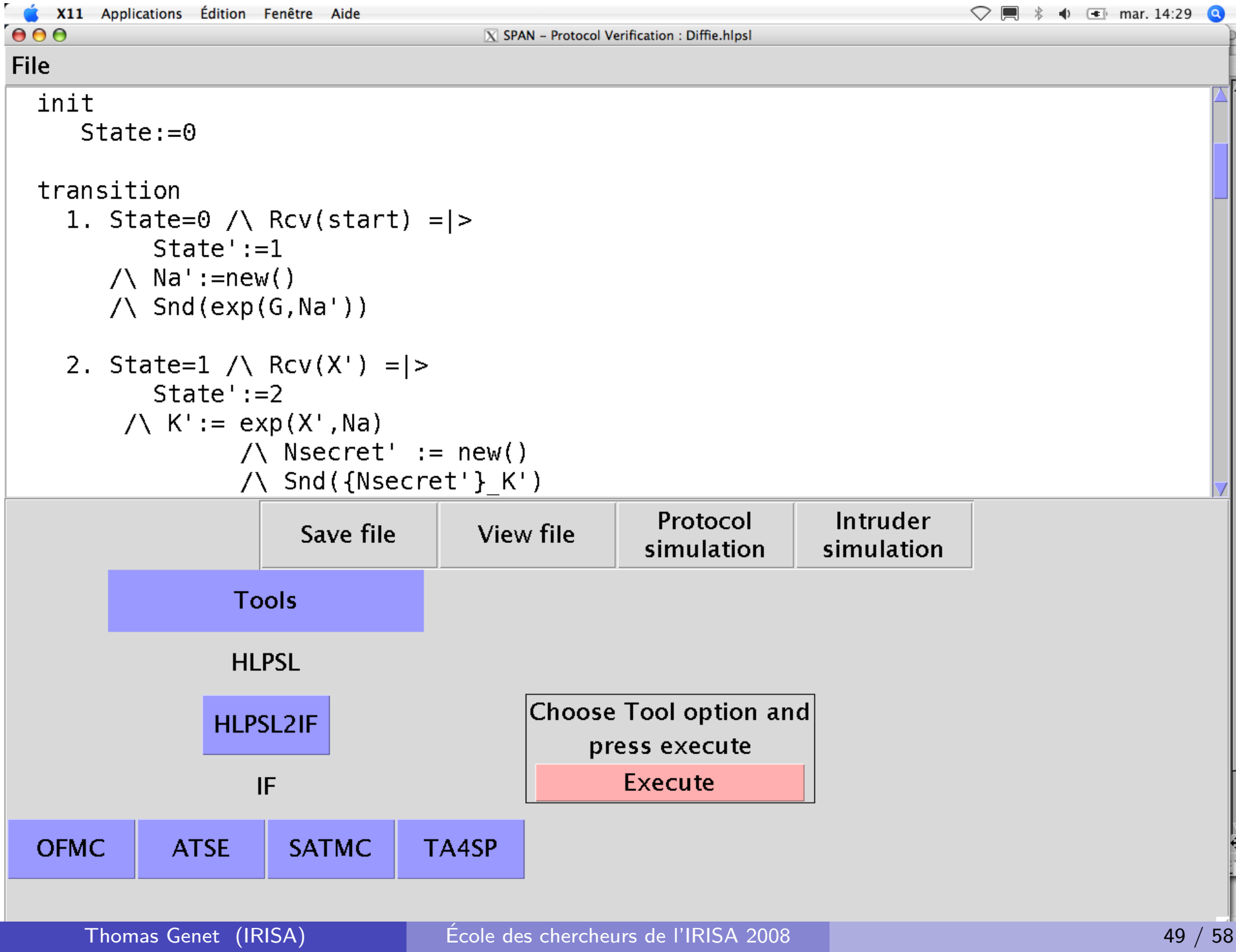
Conclusion

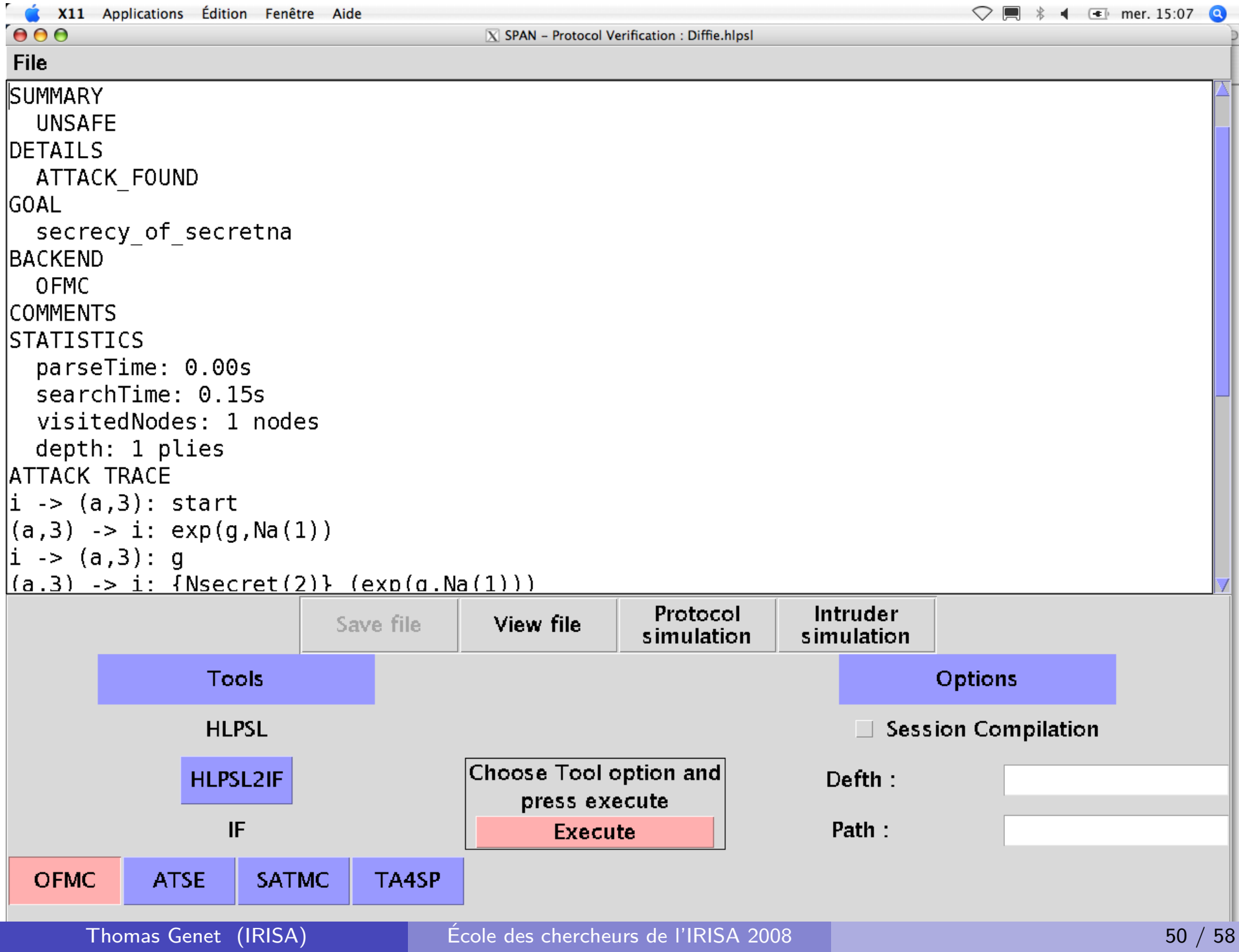
- Début de convergence des intérêts académiques/industriels
  - Niveau de formalisme « presque acceptable » par un industriel
  - Un format $\rightsquigarrow$ plusieurs utilisations :
    - ★ Cycle formalisation/simulation pour la mise au point
    - ★ Production de documents de spécification
    - ★ Recherche manuelle/automatique d'attaques
    - ★ Diffusion électronique de spécifications/scénarios
- Mais beaucoup reste à faire
  - Prise en compte de protocoles plus exotiques
    - ★ protocoles de groupes
    - ★ réseaux spécifiques
    - ★ canaux protégés
  - Prise en compte de propriétés plus riches
    - ★ authentification de groupes
    - ★ anonymat

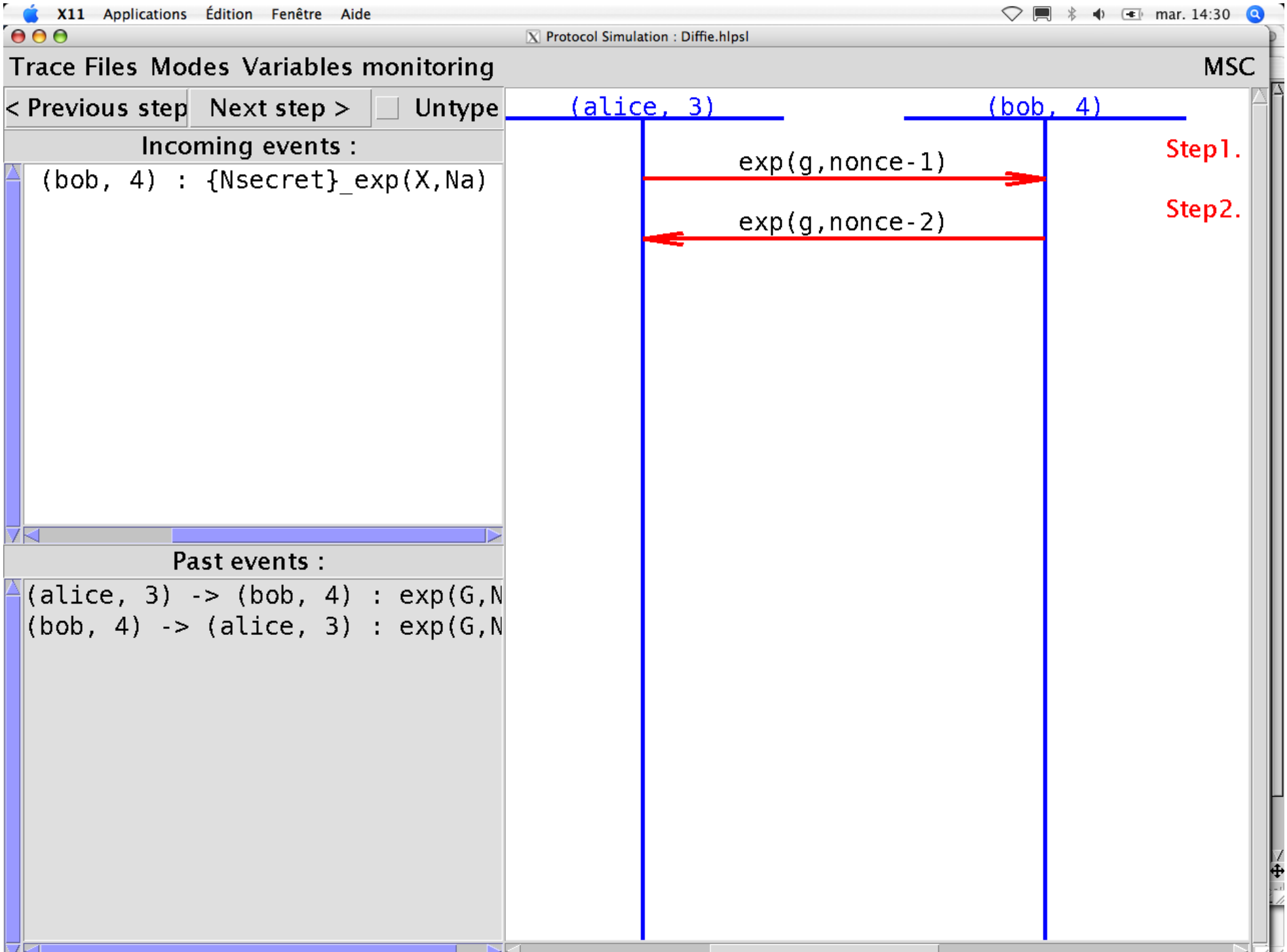
Liens vers les outils

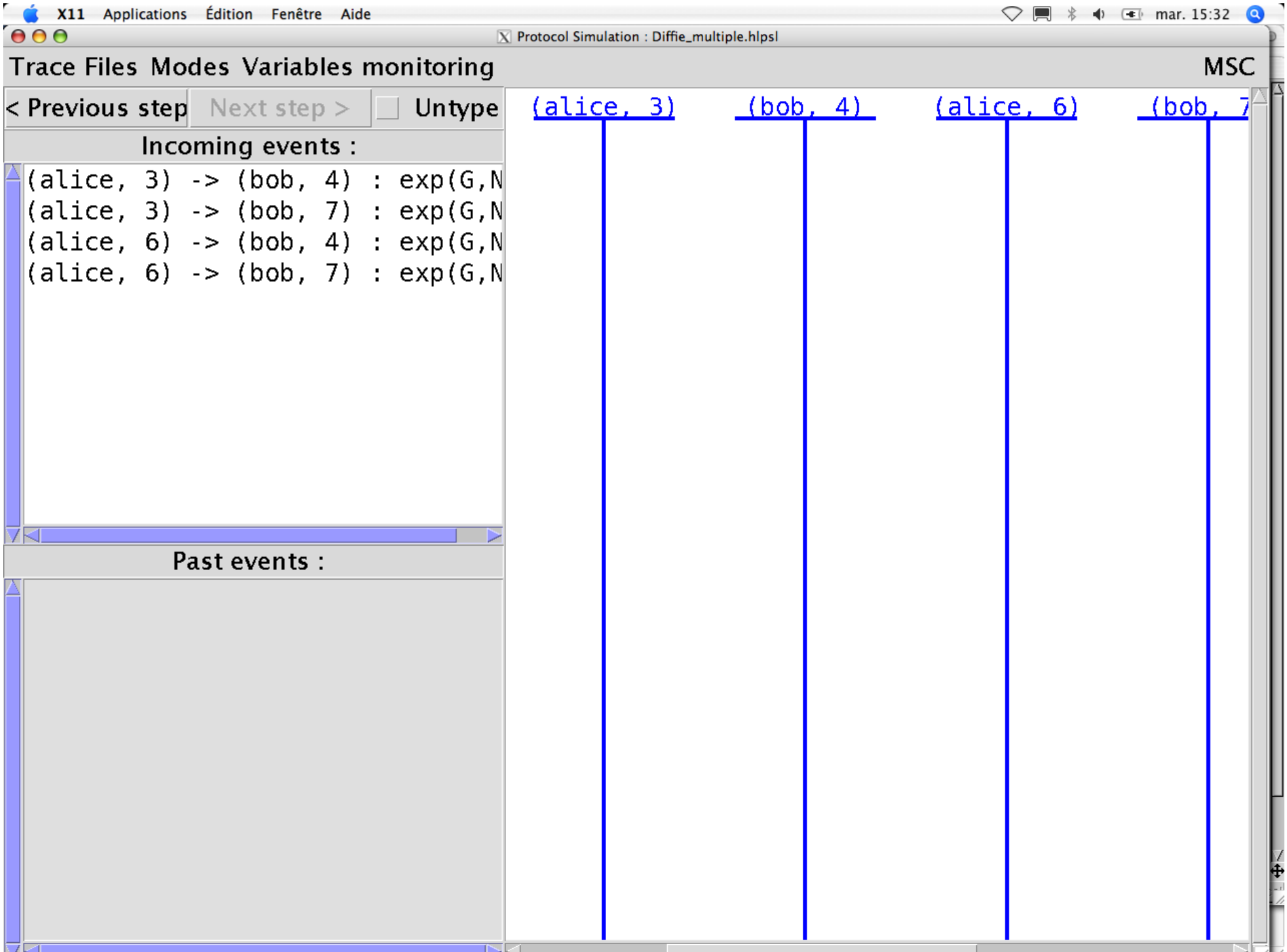
http://www.avispa-project.org/

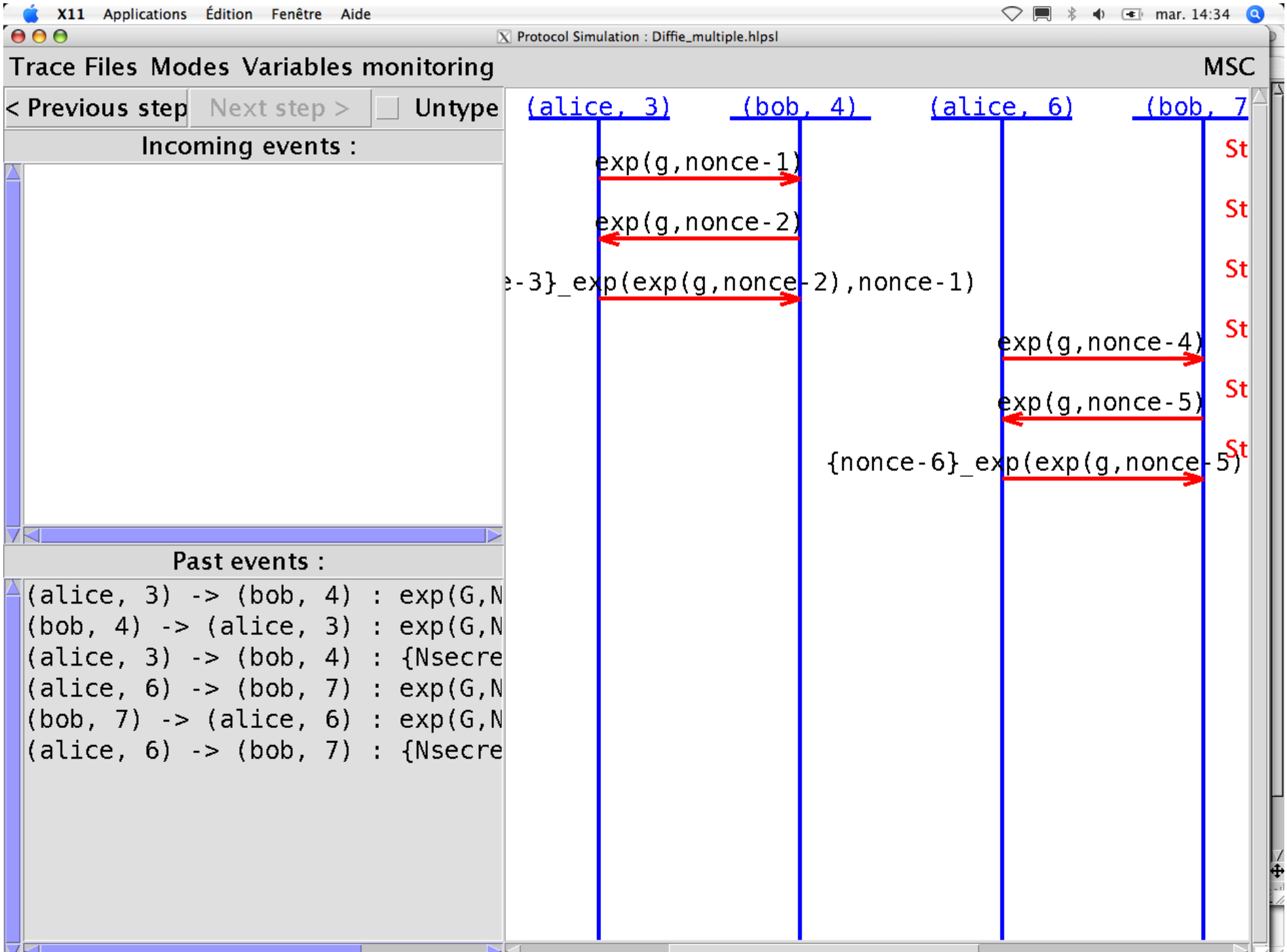
http://www.irisa.fr/lande/genet/span/


X11 Applications Édition Fenêtre Aide
mar. 14:29
SPAN – Protocol Verification : Diffie.hlpsl
File
init
State:=0
transition
1. State=0 /\ Rcv(start) =|>
State':=1
/\ Na':=new()
/\ Snd(exp(G,Na'))
2. State=1 /\ Rcv(X') =|>
State':=2
/\ K':= exp(X',Na)
/\ Nsecret' := new()
/\ Snd({Nsecret'}_K')
Save file
View file
Protocol simulation
Intruder simulation
Tools
HLPSL
HLPSL2IF
IF
Choose Tool option and press execute
Execute
OFMC
ATSE
SATMC
TA4SP


X11 Applications Édition Fenêtre Aide
mer. 15:07
SPAN – Protocol Verification : Diffie.hlpsl
File
SUMMARY
UNSAFE
DETAILS
ATTACK_FOUND
GOAL
secrecy_of_secretna
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.15s
visitedNodes: 1 nodes
depth: 1 plies
ATTACK TRACE
i -> (a,3): start
(a,3) -> i: exp(g,Na(1))
i -> (a,3): g
(a,3) -> i: {Nsecret(2)} (exp(g,Na(1)))
Save file
View file
Protocol simulation
Intruder simulation
Tools
Options
HLPSL
Session Compilation
HLPSL2IF
Choose Tool option and press execute
Execute
Defth :
IF
Path :
OFMC
ATSE
SATMC
TA4SP


X11 Applications Édition Fenêtre Aide
mar. 14:30
Protocol Simulation : Diffie.hlpsl
Trace Files Modes Variables monitoring
MSC
< Previous step
Next step >
Untype
Incoming events :
(bob, 4) : {Nsecret}_exp(X,Na)
Past events :
(alice, 3) -> (bob, 4) : exp(G,N
(bob, 4) -> (alice, 3) : exp(G,N
(alice, 3)
(bob, 4)
exp(g,nonce-1)
exp(g,nonce-2)
Step1.
Step2.


X11 Applications Édition Fenêtre Aide
mar. 15:32
Protocol Simulation : Diffie_multiple.hlpsl
Trace Files Modes Variables monitoring
MSC
< Previous step
Next step >
Untype
Incoming events :
(alice, 3) -> (bob, 4) : exp(G,N
(alice, 3) -> (bob, 7) : exp(G,N
(alice, 6) -> (bob, 4) : exp(G,N
(alice, 6) -> (bob, 7) : exp(G,N
Past events :
(alice, 3)
(bob, 4)
(alice, 6)
(bob, 7


X11 Applications Édition Fenêtre Aide
mar. 14:34
Protocol Simulation : Diffie_multiple.hlpsl
Trace Files Modes Variables monitoring
MSC
< Previous step
Next step >
Untype
Incoming events :
Past events :
(alice, 3) -> (bob, 4) : exp(G,N
(bob, 4) -> (alice, 3) : exp(G,N
(alice, 3) -> (bob, 4) : {Nsecre
(alice, 6) -> (bob, 7) : exp(G,N
(bob, 7) -> (alice, 6) : exp(G,N
(alice, 6) -> (bob, 7) : {Nsecre
(alice, 3)
(bob, 4)
(alice, 6)
(bob, 7
exp(g,nonce-1)
exp(g,nonce-2)
-3}_exp(exp(g,nonce-2),nonce-1)
exp(g,nonce-4)
exp(g,nonce-5)
{nonce-6}_exp(exp(g,nonce-5)
St

Protocol Simulation : Diffie_multiple.hlpsl

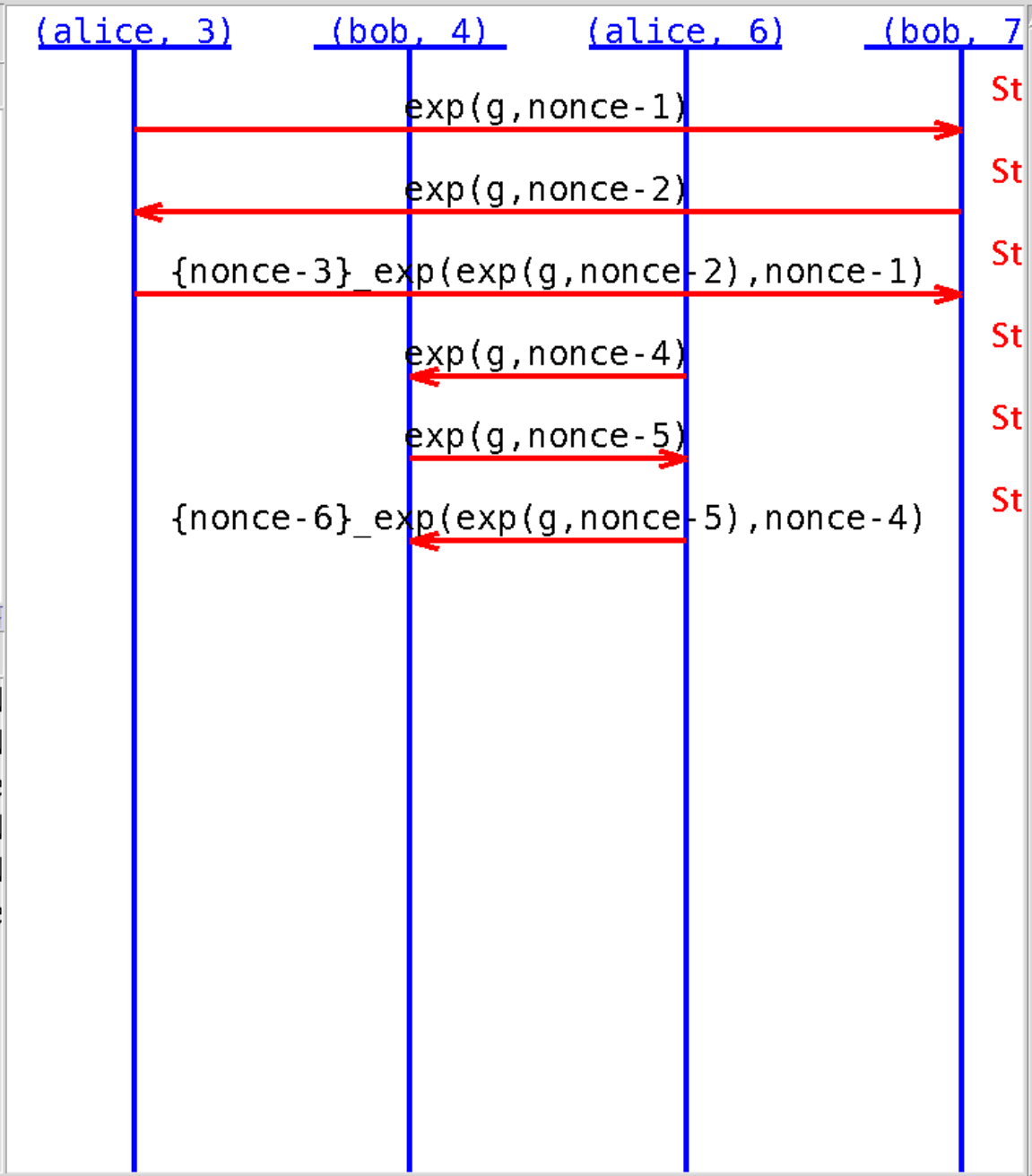
Trace Files Modes Variables monitoring MSC

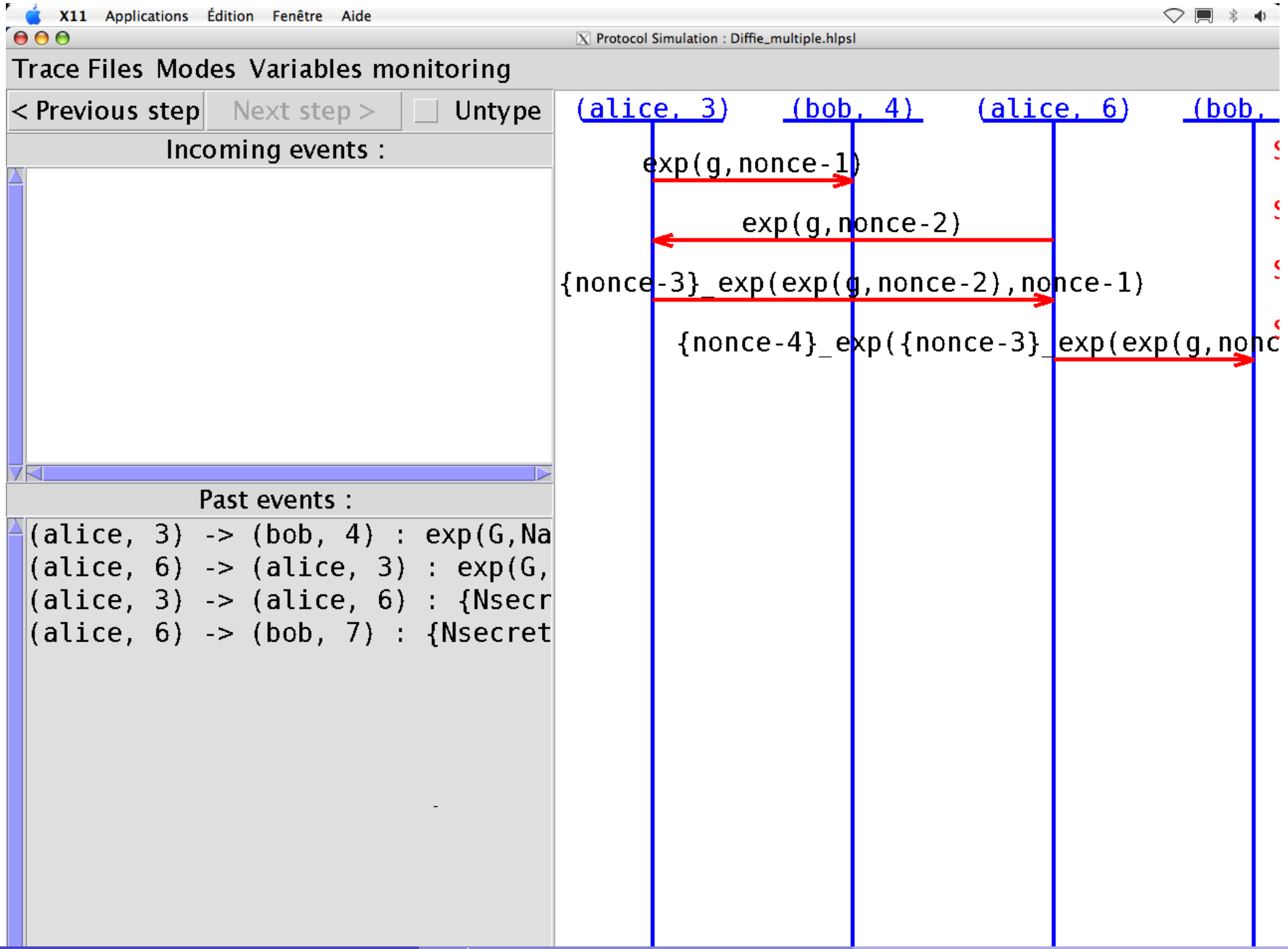
< Previous step | Next step > | Untype

Incoming events :

Past events :

```
(alice, 3) -> (bob, 7) : exp(G,N
(bob, 7) -> (alice, 3) : exp(G,N
(alice, 3) -> (bob, 7) : {Nsecre
(alice, 6) -> (bob, 4) : exp(G,N
(bob, 4) -> (alice, 6) : exp(G,N
(alice, 6) -> (bob, 4) : {Nsecre
```




X11 Applications Édition Fenêtre Aide
Protocol Simulation : Diffie_multiple.hlpsl
Trace Files Modes Variables monitoring
< Previous step
Next step >
Untype
Incoming events :
Past events :
(alice, 3) -> (bob, 4) : exp(G,Na
(alice, 6) -> (alice, 3) : exp(G,
(alice, 3) -> (alice, 6) : {Nsecr
(alice, 6) -> (bob, 7) : {Nsecret
(alice, 3)
(bob, 4)
(alice, 6)
(bob,
exp(g,nonce-1)
exp(g,nonce-2)
{nonce-3}_exp(exp(g,nonce-2),nonce-1)
{nonce-4}_exp({nonce-3}_exp(exp(g,nonc

X11 Applications Édition Fenêtre Aide mer. 15:16

Protocol Simulation : Diffie.hlpsl

Trace Files Modes Variables monitoring MSC

< Previous step | Next step > | Untype

Incoming events :

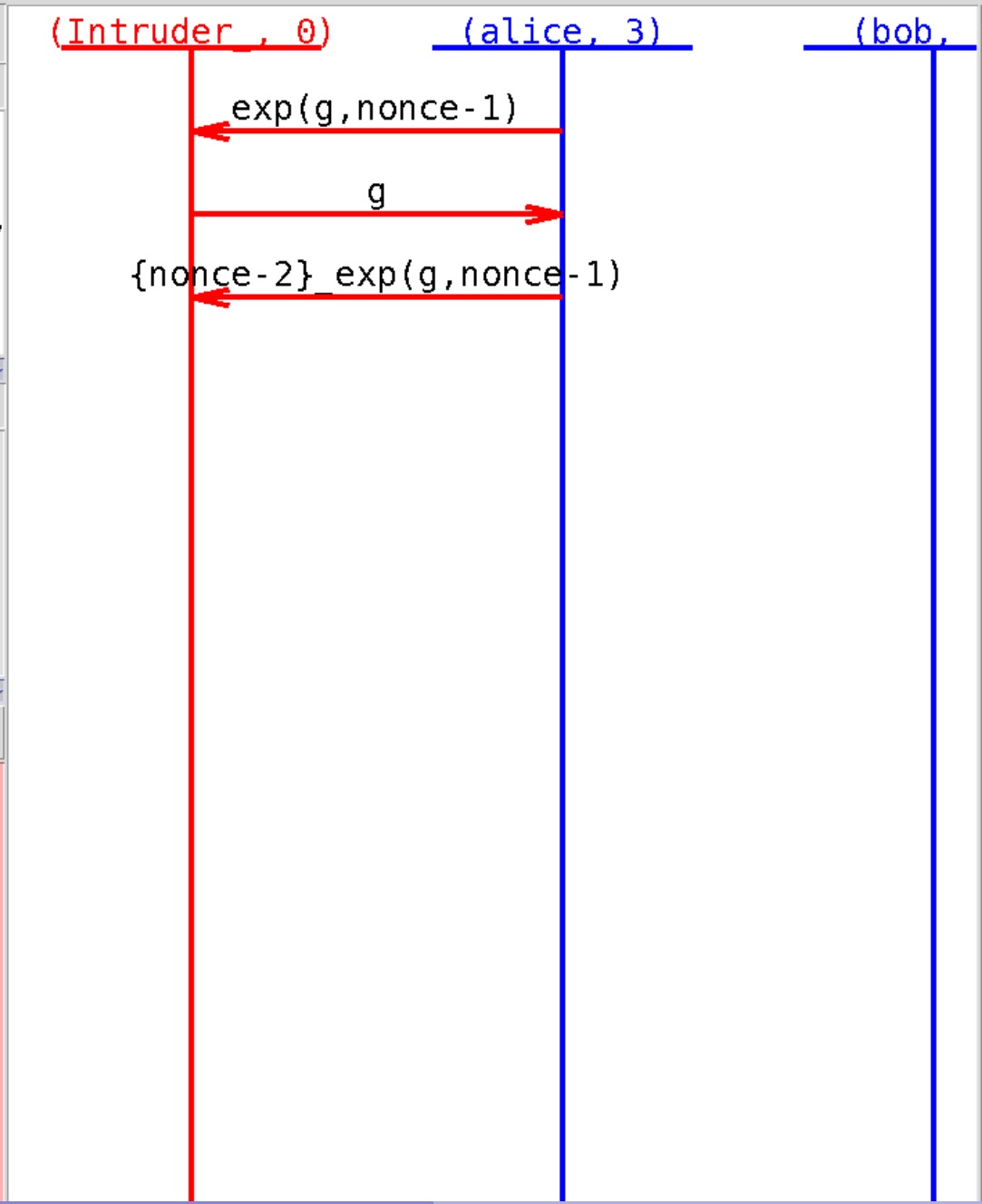
```
(Intruder_, 0) -> (bob, 4) : a
(Intruder_, 0) -> (bob, 4) : b
(Intruder_, 0) -> (bob, 4) : exp(g,
(Intruder_, 0) -> (bob, 4) : i
(Intruder_, 0) -> (bob, 4) : msg
(Intruder_, 0) -> (bob, 4) : ni
```

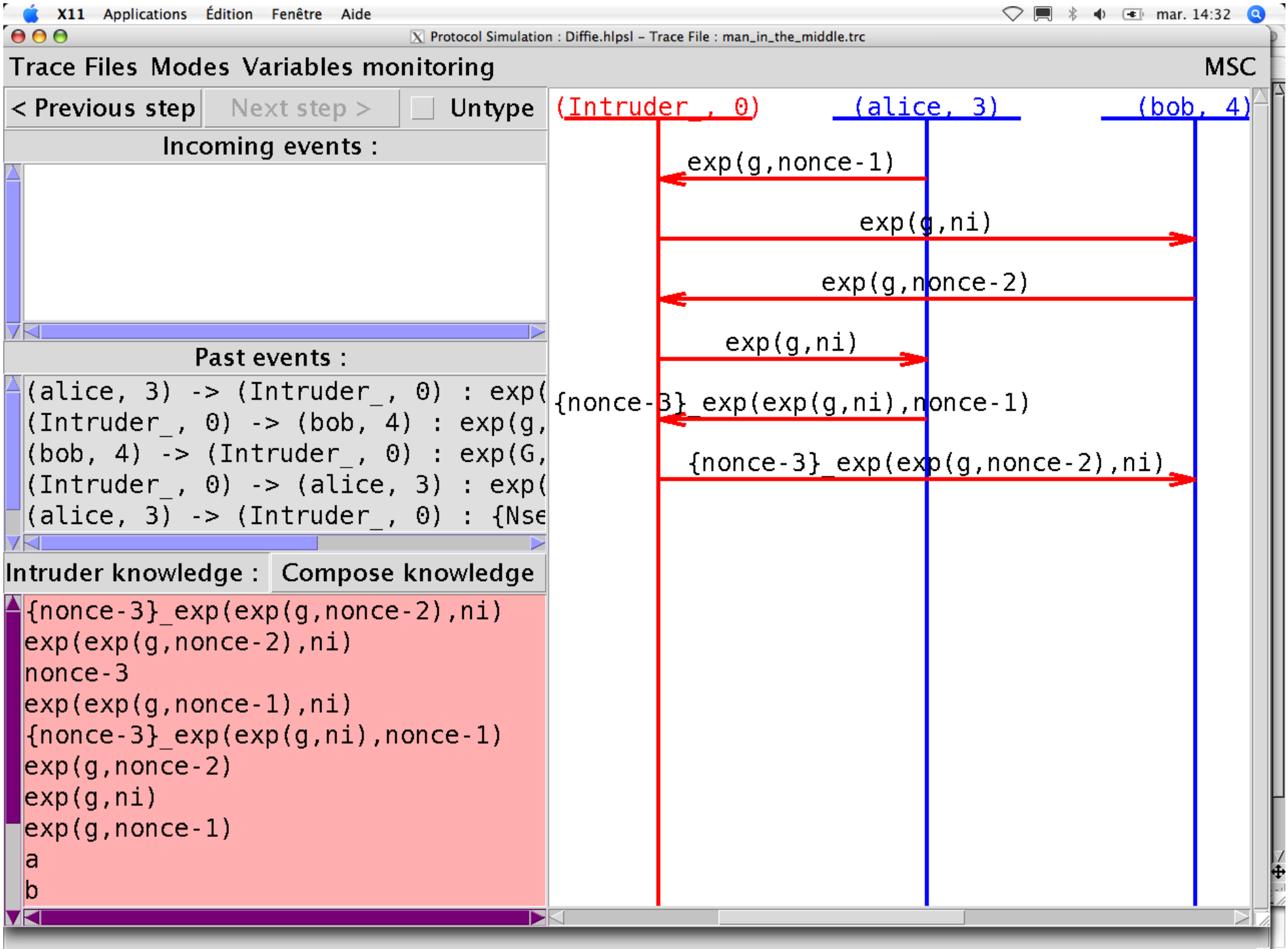
Past events :

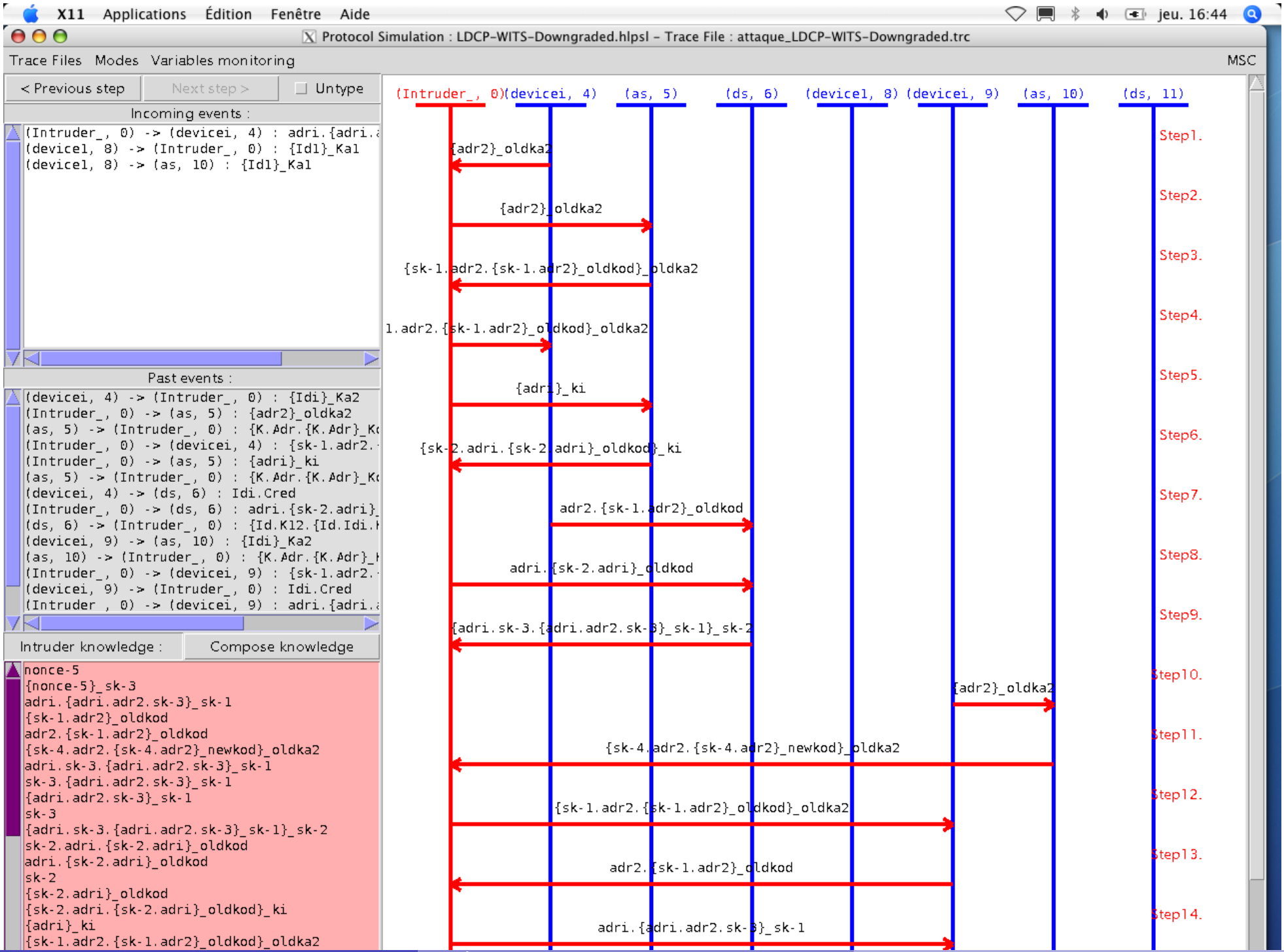
```
ntruder_, 0) : exp(G,Na)
> (alice, 3) : g
ntruder_, 0) : {Nsecret}_exp(X,Na)
```

Intruder knowledge : | Compose knowledge

```
nonce-2
{nonce-2}_exp(g,nonce-1)
exp(g,nonce-1)
a
b
g
ni
i
msg
```




X11 Applications Édition Fenêtre Aide
mar. 14:32
Protocol Simulation : Diffie.hlpsl – Trace File : man_in_the_middle.trc
Trace Files Modes Variables monitoring
MSC
< Previous step
Next step >
Untype
Incoming events :
Past events :
(alice, 3) -> (Intruder_, 0) : exp(
(Intruder_, 0) -> (bob, 4) : exp(g,
(bob, 4) -> (Intruder_, 0) : exp(G,
(Intruder_, 0) -> (alice, 3) : exp(
(alice, 3) -> (Intruder_, 0) : {Nse
Intruder knowledge :
Compose knowledge
{nonce-3}_exp(exp(g,nonce-2),ni)
exp(exp(g,nonce-2),ni)
nonce-3
exp(exp(g,nonce-1),ni)
{nonce-3}_exp(exp(g,ni),nonce-1)
exp(g,nonce-2)
exp(g,ni)
exp(g,nonce-1)
a
b
(Intruder_, 0)
(alice, 3)
(bob, 4)
exp(g,nonce-1)
exp(g,ni)
exp(g,nonce-2)
exp(g,ni)
{nonce-3}_exp(exp(g,ni),nonce-1)
{nonce-3}_exp(exp(g,nonce-2),ni)


X11 Applications Édition Fenêtre Aide
jeu. 16:44
Protocol Simulation : LDCP-WITS-Downgraded.hlpsl – Trace File : attaque_LDCP-WITS-Downgraded.trc
Trace Files Modes Variables monitoring
MSC
< Previous step
Next step >
Untype
Incoming events :
(Intruder_, 0) -> (devicei, 4) : adri.{adri.
(device1, 8) -> (Intruder_, 0) : {Id1}_Ka1
(device1, 8) -> (as, 10) : {Id1}_Ka1
Past events :
(devicei, 4) -> (Intruder_, 0) : {Idi}_Ka2
(Intruder_, 0) -> (as, 5) : {adr2}_oldka2
(as, 5) -> (Intruder_, 0) : {K.Adr.{K.Adr}_K
(Intruder_, 0) -> (devicei, 4) : {sk-1.adr2.
(Intruder_, 0) -> (as, 5) : {adri}_ki
(as, 5) -> (Intruder_, 0) : {K.Adr.{K.Adr}_K
(devicei, 4) -> (ds, 6) : Idi.Cred
(Intruder_, 0) -> (ds, 6) : adri.{sk-2.adri}_
(ds, 6) -> (Intruder_, 0) : {Id.K12.{Id.Idi.
(devicei, 9) -> (as, 10) : {Idi}_Ka2
(as, 10) -> (Intruder_, 0) : {K.Adr.{K.Adr}_
(Intruder_, 0) -> (devicei, 9) : {sk-1.adr2.
(devicei, 9) -> (Intruder_, 0) : Idi.Cred
(Intruder , 0) -> (devicei, 9) : adri.{adri.
Intruder knowledge :
Compose knowledge
nonce-5
{nonce-5}_sk-3
adri.{adri.adr2.sk-3}_sk-1
{sk-1.adr2}_oldkod
adr2.{sk-1.adr2}_oldkod
{sk-4.adr2.{sk-4.adr2}_newkod}_oldka2
adri.sk-3.{adri.adr2.sk-3}_sk-1
sk-3.{adri.adr2.sk-3}_sk-1
{adri.adr2.sk-3}_sk-1
sk-3
{adri.sk-3.{adri.adr2.sk-3}_sk-1}_sk-2
sk-2.adri.{sk-2.adri}_oldkod
adri.{sk-2.adri}_oldkod
sk-2
{sk-2.adri}_oldkod
{sk-2.adri.{sk-2.adri}_oldkod}_ki
{adri}_ki
{sk-1.adr2.{sk-1.adr2}_oldkod}_oldka2
(Intruder_, 0)
(devicei, 4)
(as, 5)
(ds, 6)
(device1, 8)
(devicei, 9)
(as, 10)
(ds, 11)
Step1.
{adr2}_oldka2
Step2.
{adr2}_oldka2
Step3.
{sk-1.adr2.{sk-1.adr2}_oldkod}_oldka2
Step4.
1.adr2.{sk-1.adr2}_oldkod}_oldka2
Step5.
{adri}_ki
Step6.
{sk-2.adri.{sk-2.adri}_oldkod}_ki
Step7.
adr2.{sk-1.adr2}_oldkod
Step8.
adri.{sk-2.adri}_oldkod
Step9.
{adri.sk-3.{adri.adr2.sk-3}_sk-1}_sk-2
Step10.
{adr2}_oldka2
Step11.
{sk-4.adr2.{sk-4.adr2}_newkod}_oldka2
Step12.
{sk-1.adr2.{sk-1.adr2}_oldkod}_oldka2
Step13.
adr2.{sk-1.adr2}_oldkod
Step14.
adri.{adri.adr2.sk-3}_sk-1