Contributions à la vérification automatique de protocoles de groupes

Najah Chridi

Encadrants : Laurent Vigneron, Michaël Rusinowitch

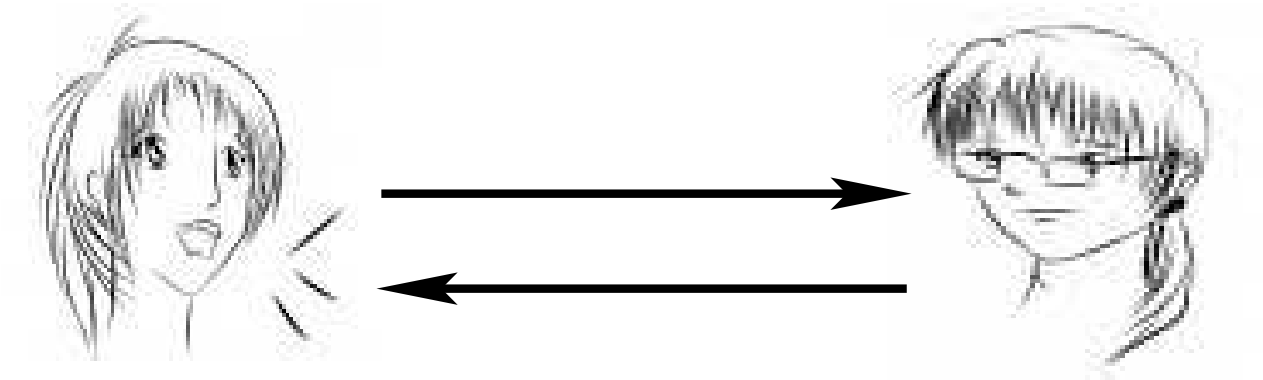
Soutenance de thèse : 11 Septembre 2009

Plan

Protocoles Cryptographiques

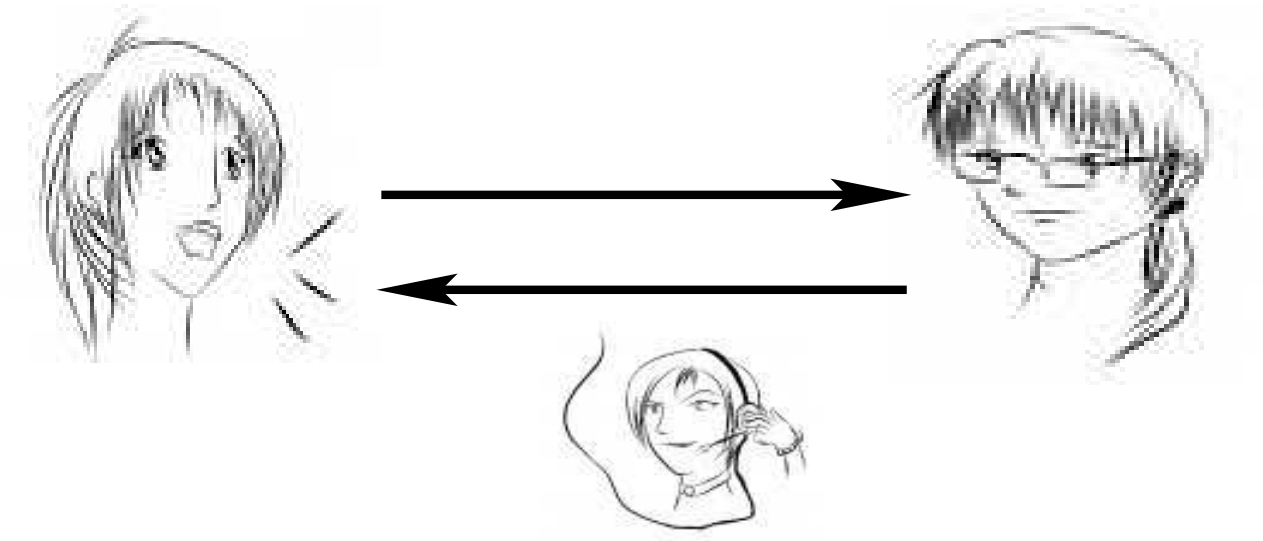
Protocoles Cryptographiques

- Un protocole cryptographique définit un échange de messages sur un réseau ;



Protocoles Cryptographiques

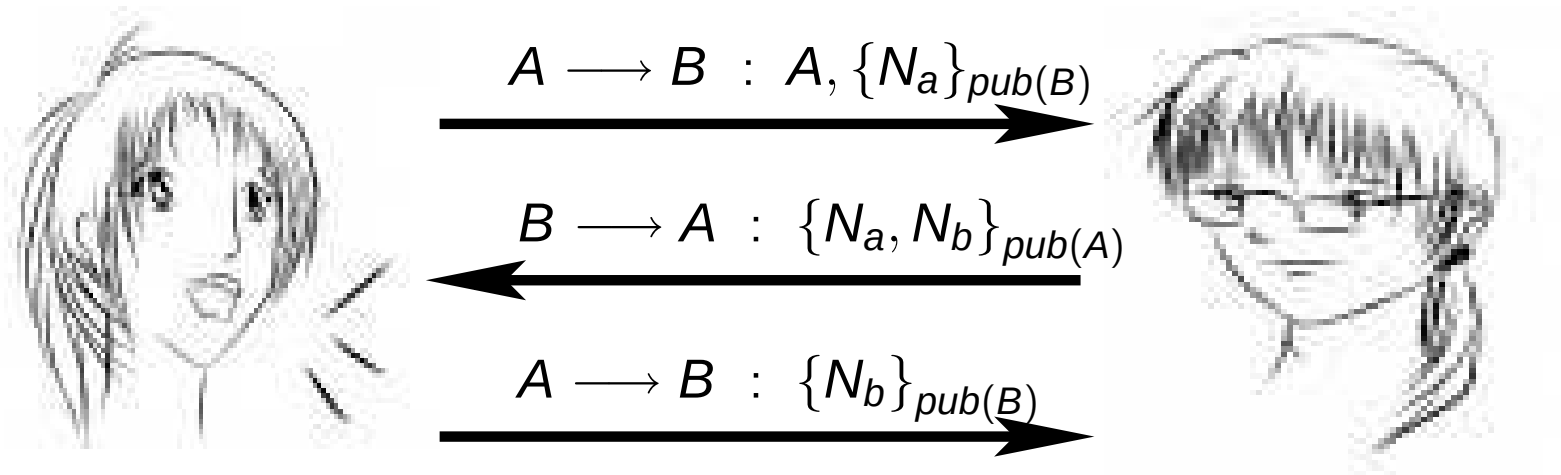
- Un protocole cryptographique définit un échange de messages sur un réseau ;



  - présence d'un intrus ;
    - Passif : écoute ;
    - Actif : écoute, intercepte, bloque, (re)joue les messages.

Protocoles Cryptographiques

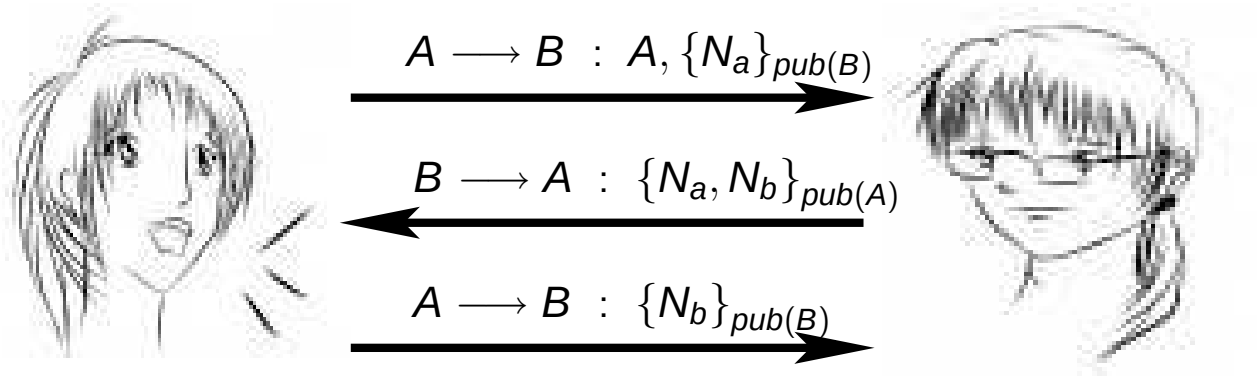
- Un protocole cryptographique définit un échange de messages sur un réseau ;



- présence d'un intrus ;
- des primitives cryptographiques ;
  - Concaténation ;
  - Chiffrement : symétrique, asymétrique ;

Protocoles Cryptographiques

- Un protocole cryptographique définit un échange de messages sur un réseau ;



  - présence d'un intrus ;
  - des primitives cryptographiques ;
  - des propriétés de sécurité ;
    - Secret, authentification ;
    - N_b secret entre A et B ?
    - Quand B reçoit $\{N_b\}_{pub(B)}$, ce message provient-il réellement de A ?

Vérification Symbolique des protocoles

- Termes symboliques : m, $\langle m, m' \rangle$, $\{m\}_{pk(A)}$, $\{m\}_k$, ...

Vérification Symbolique des protocoles

- Termes symboliques : m, $\langle m, m' \rangle$, $\{m\}_{pk(A)}$, $\{m\}_k$, ...
- Modèle de l'intrus : Modèle de Dolev-Yao [1981].

Il peut
espionner, enregistrer, modifier, répondre,
initier des sessions parallèles
(Intrus=réseau)

Il ne peut pas
déchiffer sans la clef de déchiffrement,
créer des messages chiffrés sans le message et la clef de chiffrement
(Hypothèse de chiffrement parfait)

Vérification Symbolique des protocoles

- Termes symboliques : m, $\langle m, m' \rangle$, $\{m\}_{pk(A)}$, $\{m\}_k$, ...
- Modèle de l'intrus : Modèle de Dolev-Yao [1981].
- Plusieurs études pour vérifier les protocoles classiques ;
  - propriétés d'atteignabilité : secret, authentification ;
  - en général non-décidable ;
  - classes restrictives : e.g. borner le nombre de sessions ;

Vérification Symbolique des protocoles

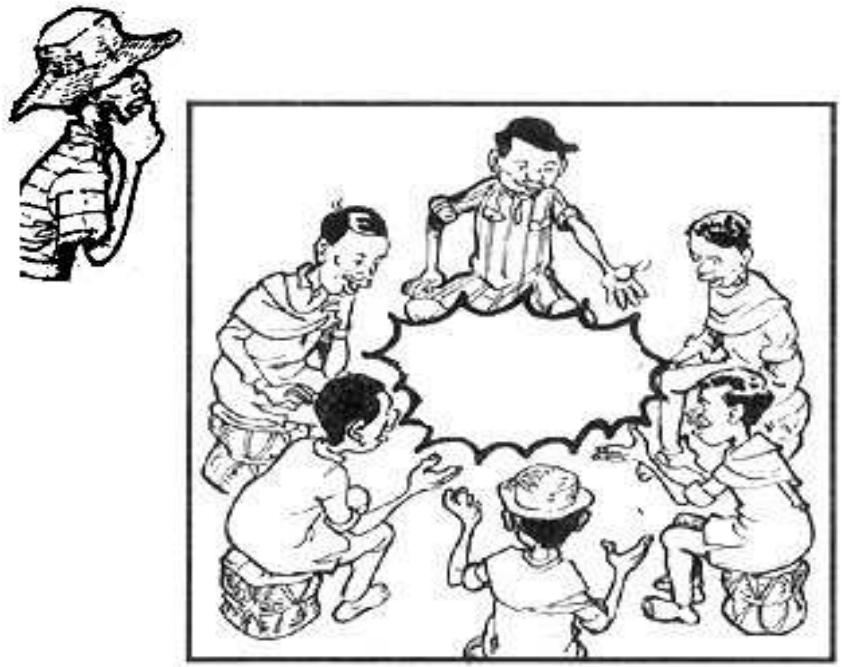
- Termes symboliques : m, $\langle m, m' \rangle$, $\{m\}_{pk(A)}$, $\{m\}_k$, ...
- Modèle de l'intrus : Modèle de Dolev-Yao [1981].
- Plusieurs études pour vérifier les protocoles classiques ;
  - propriétés d'atteignabilité : secret, authentification ;
  - en général non-décidable ;
  - classes restrictives : e.g. borner le nombre de sessions ;

Classe de protocoles et de propriétés plus difficiles : protocoles de groupes.

Plan

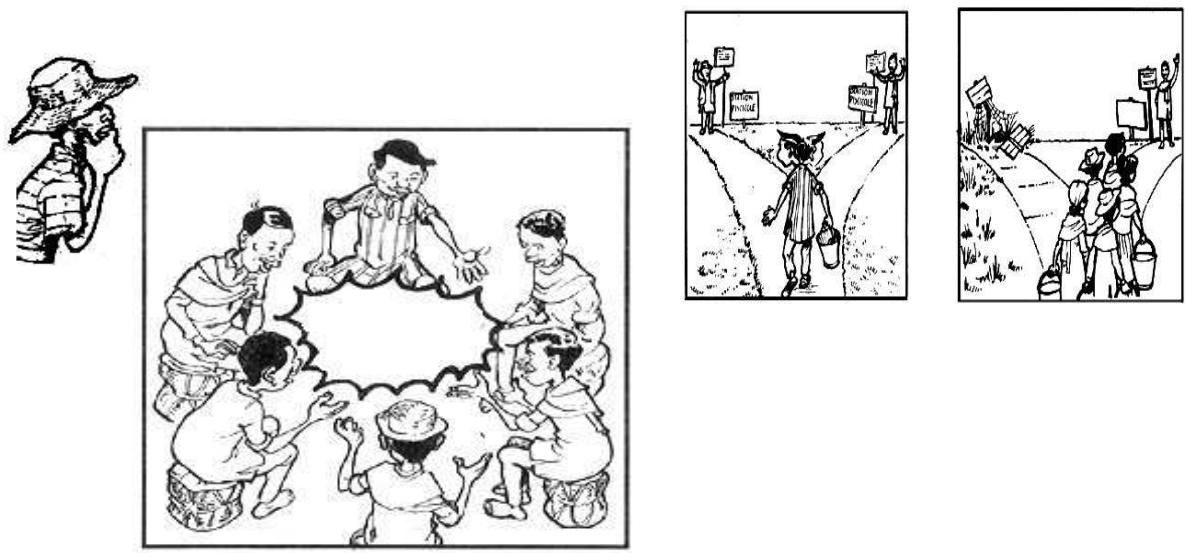
Protocoles de groupes

- Communications de groupe omniprésentes dans la plupart des applications distribuées :
  - vidéo conférences, jeux de groupe intéractifs, télévision à la demande, . . .



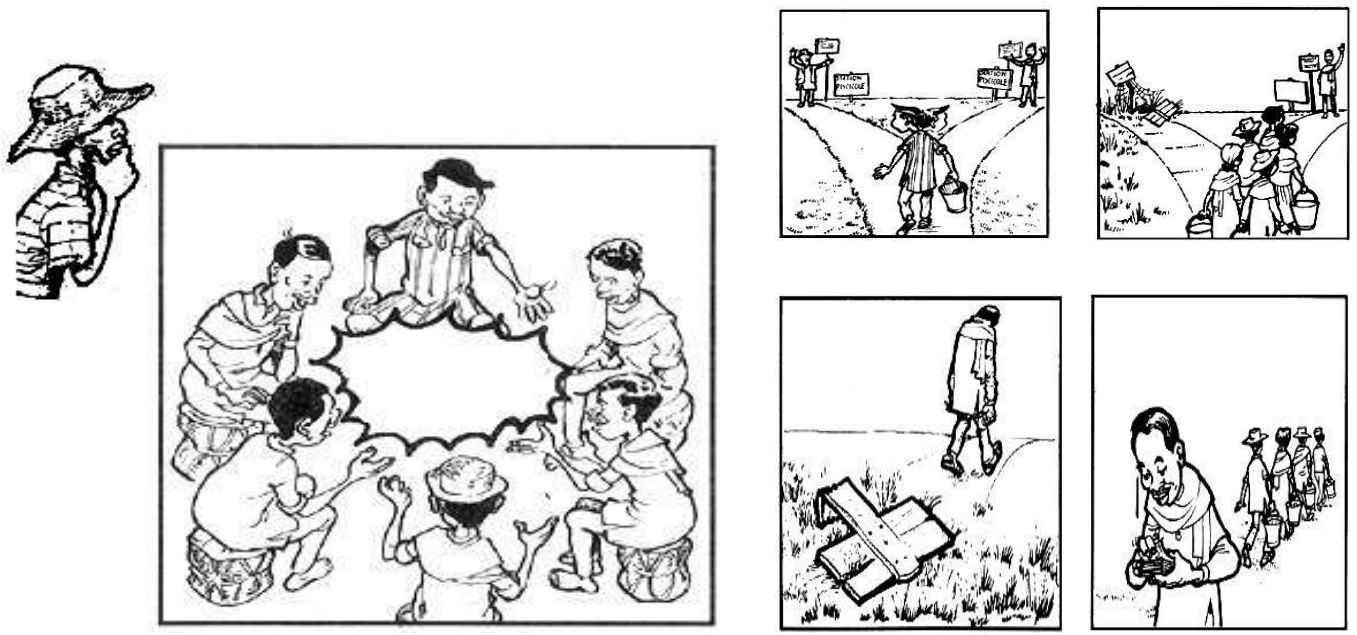
Protocoles de groupes

- Communications de groupe omniprésentes dans la plupart des applications distribuées :
  - vidéo conférences, jeux de groupe intéractifs, télévision à la demande, . . .



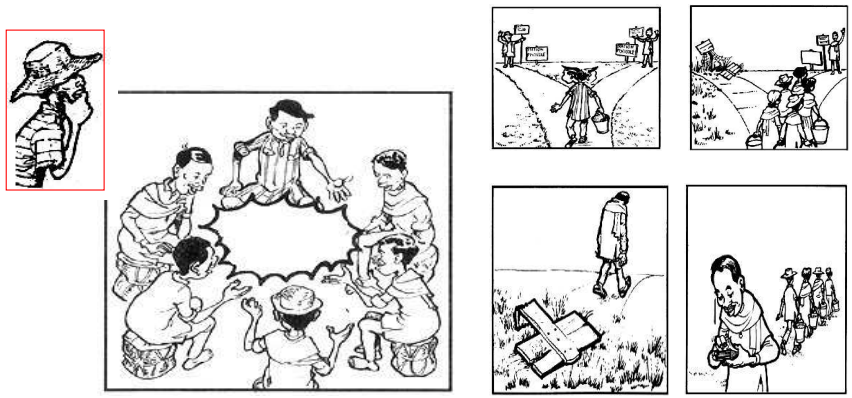
Protocoles de groupes

- Communications de groupe omniprésentes dans la plupart des applications distribuées :
  - vidéo conférences, jeux de groupe intéractifs, télévision à la demande, . . .



Protocoles de groupes

- Communications de groupe omniprésentes dans la plupart des applications distribuées :
  - vidéo conférences, jeux de groupe intéractifs, télévision à la demande, . . .



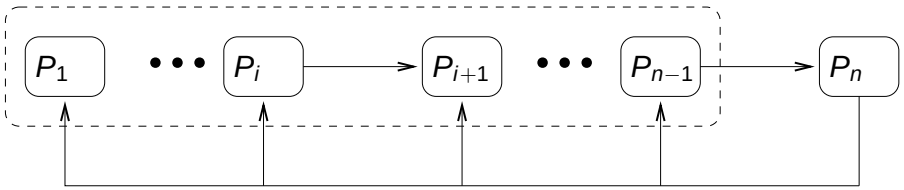
$\Rightarrow$ Gestion de clefs : préserver la sécurité de groupe même en présence d'un intrus.
  - mise en place d'une clef de groupe permettant de sécuriser les communications de groupe ;
  - mise à jour de cette clef quand c'est nécessaire.

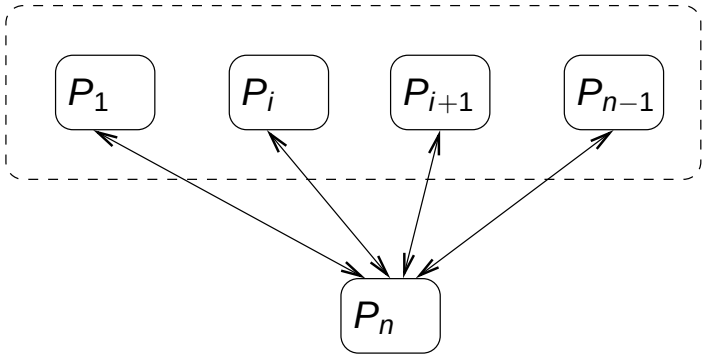
Gestion de clefs

- Gestion centralisée :
  - Une seule entité choisit la clef de groupe et la transfère aux autres membres du groupe (ex : TTP) ;
  -  Cible priviligiée pour d'éventuelles attaques.
- Gestion distribuée (GKAP) :
  - La clef de groupe se base sur les contributions de tous les membres du groupe ;
  - Idéalement, aucun des participants n'est capable de prévoir la valeur de cette clef ;

Gestion de clefs

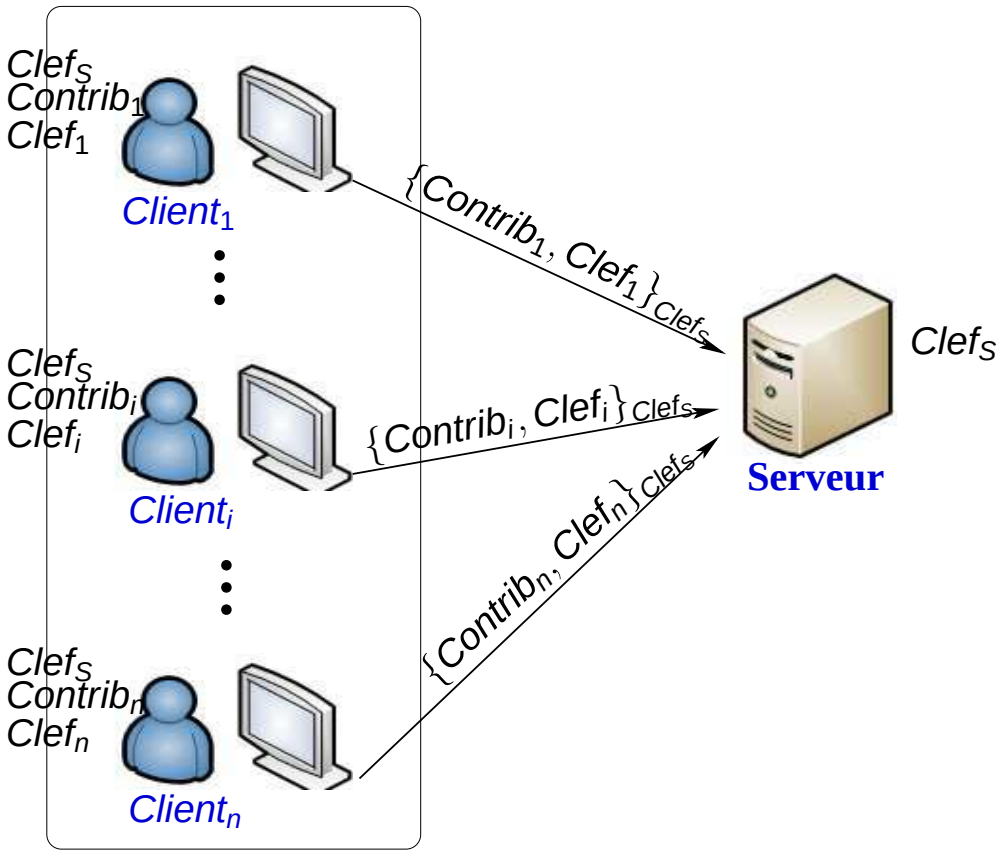
- Gestion centralisée ;
- Gestion distribuée (GKAP) :
  - La clef de groupe se base sur les contributions de tous les membres du groupe ;
  - Principalement deux architectures :





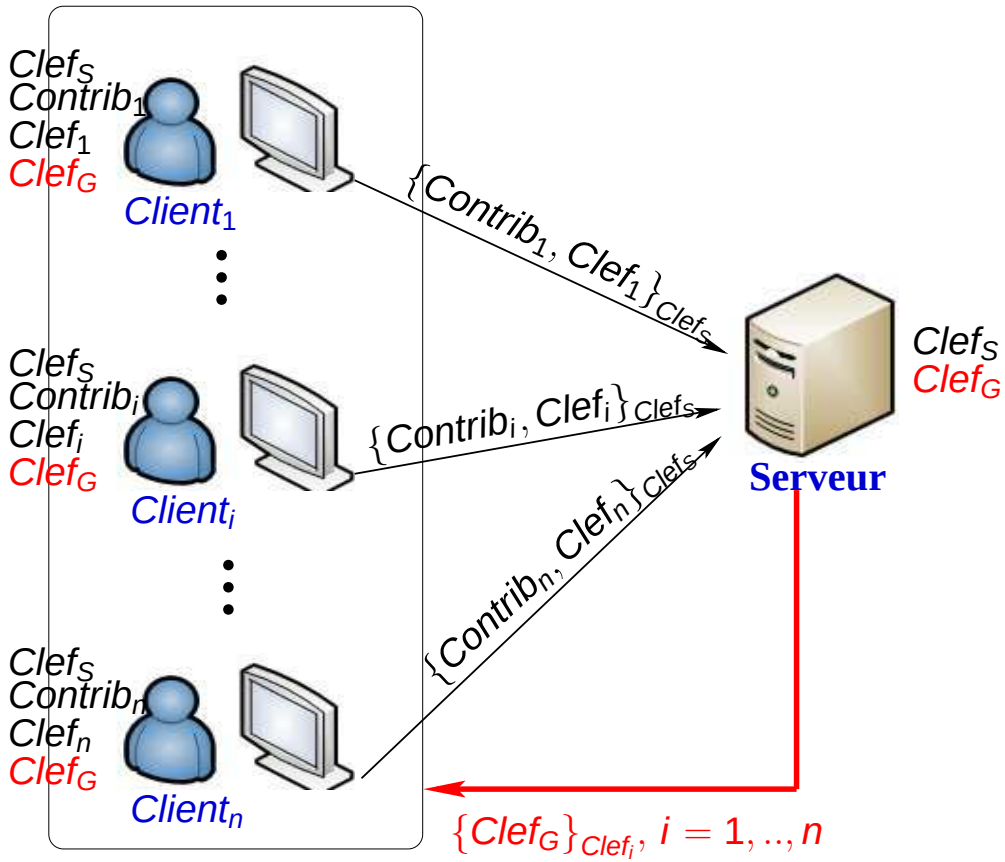
Gestion de clefs

- Gestion centralisée ;
- Gestion distribuée ;



Gestion de clefs

- Gestion centralisée ;
- Gestion distribuée ;



Défis de vérification des PGs

- propriétés de sécurité spécifiques ;
  - génération de la clef de groupe : propriétés liées à un groupe statique ;
    - chacun des membres du groupe contribue à la clef de groupe ;
    - la clef de groupe ne dépend que de ces contributions ;
    - tous les membres du groupe se mettent d'accord sur la même clef de groupe ;

Défis de vérification des PGs

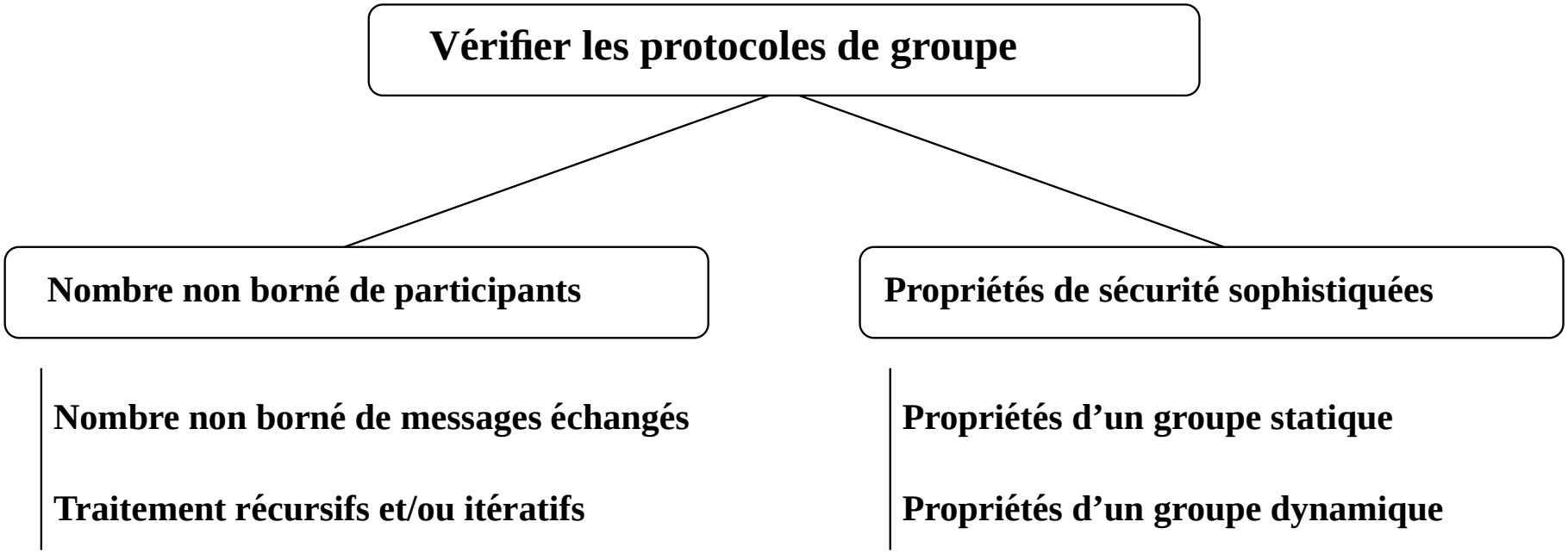
- propriétés de sécurité spécifiques ;
  - génération de la clef de groupe : propriétés liées à un groupe statique ;
    - chacun des membres du groupe contribue à la clef de groupe ;
    - la clef de groupe ne dépend que de ces contributions ;
    - tous les membres du groupe se mettent d'accord sur la même clef de groupe ;
  - mise à jour de la clef : propriétés liées à un groupe dynamique ;
    - les nouveaux membres du groupe n'ont pas accès aux communications précédentes ;
    - les anciens membres du groupe n'ont pas accès aux communications futures ;

Défis de vérification des PGs

- propriétés de sécurité spécifiques ;
  - génération de la clef de groupe : propriétés liées à un groupe statique ;
  - mise à jour de la clef : propriétés liées à un groupe dynamique ;
- nombre non borné de participants ;
  - nombre non borné de messages échangés ;
  - messages à structure non bornée ;
  - traitements itératifs et récursifs ;
  - Exemple :

$$\langle \{m_1\}_k, .., \{m_i\}_k, .., \{m_n\}_k \rangle \longrightarrow f(\langle m_1, .., m_i, .., m_n \rangle)$$

Défis de vérification des PGs



Vérification des protocoles de groupes

- Paulson [1997] : étude du protocole RA ;
- Résultats expérimentaux :
  - manuels : Pereira-Quisquater [2003] ;
  - (semi-)automatiques : Meadows [2001], Steel [2004].
- Résultats théoriques :
  - Kremer, Mercier, Treinen [2008]
    - ▲ automates d'arbres avancés ;
    - ▼ intrus passif.
  - Küsters, Wilke [2004]
    - ▲ automates d'arbres ;
    - ▼ clefs atomiques.
  - Truderung [2005]
    - ▲ théorie de sélection : une classe de théorie de Horn ;
    - ▼ clefs atomiques (CA).
    - Extension1 : Küsters-Truderung [2007] (XOR, CA) ;
    - Extension2 : Kürtz [2007] (fraîcheur de nonces, CA).

Contributions de la thèse

- Expérimentales :
  - Modélisation et vérification de quelques PGs ;
    - Asokan-Ginzboorg, GDH, Tanaka-Sato, Iolus.
  - Vérification d'une architecture de PGs en collaboration avec l'équipe MADYNES de Loria [SAR'06,AT'07] ;

Contributions de la thèse

- Expérimentales :
  - Modélisation et vérification de quelques PGs ;
    - Asokan-Ginzboorg, GDH, Tanaka-Sato, Iolus.
  - Vérification d'une architecture de PGs en collaboration avec l'équipe MADYNES de Loria [SAR'06,AT'07] ;
  - Définition d'un modèle de services pour détecter des types d'attaques sur les PGs [CRISIS'05,REE'06] ;
  - Proposition d'une procédure de recherche d'attaques sur les PGs [CSTVA'06,FTCP'07] ;

Contributions de la thèse

- Expérimentales :
  - Modélisation et vérification de quelques PGs ;
    - Asokan-Ginzboorg, GDH, Tanaka-Sato, Iolus.
  - Vérification d'une architecture de PGs en collaboration avec l'équipe MADYNES de Loria [SAR'06,AT'07] ;
  - Définition d'un modèle de services pour détecter des types d'attaques sur les PGs [CRISIS'05,REE'06] ;
  - Proposition d'une procédure de recherche d'attaques sur les PGs [CSTVA'06,FTCP'07] ;
- Théoriques :
  - Proposition d'un modèle synchrone permettant le traitement des listes paramétrés [LOPSTR'08] ;
  - Proposition d'une procédure de décision pour une classe de PGs [CSF'09] ;

Plan

Modélisation des protocoles de groupe [CRISIS'05,REE'06]

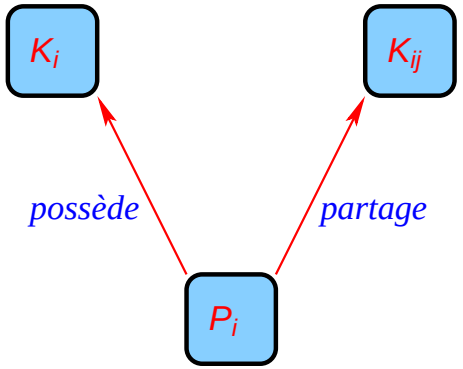
Un protocole de groupe : $<\mathbf{P}, \mathbf{K}, \mathbf{S}>$ avec,

- **P** : membres du groupe ;
- **K** : connaissances des participants ;
- **S** : services.
  - Service : toute contribution d'un participant pour engendrer la clef de groupe ;
  - Une contribution de P_i à P_j : toute information engendrée par P_i, utile à P_j afin de déduire la clef de groupe.

Modélisation des protocoles de groupe [CRISIS'05,REE'06]

Un protocole de groupe : $<\mathbf{P}, \mathbf{K}, \mathbf{S}>$ avec,

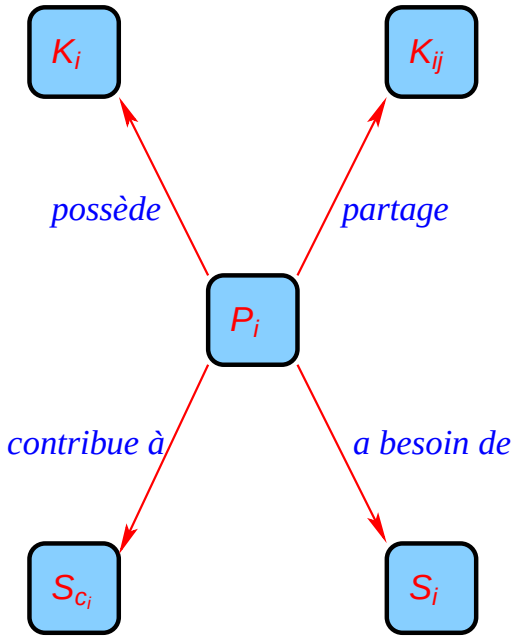
- **P** : participants membres du groupe ;
- **K** : connaissances des participants ;
  - $K_i \subseteq K$: connaissances privées de P_i ;
  - $K_{ij} \subseteq K$: connaissances partagées entre P_i et P_j.
- **S** : services.



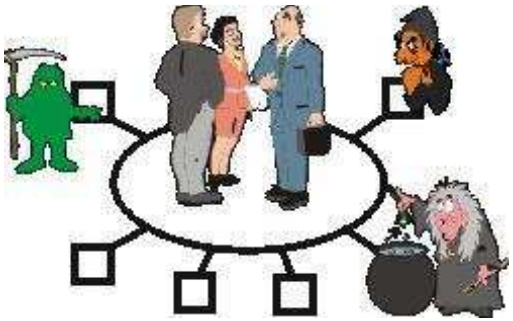
Modélisation des protocoles de groupe [CRISIS'05,REE'06]

Un protocole de groupe : $<\mathbf{P}, \mathbf{K}, \mathbf{S}>$ avec,

- **P** : membres du groupe ;
- **K** : connaissances des participants ;
  - $K_i \subseteq K$: connaissances privées de P_i ;
  - $K_{ij} \subseteq K$: connaissances partagées entre P_i et P_j.
- **S** : services.
  - $S_i \subseteq S$: l'ensemble minimal des services utiles à P_i pour construire la clef de groupe ;
  - $S_{c_i} = \{s \in S \mid \exists t \text{ sous-terme de } s, \text{ tel que } t \in K_i\}$.

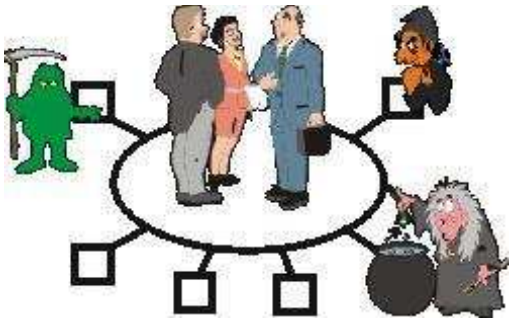


Exemple : A-GDH



$$
\begin{array}{lcll}
P_1 & \longrightarrow & P_2 & : \ \alpha, \alpha^{r_1} \\
P_2 & \longrightarrow & P_3 & : \ \alpha^{r_2}, \alpha^{r_1}, \alpha^{r_1 r_2} \\
P_3 & \longrightarrow & P_1 & : \ \alpha^{r_2 r_3 k_{13}} \\
P_3 & \longrightarrow & P_2 & : \ \alpha^{r_1 r_3 k_{23}} \\
\textit{Clef du groupe} & & & : \ \alpha^{r_1 r_2 r_3}
\end{array}
$$

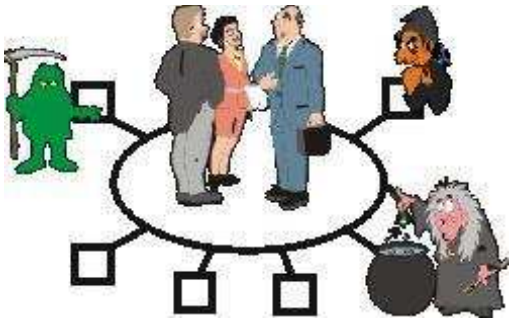
Exemple : A-GDH



$$
\begin{array}{lcll}
P_1 & \longrightarrow & P_2 & : \ \alpha, \alpha^{r_1} \\
P_2 & \longrightarrow & P_3 & : \ \alpha^{r_2}, \alpha^{r_1}, \alpha^{r_1 r_2} \\
P_3 & \longrightarrow & P_1 & : \ \alpha^{r_2 r_3 k_{13}} \\
P_3 & \longrightarrow & P_2 & : \ \alpha^{r_1 r_3 k_{23}}
\end{array}
$$

$$
\begin{aligned}
P &= \{P_1, P_2, P_3\}, \\
K &= \{r_1, r_2, r_3, k_{13}, k_{23}\}, \\
S &= \{\alpha^{r_1}, \alpha^{r_2}, \alpha^{r_1 r_2}, \alpha^{r_1 r_3 k_{23}}, \alpha^{r_2 r_3 k_{13}}\}
\end{aligned}
$$

Exemple : A-GDH



$$
\begin{array}{lcll}
P_1 & \longrightarrow & P_2 & : \ \alpha, \alpha^{r_1} \\
P_2 & \longrightarrow & P_3 & : \ \alpha^{r_2}, \alpha^{r_1}, \alpha^{r_1 r_2} \\
P_3 & \longrightarrow & P_1 & : \ \alpha^{r_2 r_3 k_{13}} \\
P_3 & \longrightarrow & P_2 & : \ \alpha^{r_1 r_3 k_{23}}
\end{array}
$$

$$
\begin{aligned}
P &= \{P_1, P_2, P_3\}, \\
K &= \{r_1, r_2, r_3, k_{13}, k_{23}\}, \\
S &= \{\alpha^{r_1}, \alpha^{r_2}, \alpha^{r_1 r_2}, \alpha^{r_1 r_3 k_{23}}, \alpha^{r_2 r_3 k_{13}}\}
\end{aligned}
$$

$\mathbf{P_i}$	$\mathbf{S_i}$	$\mathbf{S_{c_i}}$	$\mathbf{K_i}$	$\cup_j \mathbf{K_{ij}}$
$\mathbf{P_1}$	$\alpha^{r_2 r_3 k_{13}}$	$\alpha^{r_1}, \alpha^{r_1 r_2}, \alpha^{r_1 r_3 k_{13}}$	r_1	k_{13}
$\mathbf{P_2}$	$\alpha^{r_1 r_3 k_{23}}$	$\alpha^{r_2}, \alpha^{r_1 r_2}, \alpha^{r_2 r_3 k_{13}}$	r_2	k_{23}
$\mathbf{P_3}$	$\alpha^{r_1 r_2}$	$\alpha^{r_2 r_3 k_{13}}, \alpha^{r_1 r_3 k_{23}}$	r_3	k_{13}, k_{23}

Dynamisme du groupe, notion d'événement

- Changement de composition : ajout ou suppression d'un ou plusieurs membres du groupe ;
- Changement de structure : montée ou descente de classe dans les groupes hiérarchiques ;

$\Rightarrow$

$$GP^{T} = \{ <P^{\tau}, K^{\tau}, S^{\tau}> \mid \tau \in T \}$$

- Événement : le passage d'un état du groupe GP^{τ} à un autre état $GP^{\tau'}$;

$$<P, K, S>^{\tau} \longrightarrow_{event} <P, K, S>^{\tau'} \ .$$

si

$$(P^{\tau} \neq P^{\tau'}) \vee (P^{\tau} = P^{\tau'} \wedge K^{\tau} \neq K^{\tau'}) \ .$$

Caractéristiques

Intéractions entre les différents ensembles du système.

1. Unicité des identificateurs des agents ;
2. Visibilité des connaissances privées ;
3. Visibilité des connaissances partagées ;
4. Distinction des services utiles ;
5. Indépendance des services utiles ;
6. Correspondance de services ;
7. Déduction de la même clef de groupe ;
   $\forall P_i, P_j \in \mathcal{P}, i \neq j, KeyView_i = KeyView_j$
8. Services minimaux.

Propriétés de sécurité des PGs

- Propriétés statiques.
  - Authentification implicite de la clef ;
    - $K_I, \cup_j K_{Ij}, \cup_{P_k \in P} S_{c_k} \not\models KeyVue_i(K_i, \cup_j K_{ij}, S_i)$.
  - Secret de la clef ;
  - Confirmation de la clef ;
  - Intégrité ;
    - confirmation des services + correspondance des services ;
    - $\forall P_i \in P, \forall s \in S_i, \forall t$ sous-terme de $s, t \notin K_I$.

Propriétés de sécurité des PGs

- Propriétés statiques.
  - Authentification implicite de la clef ;
    - $K_I, \cup_j K_{Ij}, \cup_{P_k \in P} S_{c_k} \not\models KeyVue_i(K_i, \cup_j K_{ij}, S_i)$.
  - Secret de la clef ;
  - Confirmation de la clef ;
  - Intégrité ;
    - confirmation des services + correspondance des services ;
    - $\forall P_i \in P, \forall s \in S_i, \forall t$ sous-terme de $s, t \notin K_I$.
- Propriétés dynamiques.
  - Secret futur ;
    - $\forall \tau_i, \tau_j \in T, i < j, \{\mathcal{K}_G^{\tau_1}, \mathcal{K}_G^{\tau_2}, \ldots, \mathcal{K}_G^{\tau_i}\} \not\models \mathcal{K}_G^{\tau_j}$.
    - Protection contre une attaque par service connu ;
    - Protection contre une attaque par clef connue ;
  - Secret passé ;
  - Indépendance de la clef de groupe.

Première application

Détection de types d'attaques à partir des scénarii.

Protocoles étudiés :

- A-GDH ;
- SA-GDH ;
- Asokan-Ginzboorg ;
- Bresson-Chevassut-Essiari-Pointcheval.

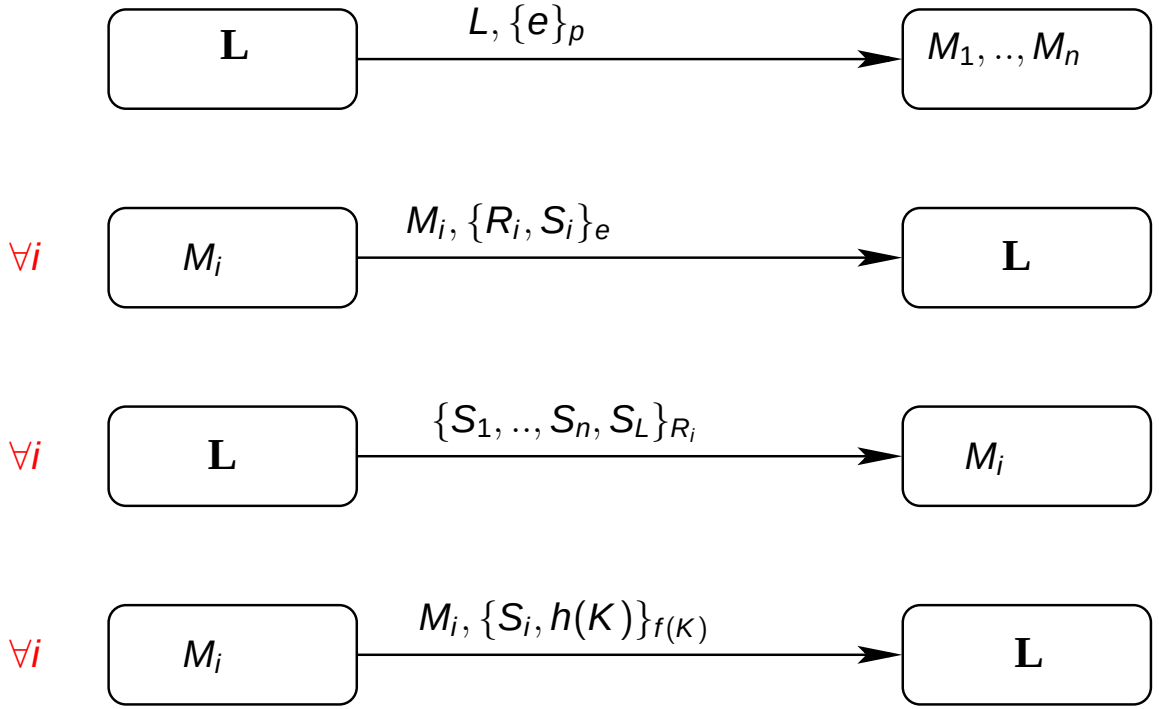
A-GDH

- caractéristique non vérifiée
  - déduction de la même clef de groupe.
- propriétés non vérifiées
  - authentification implicite ;
  - secret de la clef ;
  - confirmation de la clef ;
  - intégrité.

Deuxième application [CSTVA'06,FTCP'07]

- Procédure de recherche d'attaques se basant sur le modèle de services ;
- Données de la procédure :
  1. propriétés de sécurité : secret, authentification, accord de clefs ;
     - Accord de clefs :
       $\forall P_i, P_j \in \mathbf{P}$ *avec* $i \neq j$, $KeyView_i = KeyView_j$.
  2. protocole à vérifier
     - défini par $(\{\mathcal{R}_p \rightarrow \mathcal{S}_p\}_{p\in\mathcal{P}}, <_{\mathcal{P}}, S_0)$
     - $\{\mathcal{R}_p \rightarrow \mathcal{S}_p\}_{p\in\mathcal{P}}$: l'ensemble des pas du protocole ;
     - $<_{\mathcal{P}}$: un ordre partiel sur l'ensemble des pas du protocole ;
     - S_0 : l'ensemble des connaissances initiales de l'intrus.

Étude de cas : Asokan-Ginzboorg [2000]



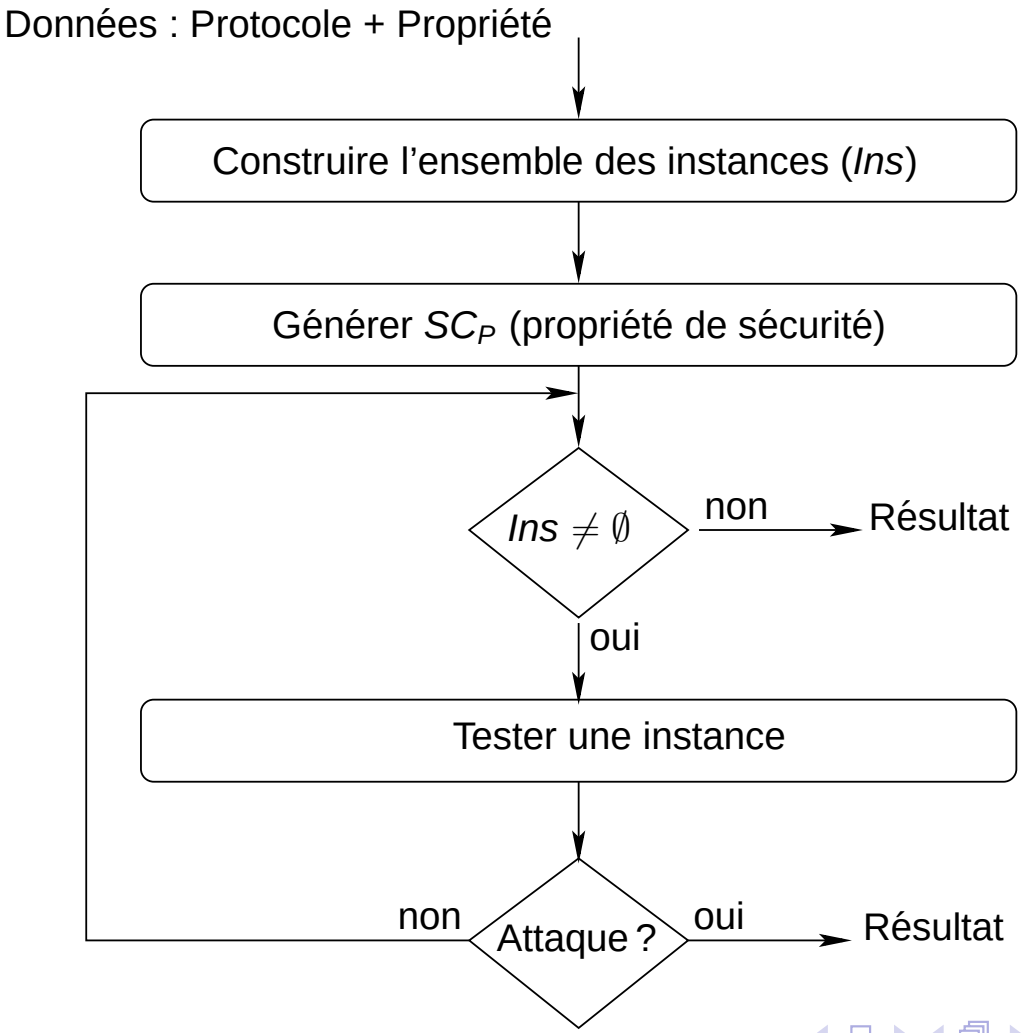
$$K = S_1, .., S_n, S_L$$

Application à l'étude de cas

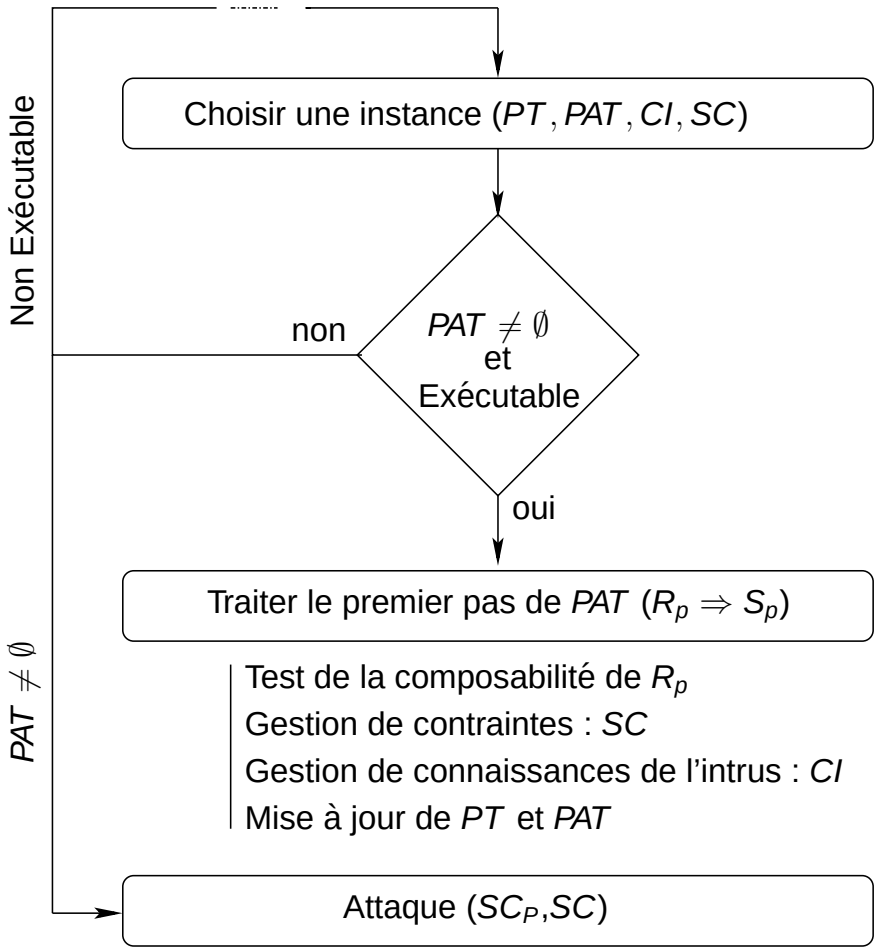
$\mathbf{pas_{111}}$:	$Init$	$\longrightarrow$	$a_1, \{e_1\}_p$
$\mathbf{pas_{121}}$:	$X_1, \{X_2, X_3\}_{e_1}$	$\longrightarrow$	$\{X_3, s_{11}\}_{X_2}$
$\mathbf{pas_{131}}$:	$X_1, \{X_3, h(X_3, s_{11})\}_{f(X_3,s_{11})}$	$\longrightarrow$	End
			$KeyView_{11} : f(X_3, s_{11})$
$\mathbf{pas_{211}}$:	$X_4, \{X_5\}_p$	$\longrightarrow$	$a_2, \{r_2, s_{21}\}_{X_5}$
$\mathbf{pas_{221}}$:	$\{X_6, X_7\}_{r_2}$	$\longrightarrow$	$a_2, \{s_{21}, h(X_6, X_7)\}_{f(X_6,X_7)}$
			$KeyView_{21} : f(X_6, X_7)$
$\mathbf{pas_{212}}$:	$Init$	$\longrightarrow$	$a_2, \{e_2\}_p$
$\mathbf{pas_{222}}$:	$X_8, \{X_9, X_{10}\}_{e_2}$	$\longrightarrow$	$\{X_{10}, s_{22}\}_{X_9}$
$\mathbf{pas_{232}}$:	$X_8, \{X_{10}, h(X_{10}, s_{22})\}_{f(X_{10},s_{22})}$	$\longrightarrow$	End
			$KeyView_{22} : f(X_{10}, s_{22})$
$\mathbf{pas_{112}}$:	$X_{11}, \{X_{12}\}_p$	$\longrightarrow$	$a_1, \{r_1, s_{12}\}_{X_{12}}$
$\mathbf{pas_{122}}$:	$\{X_{13}, X_{14}\}_{r_1}$	$\longrightarrow$	$a_1, \{s_{12}, h(X_{13}, X_{14})\}_{f(X_{13},X_{14})}$
			$KeyView_{12} : f(X_{13}, X_{14})$

- pas_{ijk} représente la j-ième étape du protocole jouée par le i-ième participant dans la k-ième session.
- $KeyView_{ij}$ est la vue de la clef du groupe du participant i durant la session j.
- Violation d'Accord de clef : $KeyView_{11} \neq KeyView_{21}$ ou $KeyView_{12} \neq KeyView_{22}$.

Procédure de recherche d'attaque



Procédure de recherche d'attaque

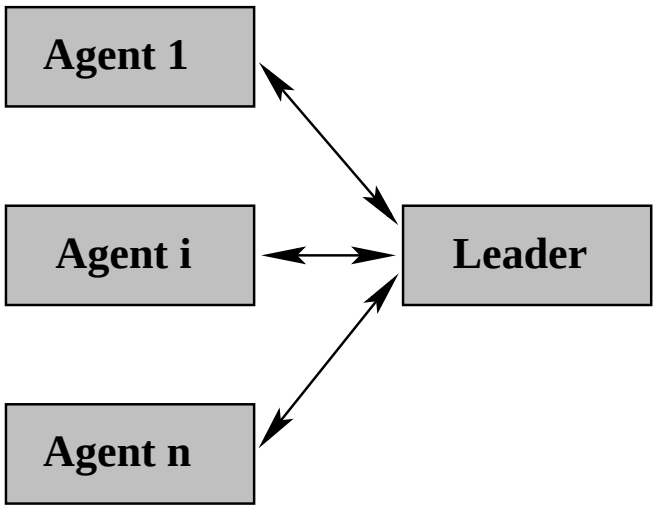


Bilan des protocoles étudiés

- Asokan-Ginzboorg (voir l'attaque) :
  - Deux sessions en parallèle, à deux participants chacune ;
  - Attaque d'accord de clefs dans chacune des sessions.
- GDH :
  - Deux attaques d'authentification pour quatre participants ;
  - Généralisation pour n participants ;
- A-GDH :
  - Exécution d'une session à quatre participants (voir l'attaque) ;
  - Généralisation à une session à n participants (voir l'attaque) ;
  - Résumé de l'attaque :
    - L'intrus fait partie du groupe ($I = P_i$) ;
    - Modification de la dernière composante du message envoyé au dernier participant ;
    - Isolation du dernier participant : il a une vue de la clef différente de celle du reste du groupe ; L'intrus possède les deux vues de la clef.

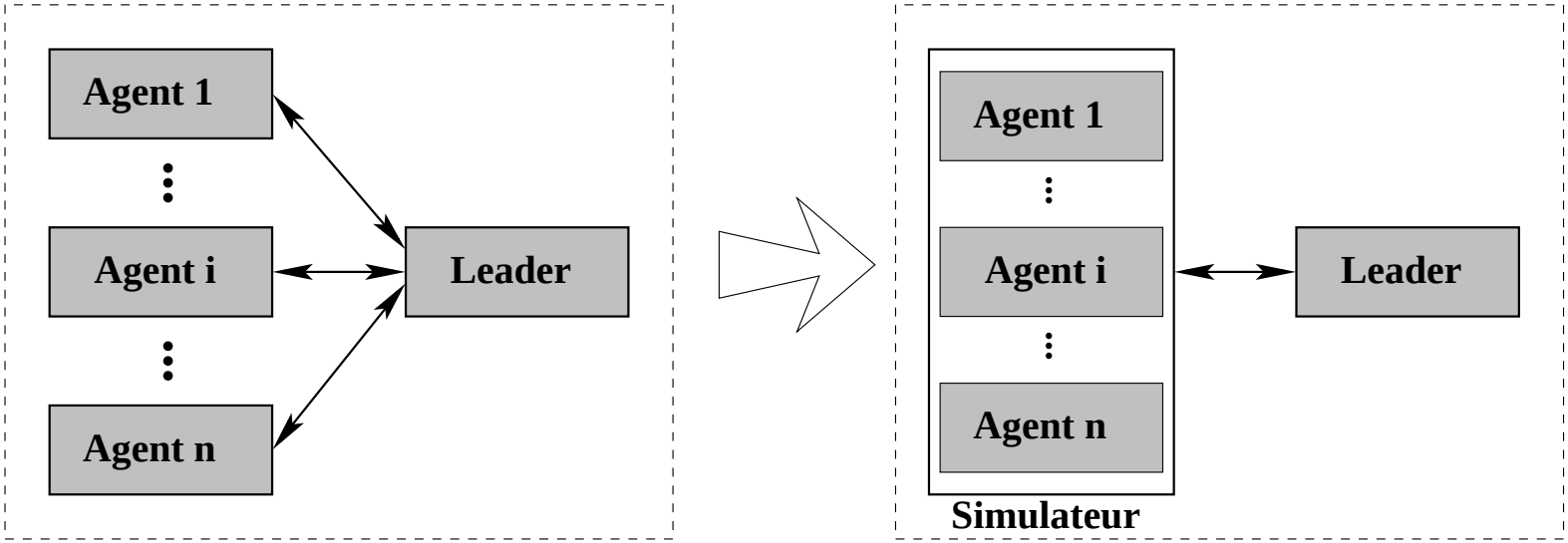
Plan

Idée de Base [LOPSTR'08]



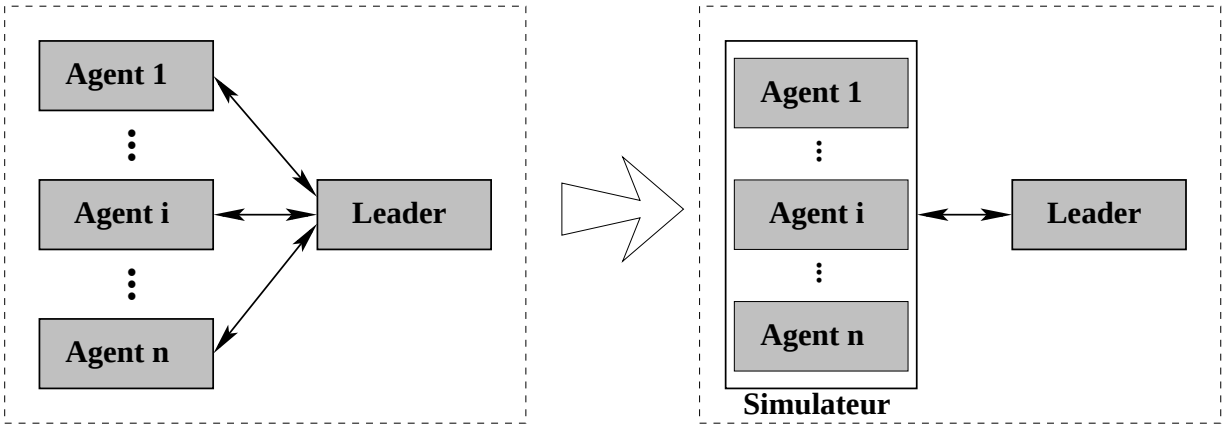
Idée de Base [LOPSTR'08]

- Transformation en version synchrone : introduction de la notion de simulateur ;



Idée de Base [LOPSTR'08]

- Transformation en version synchrone : introduction de la notion de simulateur ;



- Équivalence des deux modèles ;
  - correction ;
  - complétude dépend du protocole ;
    - équivalence pour Asokan-Ginzboorg.

Modèle de protocole, [Asokan-Ginzboorg à $n+1$ participants]

Version Initiale

1. $a_{n+1} \longrightarrow$ Tous : $\langle a_{n+1}, \{e\}_p \rangle$
2. $a_i \longrightarrow a_{n+1}$: $\langle a_i, \{\langle r_i, s_i \rangle\}_e \rangle$, $i = 1, .., n$
3. $a_{n+1} \longrightarrow a_i$: $\{\langle s_1, .., s_{n+1} \rangle\}_{r_i}$, $i = 1, .., n$
4. $a_i \longrightarrow a_{n+1}$: $\langle a_i, \{\langle s_i, h(s) \rangle\}_{f(s)} \rangle$, $i = 1, .., n$ et $s = \langle s_1, .., s_{n+1} \rangle$

Version Synchrone

1. $L \longrightarrow S$: $mpair(i, \langle a_{n+1}, \{e\}_p \rangle)$
2. $S \longrightarrow L$: $mpair(i, \langle a_i, \{\langle r_i, s_i \rangle\}_e \rangle)$
3. $L \longrightarrow S$: $mpair(i, \{\langle mpair(j, s_j), s' \rangle\}_{r_i})$
4. $S \longrightarrow L$: $mpair(i, \langle a_i, \{\langle s_i, h(\langle mpair(k, s_k), s' \rangle)\rangle\}_{f(mpair(k,s_k).s')} \rangle)$

Langage de messages

$\mathcal{T} = \{\mathcal{T}\}^p_{\mathcal{T}} \mid \{\mathcal{T}\}^s_{\mathcal{T}} \mid h(\mathcal{T}) \mid \langle \mathcal{T}, \ldots, \mathcal{T} \rangle \mid \mathcal{X} \mid \mathcal{C}$

Modèle de protocole, [Asokan-Ginzboorg à $n+1$ participants]

Version Initiale

1. $a_{n+1} \longrightarrow$ Tous : $\langle a_{n+1}, \{e\}_p \rangle$
2. $a_i \longrightarrow a_{n+1}$: $\langle a_i, \{\langle r_i, s_i \rangle\}_e \rangle$, $i = 1, .., n$
3. $a_{n+1} \longrightarrow a_i$: $\{\langle s_1, .., s_{n+1} \rangle\}_{r_i}$, $i = 1, .., n$
4. $a_i \longrightarrow a_{n+1}$: $\langle a_i, \{\langle s_i, h(s) \rangle\}_{f(s)} \rangle$, $i = 1, .., n$ et $s = \langle s_1, .., s_{n+1} \rangle$

Version Synchrone

1. $L \longrightarrow S$: $mpair(i, \langle a_{n+1}, \{e\}_p \rangle)$
2. $S \longrightarrow L$: $mpair(i, \langle a_i, \{\langle r_i, s_i \rangle\}_e \rangle)$
3. $L \longrightarrow S$: $mpair(i, \{\langle mpair(j, s_j), s' \rangle\}_{r_i})$
4. $S \longrightarrow L$: $mpair(i, \langle a_i, \{\langle s_i, h(\langle mpair(k, s_k), s' \rangle)\rangle\}_{f(mpair(k,s_k).s')} \rangle)$

Langage de messages

$\mathcal{T} = \{\mathcal{T}\}^p_{\mathcal{T}} \mid \{\mathcal{T}\}^s_{\mathcal{T}} \mid h(\mathcal{T}) \mid \langle \mathcal{T}, \ldots, \mathcal{T} \rangle \mid mpair(\mathcal{I}, \mathcal{T}) \mid \mathcal{X} \mid \mathcal{X}_{\mathcal{I}} \mid \mathcal{C} \mid \mathcal{C}_{\mathcal{I}}$

Indécidabilité

Theorem 1

Le problème d'insécurité des protocoles de groupe dans le modèle synchrone avec l'opérateur mpair est indécidable.

Démonstration.

En codant le problème PCP (Post Correspondance Problem) à deux lettres. □

Indécidabilité

Theorem 1

Le problème d'insécurité des protocoles de groupe dans le modèle synchrone avec l'opérateur mpair est indécidable.

Démonstration.

En codant le problème PCP (Post Correspondance Problem) à deux lettres. □

$\Rightarrow$ Besoin d'une classe plus restrictive : les protocoles bien-tagués à clefs autonomes.

Autonomie des *mpair*s

- Protocole autonome :
  $\mathcal{P} = \{R_i \Rightarrow S_i | i \in J\}$ est autonome ssi : $\mathcal{P} = R_i \Rightarrow S_i$
  - Pas d'indices hors des *mpair* ;
  - R_i et S_i autonomes.

$VI(R_i) = VI(S_i) = \emptyset$

R_i **et** S_i **autonomes**

VI : Variables d'Indices

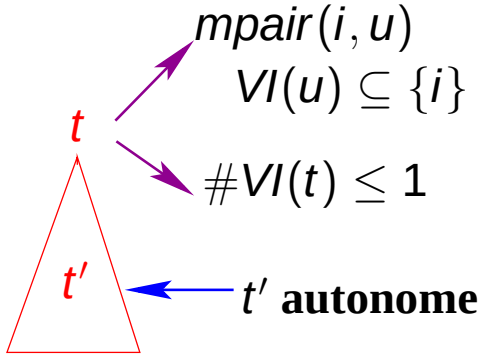
Autonomie des *mpair*s

- Protocole autonome :
  $\mathcal{P} = \{R_i \Rightarrow S_i | i \in J\}$ est autonome ssi : $\mathcal{P} = R_i \Rightarrow S_i$
  - Pas d'indices hors des *mpair* ;
  - R_i et S_i autonomes.

$VI(R_i) = VI(S_i) = \emptyset$

R_i **et** S_i **autonomes**

- Terme autonome t :
  - si $t = mpair(i, u)$, les variables d'indices de u sont dans $\{i\}$;
  - sinon, t contient au plus une variable d'indices
  - $\forall t' < t$, t' est autonome.



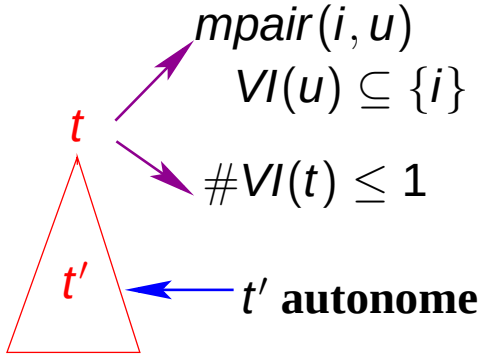
VI : Variables d'Indices

Autonomie des *mpair*s

- Protocole autonome :
  $\mathcal{P} = \{R_i \Rightarrow S_i | i \in J\}$ est autonome ssi : $\mathcal{P} = R_i \Rightarrow S_i$
  - Pas d'indices hors des *mpair* ;
  - R_i et S_i autonomes.

  $VI(R_i) = VI(S_i) = \emptyset$

  R_i **et** S_i **autonomes**
- Terme autonome t :
  - si $t = mpair(i, u)$, les variables d'indices de u sont dans $\{i\}$;
  - sinon, t contient au plus une variable d'indices
  - $\forall t' < t$, t' est autonome.



- Exemples :
  - ▲ $mpair(i, \langle a_i, mpair(j, \{c_j\}_k)\rangle)$.
  - ▼ $mpair(i, \langle a_i, mpair(j, \{c_j\}_{d_i})\rangle)$.

VI : Variables d'Indices

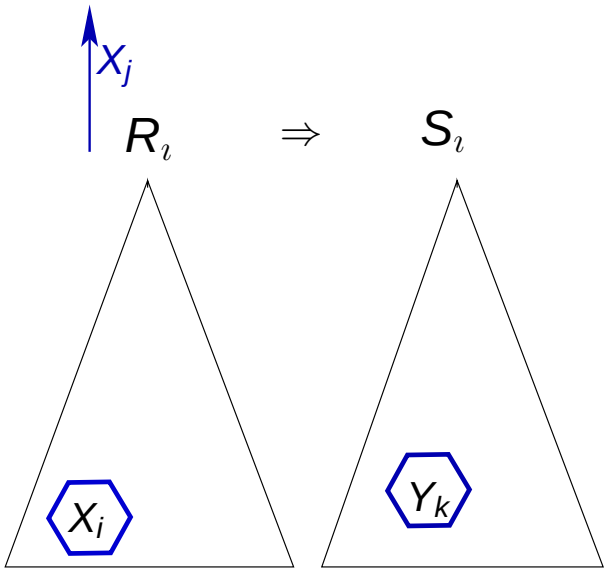
Protocoles bien-tagués

- Terme tagué t : $[e_i, t]$, noté aussi t^i ;

Protocoles bien-tagués

- Terme tagué t : $[e_i, t]$, noté aussi t^i ;

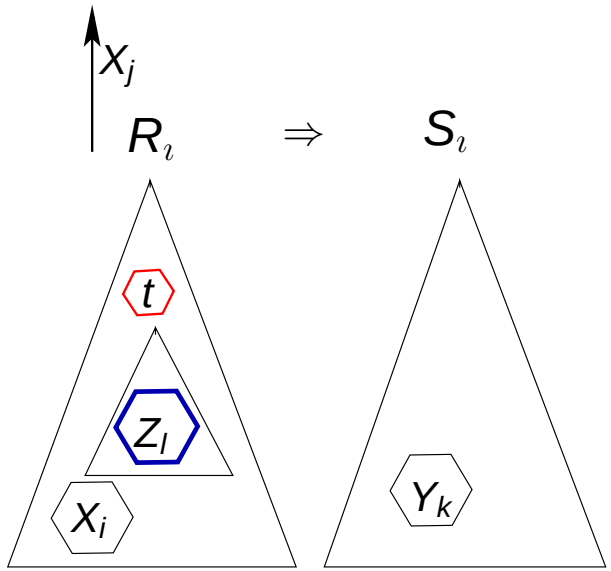
2. Un protocole $\mathcal{P} = \{R_\imath \Rightarrow S_\imath | \imath \in J\}$ est bien-tagué ssi :
   2.1. les variables indicées sont taguées sauf la première occurence ;



Protocoles bien-tagués

- Terme tagué t : $[e_i, t]$, noté aussi t^i ;

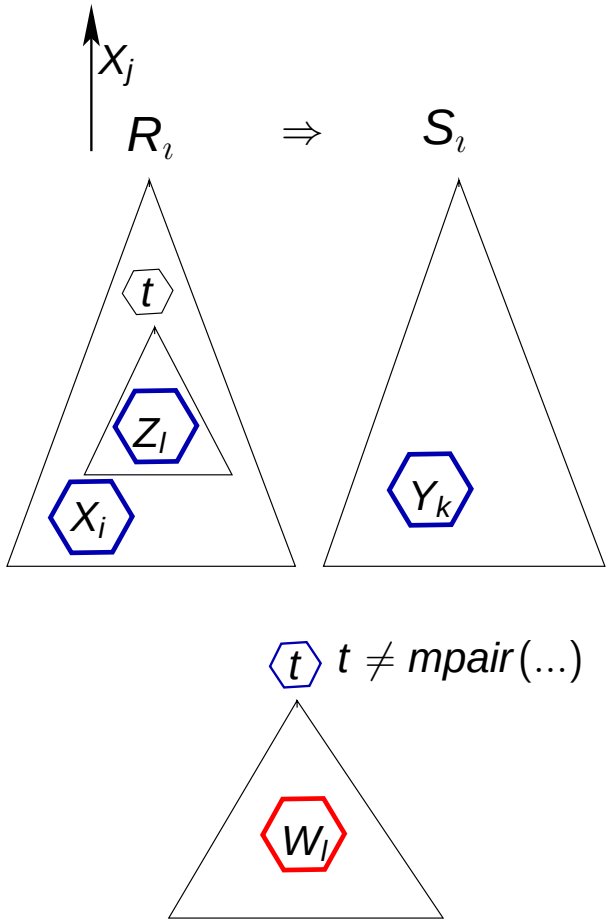
2. Un protocole $\mathcal{P} = \{R_\imath \Rightarrow S_\imath | \imath \in J\}$ est bien-tagué ssi :
   - 2.1. les variables indicées sont taguées sauf la première occurence ;
   - 2.2. si t dans $R_\imath$ est tagué, alors ses variables indicées sont aussi taguées ;
     - $mpair(i, \{X_i\}_k{}^i)$;



Protocoles bien-tagués

- Terme tagué t : $[e_i, t]$, noté aussi t^i ;

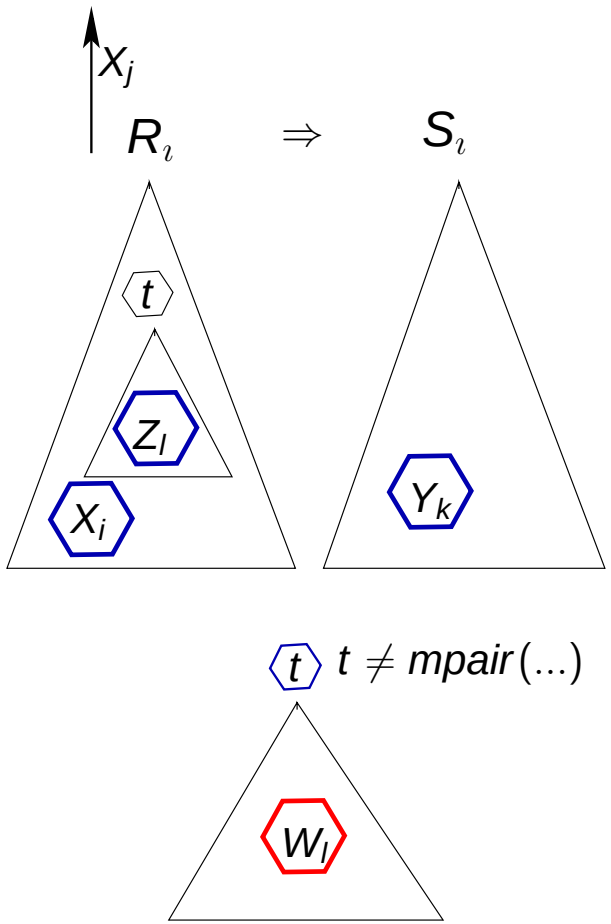
2. Un protocole $\mathcal{P} = \{R_\imath \Rightarrow S_\imath | \imath \in J\}$ est bien-tagué ssi :
   - 2.1. les variables indicées sont taguées sauf la première occurence ;
   - 2.2. si t dans $R_\imath$ est tagué, alors ses variables indicées sont aussi taguées ;
     - $mpair(i, \{X_i\}_k{}^i)$;
   - 2.3. pour tout terme $t = f(s_1, .., s_k)$ de $\mathcal{P}$ avec $f \neq mpair$, t hérite du tag de ses sous-termes ;
     - $mpair(i, \langle t, X_i{}^i \rangle)$;



Protocoles bien-tagués

- Terme tagué t : $[e_i, t]$, noté aussi t^i ;

2. Un protocole $\mathcal{P} = \{R_\imath \Rightarrow S_\imath | \imath \in J\}$ est bien-tagué ssi :
   - 2.1. les variables indicées sont taguées sauf la première occurence ;
   - 2.2. si t dans $R_\imath$ est tagué, alors ses variables indicées sont aussi taguées ;
     - $mpair(i, \{X_i\}_k{}^i)$;
   - 2.3. pour tout terme $t = f(s_1, .., s_k)$ de $\mathcal{P}$ avec $f \neq mpair$, t hérite du tag de ses sous-termes ;
     - $mpair(i, \langle t, X_i{}^i \rangle)$;
   - 2.4. $\mathcal{P}$ est autonome.



Protocoles à clefs autonomes

3. Un protocole $\mathcal{P}$ est à clefs autonomes ssi les clefs n'ont pas de variables d'indices hors des *mpair*s.

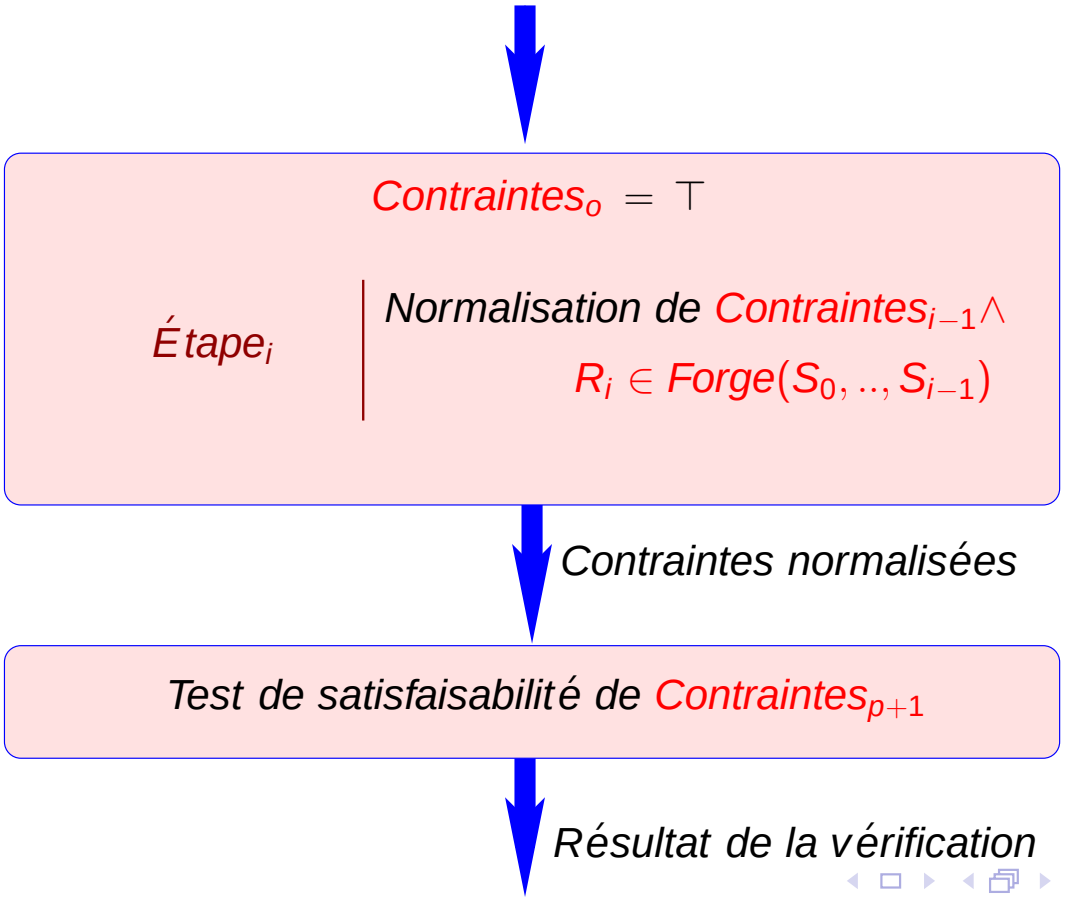
- ▲ $\{t\}_{mpair(i,s_i)}$ et $\{t\}_k$;
- ▼ $\{t\}_{s_i}$ and $\{t\}_{X_i}$.

- ▲ Exemple de protocole bien-tagué à clefs autonomes :
$P = \left\{ mpair(i, X_i) \Rightarrow mpair(j, \langle c_j, mpair(i, \{{X_i}^i\}_k{}^i)\rangle) \right\}$

Algorithme

$$Protocole : P = \{R_i \Rightarrow S_i\}_{i=1,..,p}$$

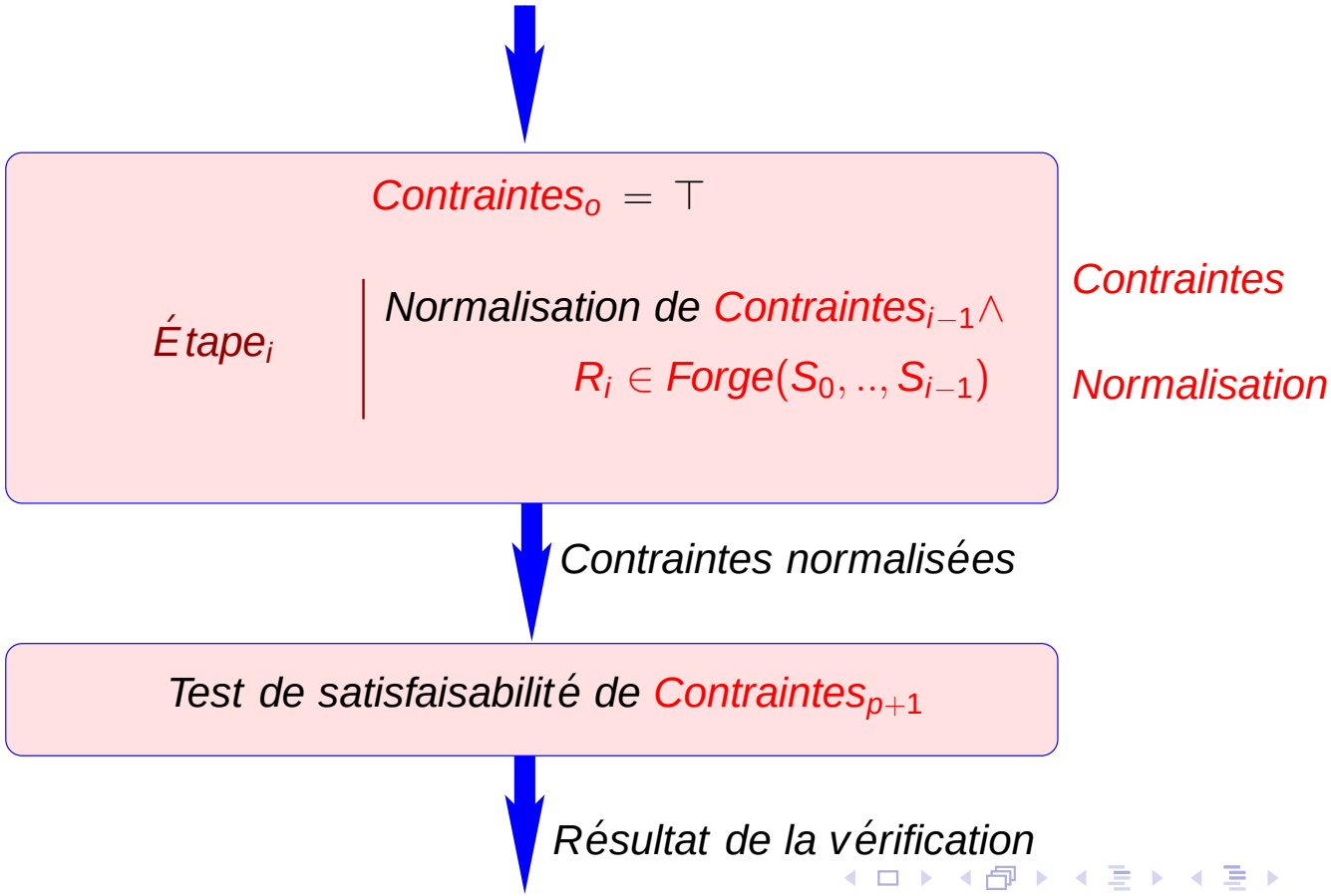
$$R_{p+1} = Secret,\ S_0 : Connaissances\ initiales\ de\ l'intrus$$



Algorithme

$$Protocole : P = \{R_i \Rightarrow S_i\}_{i=1,..,p}$$

$$R_{p+1} = Secret,\ S_0 : Connaissances\ initiales\ de\ l'intrus$$



Système de contraintes

- système de contraintes : disjonction de conjonction de contraintes.

$$\forall Q\ \exists R\ (B_1 \vee \ldots \vee B_p)$$

- bloc de contraintes : conjonction de contraintes.

$$(ctr_1 \wedge \ldots \wedge ctr_l)$$

- contraintes élémentaires : expriment les différentes manières de construction d'un terme par l'intrus (composition ou décomposition).
  - Exemples : $t \in Forge(E)$, $t \in Forge_c(E)$.

Normalisation

Des règles de transformation de systèmes de contraintes, organisées en six groupes G_1,..,G_6.

- G_1 : garder des propriétés de syntaxe ou de formattage ;
  - Exemple : manipuler les étiquettes des contraintes ;

Normalisation

Des règles de transformation de systèmes de contraintes, organisées en six groupes G_1,..,G_6.

- G_1 : garder des propriétés de syntaxe ou de formattage ;
  - Exemple : manipuler les étiquettes des contraintes ;
- G_2 et G_3 : énumérer les possibilités pour un terme d'être construit par l'intrus (composition ou décomposition) ;
  - $\forall Q\ \exists R\ S \vee (B \wedge mpair(k,t) \in Forge_c(E)) \Longrightarrow$
    $\forall Q.k\ \exists R\ \ S \vee (B \wedge t \in Forge(E))$

Normalisation

Des règles de transformation de systèmes de contraintes, organisées en six groupes G_1,..,G_6.

- G_1 : garder des propriétés de syntaxe ou de formattage ;
  - Exemple : manipuler les étiquettes des contraintes ;
- G_2 et G_3 : énumérer les possibilités pour un terme d'être construit par l'intrus (composition ou décomposition) ;
  - $\forall Q\ \exists R\ S \vee (B \wedge mpair(k,t) \in Forge_c(E)) \Longrightarrow$
    $\forall Q.k\ \exists R\ \ S \vee (B \wedge t \in Forge(E))$
- G_4 : coder l'algorithme d'unification sur les termes ;

Normalisation

Des règles de transformation de systèmes de contraintes, organisées en six groupes G_1,..,G_6.

- G_1 : garder des propriétés de syntaxe ou de formattage ;
  - Exemple : manipuler les étiquettes des contraintes ;
- G_2 et G_3 : énumérer les possibilités pour un terme d'être construit par l'intrus (composition ou décomposition) ;
  - $\forall Q\ \exists R\ S \vee (B \wedge mpair(k,t) \in Forge_c(E)) \Longrightarrow$
    $\forall Q.k\ \exists R\ \ S \vee (B \wedge t \in Forge(E))$
- G_4 : coder l'algorithme d'unification sur les termes ;
- G_5 et G_6 : manipuler les entrelacements possibles entre les contraintes d'un même bloc ou entre différents blocs ;

Résultat principal [CSF'09]

Theorem 2

Le problème d'insécurité des protocoles bien-tagués à clefs autonomes est décidable.

Résultat principal [CSF'09]

Theorem 2

Le problème d'insécurité des protocoles bien-tagués à clefs autonomes est décidable.

Démonstration.

La preuve est une conséquence directe des Propositions 1, 2 et 3. □

Résultats intermédiaires

Proposition 1 (Terminaison)

La normalisation termine pour les protocoles bien-tagués à clefs autonomes.

Résultats intermédiaires

Proposition 1 (Terminaison)

La normalisation termine pour les protocoles bien-tagués à clefs autonomes.

Proposition 2 (Correction et Complétude)

La normalisation est correcte et complète.

Résultats intermédiaires

Proposition 1 (Terminaison)

La normalisation termine pour les protocoles bien-tagués à clefs autonomes.

Proposition 2 (Correction et Complétude)

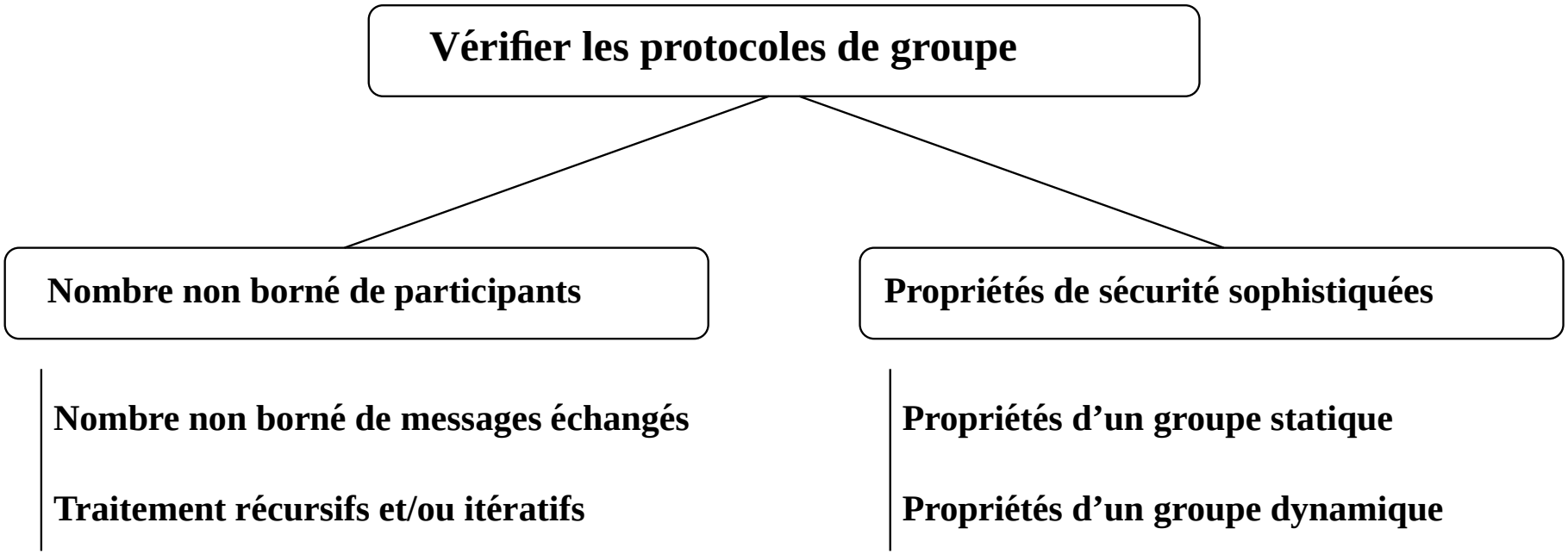
La normalisation est correcte et complète.

Proposition 3 (Satisfaisabilité)

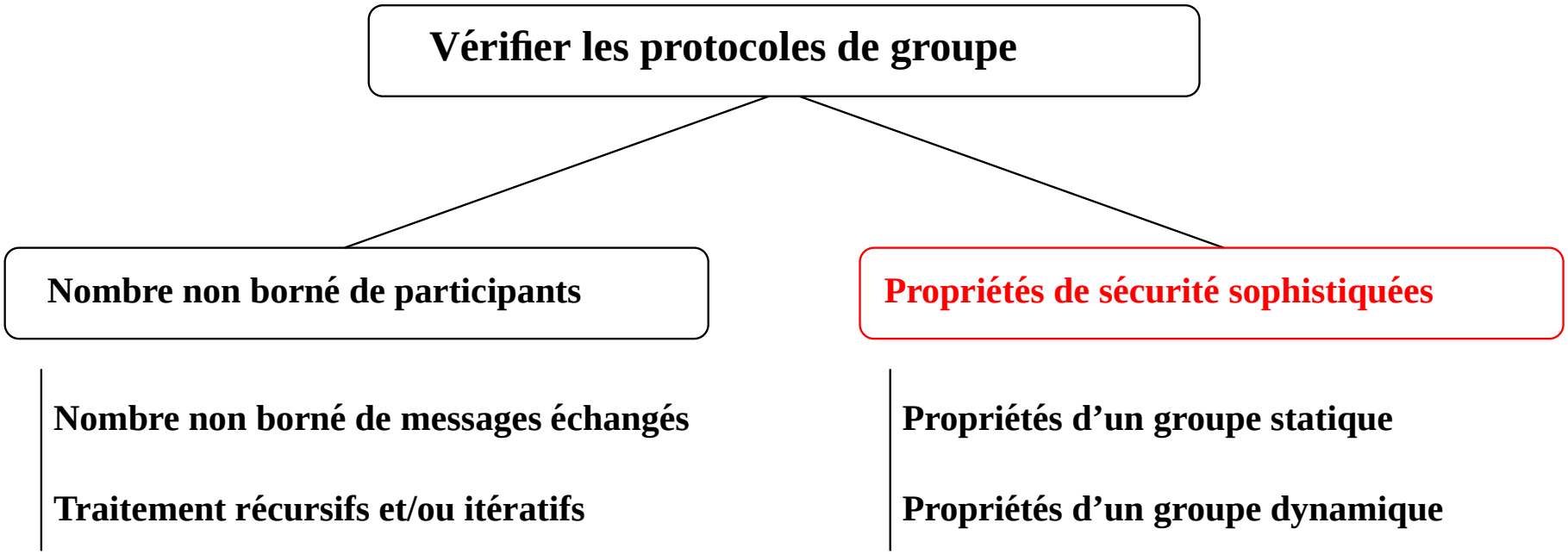
La satisfaisabilité du système résultant des contraintes normalisées est décidable.

Plan

Conclusion


Vérifier les protocoles de groupe
Nombre non borné de participants
Propriétés de sécurité sophistiquées
Nombre non borné de messages échangés
Traitement récursifs et/ou itératifs
Propriétés d'un groupe statique
Propriétés d'un groupe dynamique

Conclusion


Vérifier les protocoles de groupe
Nombre non borné de participants
Propriétés de sécurité sophistiquées
Nombre non borné de messages échangés
Traitement récursifs et/ou itératifs
Propriétés d'un groupe statique
Propriétés d'un groupe dynamique

Conclusion

- Définition d'un modèle de services pour les protocoles de groupes et plus généralement les protocoles contribuants ;

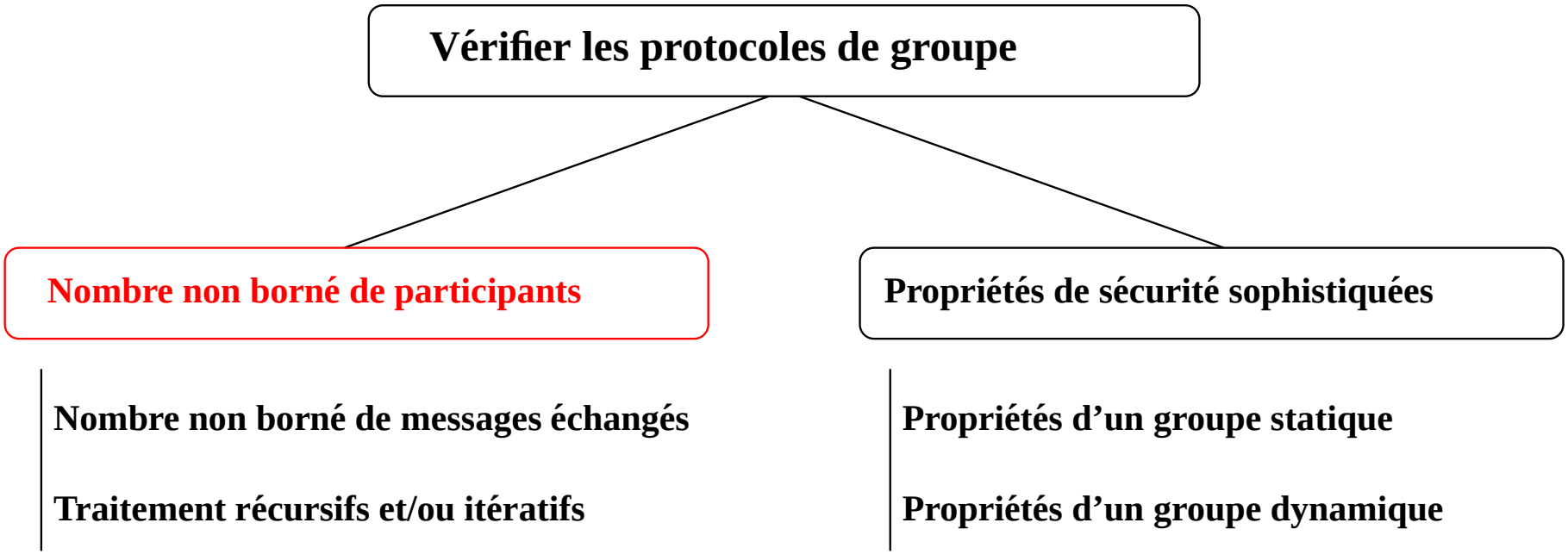
Conclusion

- Définition d'un modèle de services pour les protocoles de groupes et plus généralement les protocoles contribuants ;
- Première application : identification de différents types d'attaques sur des scénarii particuliers ;
  - protocoles testés : A-GDH, SA-GDH, Asokan-Ginzboorg, Bresson-Chevassut-Essiari-Pointcheval.

Conclusion

- Définition d'un modèle de services pour les protocoles de groupes et plus généralement les protocoles contribuants ;
- Première application : identification de différents types d'attaques sur des scénarii particuliers ;
  - protocoles testés : A-GDH, SA-GDH, Asokan-Ginzboorg, Bresson-Chevassut-Essiari-Pointcheval.
- Deuxième application : proposition d'une procédure de recherche d'attaques
  - résolution de deux systèmes de contraintes ;
  - protocoles testés : GDH, A-GDH, Asokan-Ginzboorg :
    - Attaques à nombre de participants fixe ;
    - Généralisation des attaques trouvées sur GDH et A-GDH pour couvrir le cas de n participants.

Conclusion


Vérifier les protocoles de groupe
Nombre non borné de participants
Propriétés de sécurité sophistiquées
Nombre non borné de messages échangés
Traitement récursifs et/ou itératifs
Propriétés d'un groupe statique
Propriétés d'un groupe dynamique

Conclusion

- Proposition d'un modèle synchrone pour les protocoles de groupes ;
  - Introduction de la notion de simulateur ;
  - Traitement des listes de messages dont la longueur est un paramètre ;
  - Ajout d'un nouvel opérateur *mpair* ;
    - Indécidabilité du problème d'insécurité ;

Conclusion

- Proposition d'un modèle synchrone pour les protocoles de groupes ;
  - Introduction de la notion de simulateur ;
  - Traitement des listes de messages dont la longueur est un paramètre ;
  - Ajout d'un nouvel opérateur *mpair* ;
    - Indécidabilité du problème d'insécurité ;
- Proposition d'une nouvelle procédure de décision pour la classe de protocoles bien-tagués à clefs autonomes ;
  - Correction et complétude de la normalisation ;
  - Satisfaisabilité ;
  - Terminaison ;

Perspectives liées au modèle de services

- Extension de ce modèle pour considérer d'autres propriétés liées à la topologie des groupes ;
  - Protocoles hiérarchiques : le secret d'une clef dépend du niveau des éléments détenant cette clef.
- Implantation d'un outil spécifique à la recherche d'attaque sur les protocoles de groupe traitant différentes propriétés.

Perspectives liées au modèle synchrone

- Prise en compte de propriétés algébriques des opérateurs ;
- Preuve de terminaison pour une classe plus significative de protocoles de groupe ;
  -  Affaiblir la restriction de clefs autonomes
- Considération de listes de tailles différentes ;
  -  Gestion de la dynamique des groupes : entrées et sorties des membres.

Autres perspectives

- Combinaison des deux modèles ;
  -  Introduction de relations entre les différents éléments des listes.

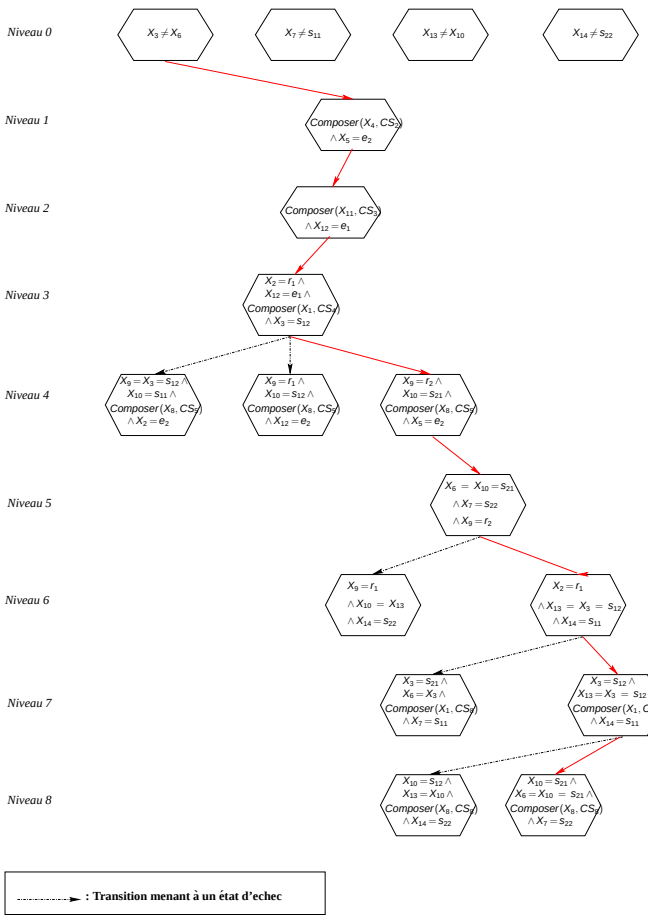
Autres perspectives

- Combinaison des deux modèles ;
  - Introduction de relations entre les différents éléments des listes.
- Simulation d'un nombre non borné de sessions par le biais du modèle de simulateur ;
  - Un message construit sur l'opérateur *mpair* représenterait les informations relatives à chacune des sessions.
  - Complétude de la transformation asynchrone-synchrone.

Autres perspectives

- Combinaison des deux modèles ;
  - Introduction de relations entre les différents éléments des listes.
- Simulation d'un nombre non borné de sessions par le biais du modèle de simulateur ;
  - Un message construit sur l'opérateur *mpair* représenterait les informations relatives à chacune des sessions.
  - Complétude de la transformation asynchrone-synchrone.
- Extension des modèles à d'autres domaines d'applications ;
  - utilisant des listes ;
  - Exemple : les services web ;

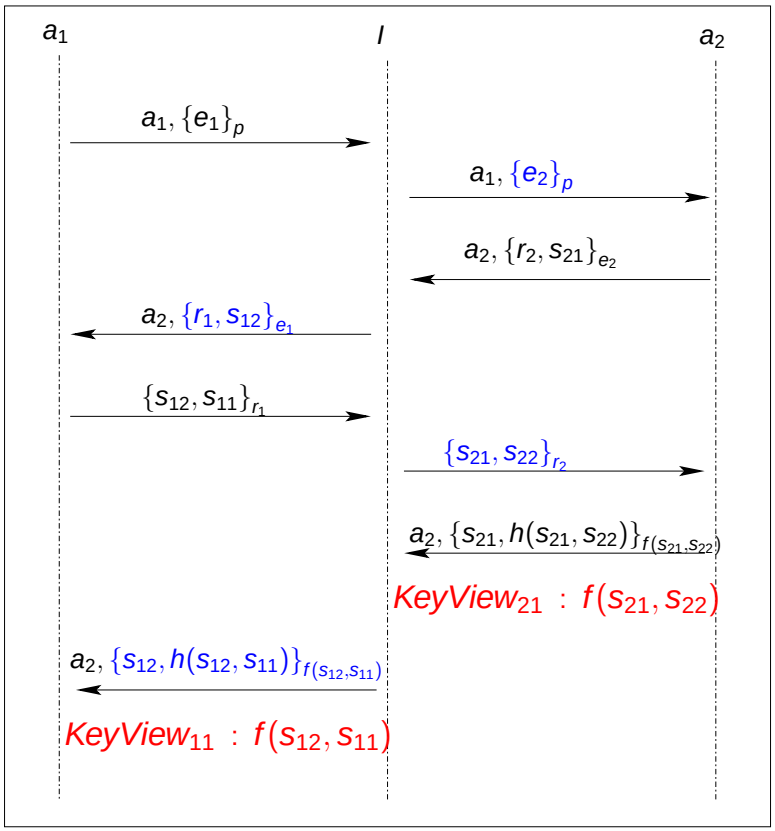
Attaque sur Asokan-Ginzboorg(Retour au bilan)



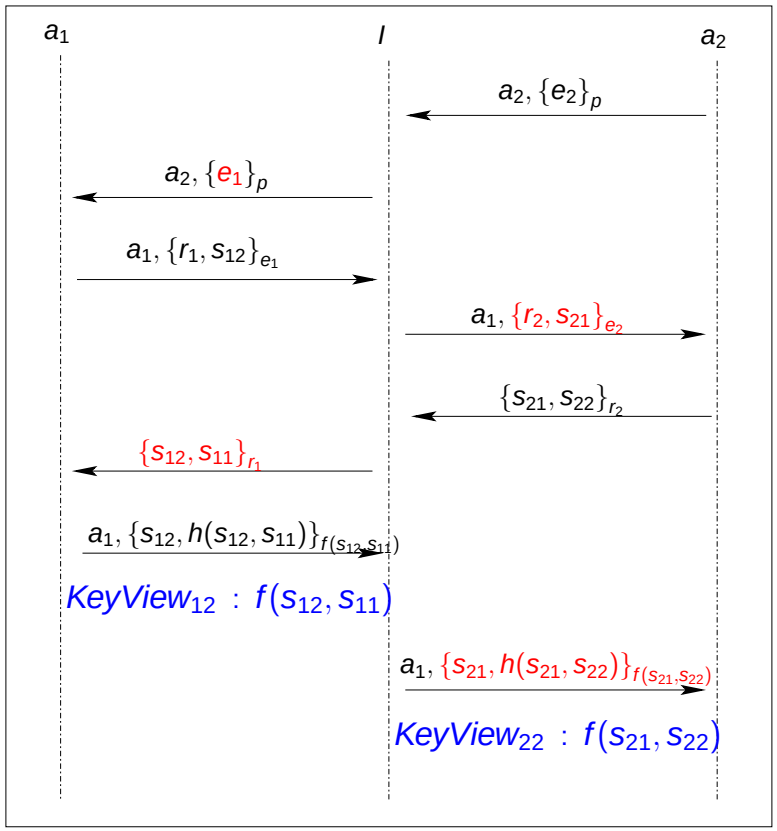
ce qui correspond à la solution σ suivante :

$$\begin{array}{lcl|lcl} X_2\sigma & = & r_1 & X_3\sigma & = & s_{12} \\ X_5\sigma & = & e_2 & X_6\sigma & = & s_{21} \\ X_7\sigma & = & s_{22} & X_9\sigma & = & r_2 \\ X_{10}\sigma & = & s_{21} & X_{10}\sigma & = & s_{21} \\ X_{12}\sigma & = & e_1 & X_{14}\sigma & = & s_{11} \end{array}$$

Attaque sur Asokan-Ginzboorg(Retour au bilan)

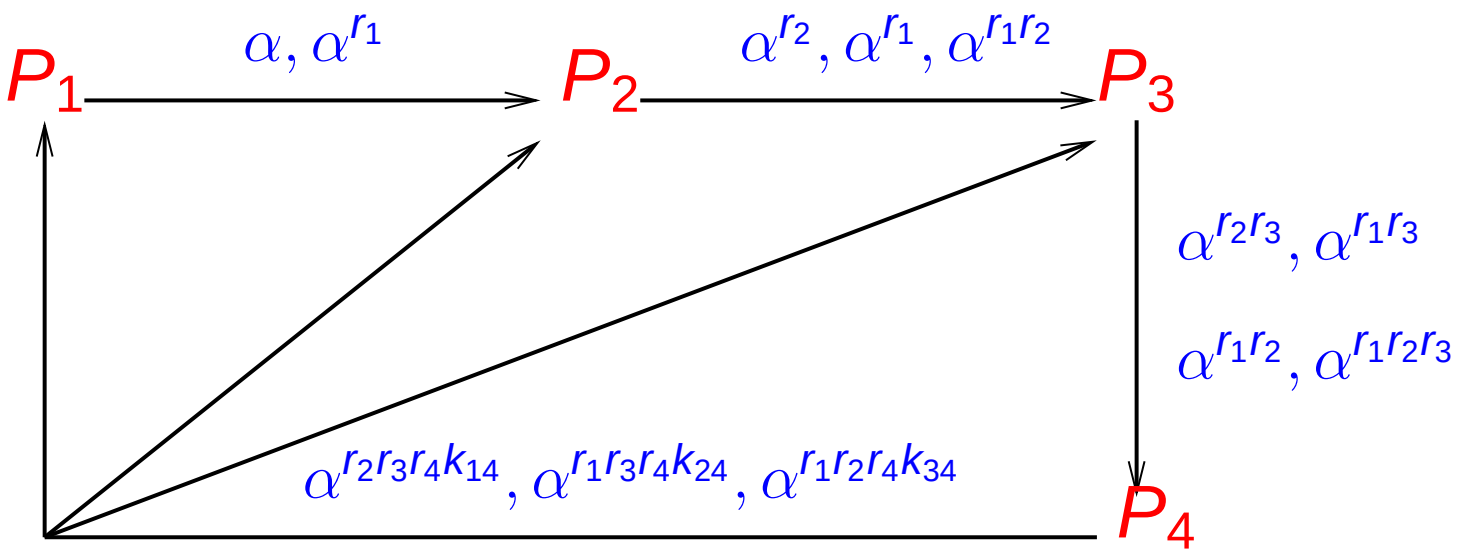


Première Session

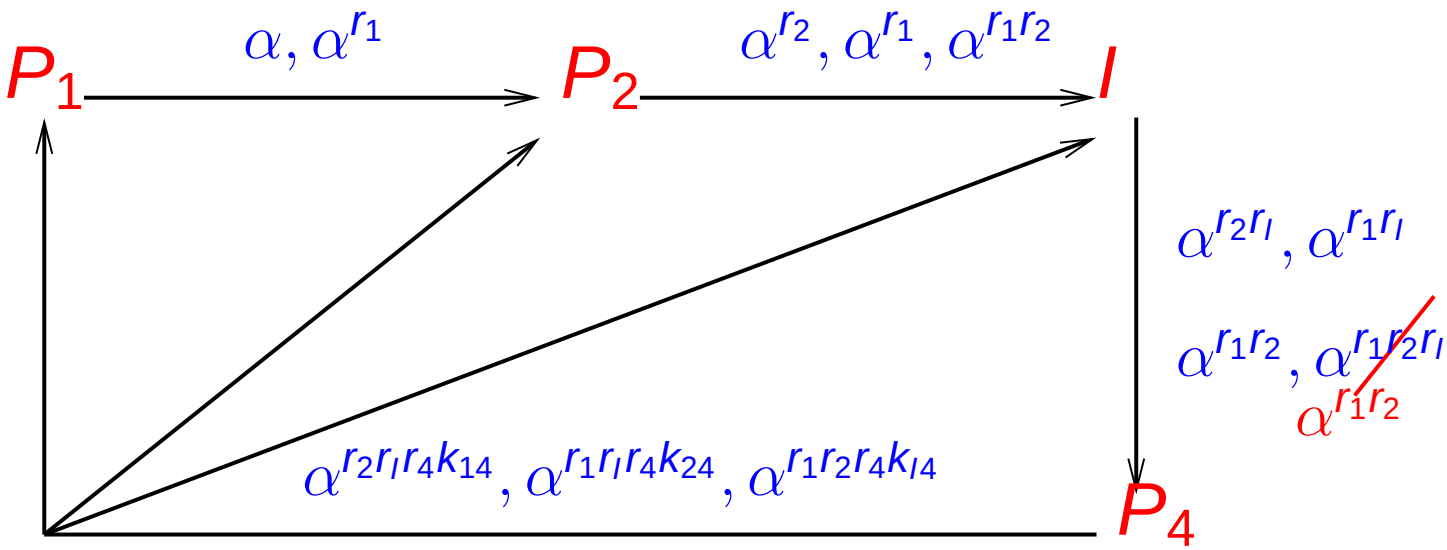


Deuxième Session

Attaque à quatre participants sur A-GDH (Retour au bilan)



Attaque à quatre participants sur A-GDH (Retour au bilan)



À la fin de cette exécution, les vues des participants sont :

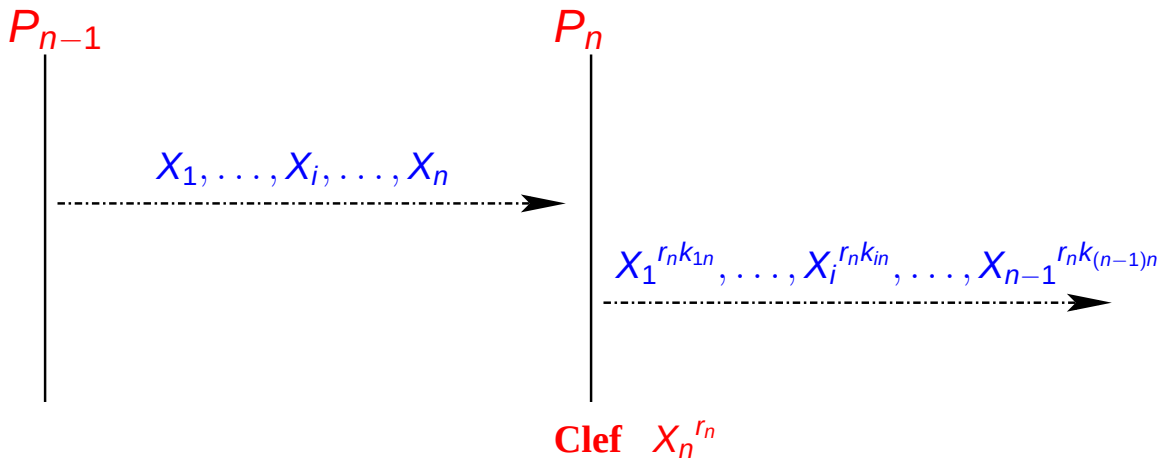
- Vue de P_1 : $\alpha^{r_1 r_2 r_I r_4}$.
- Vue de P_2 : $\alpha^{r_1 r_2 r_I r_4}$.
- Vue de P_4 : $\alpha^{r_1 r_2 r_4}$.
- Vues de I : $\alpha^{r_1 r_2 r_I r_4}$ et $\alpha^{r_1 r_2 r_4}$.

Généralisation de l'attaque sur A-GDH (Retour au bilan)

- A-GDH avec n participants $P_1, .., P_n$;
- L'intrus fait partie du groupe : $I = P_i$;
  - connaissance : k_{in} ;

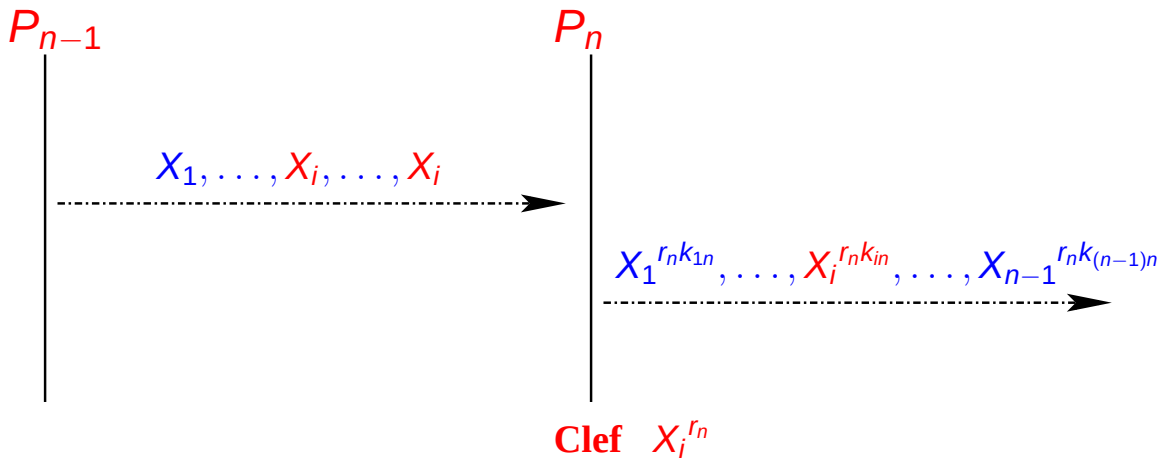
Généralisation de l'attaque sur A-GDH (Retour au bilan)

- A-GDH avec n participants $P_1, .., P_n$;
- L'intrus fait partie du groupe : $I = P_i$;
  - connaissance : k_{in} ;



Généralisation de l'attaque sur A-GDH (Retour au bilan)

- A-GDH avec n participants $P_1, .., P_n$;
- L'intrus fait partie du groupe : $I = P_i$;
  - connaissance : k_{in} ;



À la fin de cette exécution, l'intrus :

- a la vue de P_n : $X_i^{r_n}$;
- a la vue des autres participants : $X_i^{r_n r_i}$;

$\Rightarrow$ Division du groupe en deux parties.

Protocole PCP Retour

1.	$Init$	$\Rightarrow$	$a,\ b,\ 0,\ \{\langle 0,0\rangle\}_t$
2.	$mpair(i,\langle A_i,B_i\rangle)$	$\Rightarrow$	$mpair(i,\{\langle A_i,B_i\rangle\}_t)$
3.	$mpair(i,\{\langle X_i,Y_i\rangle\}_t)$	$\Rightarrow$	$mpair(i,\{\langle \alpha_1(X_i),\beta_1(Y_i)\rangle\}_u),\ldots,$ $mpair(i,\{\langle \alpha_p(X_i),\beta_p(Y_i)\rangle\}_u)$
4.	$mpair(i,\{\langle A_i,B_i\rangle\}_u),\{\langle Z,Z\rangle\}_u$	$\Rightarrow$	Sec

Test of Satisfiability of the solved form

Retour

- Entrelassements utilisant tous les patterns possibles ;
- Remplacement des contraintes $Forge_c$ par des égalités utilisant des termes différents de tous les patterns ;
- Recherche d'un entier e (dépendant du nombre de différents patterns, nombre de l'ensemble de connaissances, et du nombre de vecteurs de varibles) pour lequel la satisfaisabilité peut être vérifiée.