Sécurité et Temps dans les Systèmes Répartis

NAVARRO Stephen

Sécurité et Temps dans les Systèmes Répartis

- 1 – TERMINOLOGIE ET LIMITES IMPOSÉES
  - 1.1 Définitions
  - 1.2 Le droit
  - 1.3 La sécurité
  - 1.4 La problématique
  - 1.5 Sécurisation de l'ordre causal
- 2 – SETSAR
  - 2.1 Conception préliminaire
  - 2.2 Diagramme d'activité UML
  - 2.3 Réseau de Pétri
  - 2.4 Preuve

Stephen Navarro

Introduction

- La sécurité est un défi majeur pour Internet.
- Propriété de non répudiation de l'information.
- Le temps est considéré du point de vue de la relation de causalité qui existe entre messages.
- L'objectif est de prémunir des systèmes répartis d'attaques organisées par des opposants.

Définitions

- **Les systèmes répartis : actifs mis en commun, mode message asynchrone.**
- **La sécurité : protéger les actifs.**
- **Politique de sécurité : périmètre d'action, les droits, les attaquants, les défaillances.**
- **Propriétés : authentification, confidentialité, intégrité, non répudiation par origine, non répudiation par destination, l'auditabilité.**

Le droit procure l'arbitrage

- Aspect répressif :
  - Loi dite Godfrain Art. 323
- Aspect d'utilisation des NTIC :
  - Signature électronique Art. 1316
- Aspect du statut juridique :
  - Preuves numériques fournies par les programmes

Techniques cryptographiques utilisées (1/2)

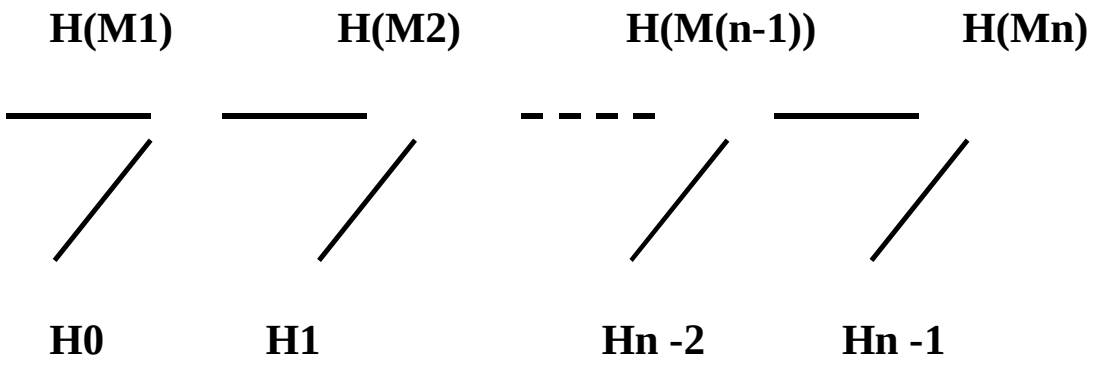
Algorithme à fonction de hachage sans clef.

- Point de vue fonctionnel :
  - Suite d'opérations à sens unique sans trappe;
  - Longueur du résumé toujours identique;
  - Fonction qui résiste à la collision;
  - 2 messages proches, 2 empreintes différentes;
  - Sécurité de cet d'algo liée à la longueur du résumé.

- Point de vue opérationnel :
  - Alice hache un message M1, obtient un résumé;
  - Alice envoie le message et le résumé;
  - Message = (M1 , $\{M1\}^{MD5}$)
  - Bob hache le message M1, obtient un résumé;
  - Bob compare le résumé avec celui d'Alice.

Techniques cryptographiques utilisées (2/2)

- Le nonce : numéro unique produit une fois.
- Etat de l'historique (h_state) : évolution.



Chaînage des résumés

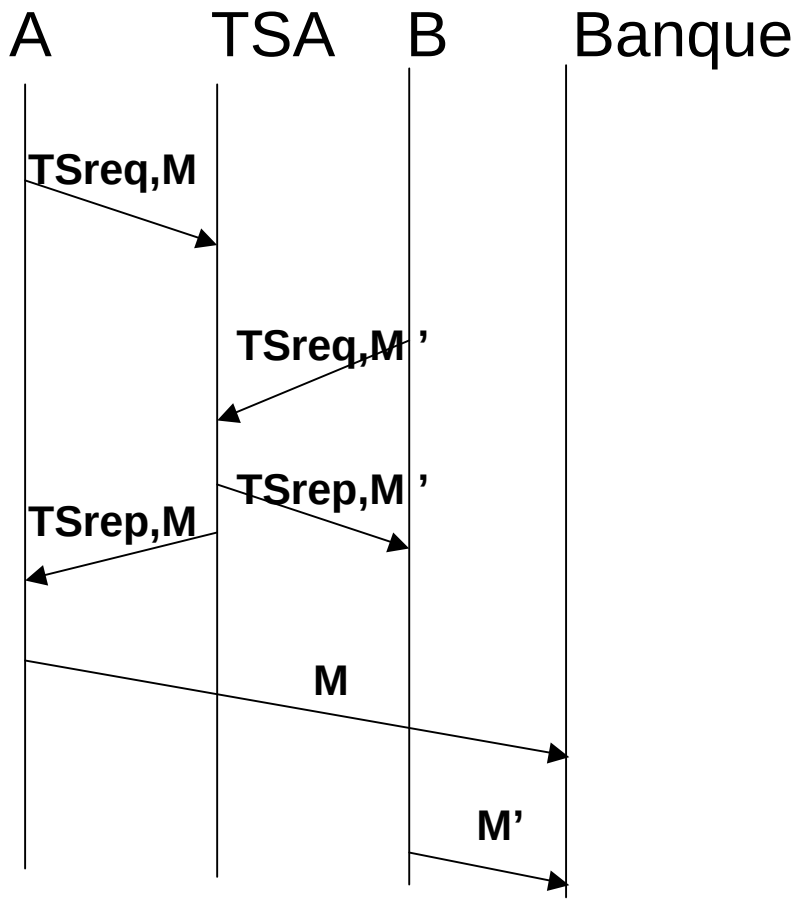
La problématique

- **Construire un protocole de sécurité :**
  - **qui assure à long terme la propriété de non répudiation par destination d'un document électronique;**
  - **qui date les événements à partir d'un marquage, en fonction de la relation d'ordre causal entre les messages;**
  - **qui permette la formalisation de contrats synallagmatiques sans intervention d'un tiers de confiance, sauf litige.**

La problématique par un exemple concret

Le risque résiduel :TSA n'est pas obligé de respecter l 'ordre causal
Estelle, TSA et la banque veulent mettre Bob en interdiction de chéquier

- **Alice demande à dater un ordre de virement M de 1ME de son compte C (solde 2ME) à au compte de Bob(solde 0): Elle prévient Bob**
- **Bob demande à dater un ordre de virement M'de 1ME à Estelle à partir de C'**
- **TSA date M' puis M**
- **La banque traite M' , fait dater M' par TSA et met Bob en interdiction de chéquier**



Stephen Navarro

Sécurisation de l'ordre causal

- Destruction de la causalité (DENIAL)
  - Prophylaxie par serveur de causalité
  - Prophylaxie par approche conservatrice
- Fabrication de la causalité (FORGERY)
  - Prévention par horloges vectorielles
  - Prévention par piggybacking

Preuve faible contre Preuve forte

- **Preuve faible : Dans un schéma de preuve faible, l'approche est passive. {A} peut prouver que dans les opérations qui lui incombent, les exigences du protocoles ont été respectées.**

- **Preuve forte : Dans un schéma de preuve forte, l'approche est active. {B} ne peut pas altérer les propriétés d'ordre sans que cela soit décelable.**

  - **Un principal peut endosser les deux rôles.**
  - **Son comportement a été byzantin.**

Sécurisation de l'ordre causal : notations

- C(m1,m2) est vrai si proc estime que m1 précède m2
- C(m1,m2) est faux si le processus estime que m1 ne précède pas m2
- R(b,m1) Bob reconnaît le message m1
- 7 R(b,h(m1)) Bob ne reconnaît pas le résumé (7 non)

- CAUS(X,A) est une preuve lors de d'exécution X de P que Alice peut affirmer vraie
- RECEPTION(X,RB) : pour toute émission de mi, correspond une réception de mi pour le récepteur B

- 1 – ÉTAT DE L'ART

- 2 – RÉSULTAT : Le protocole SETSAR

SETSAR : énoncé des besoins

- Fabriquer un protocole de communication.
- L'ajouter à un protocole P qui existe par ailleurs, l'ensemble forme S(P).
- Chaîner les messages et les résumés procure une signature de la relation de causalié entre les événements.
- N'importe quelle relation est signée, ce qui permet de prouver à posteriori que la relation a été celle-là.

Conception préliminaire

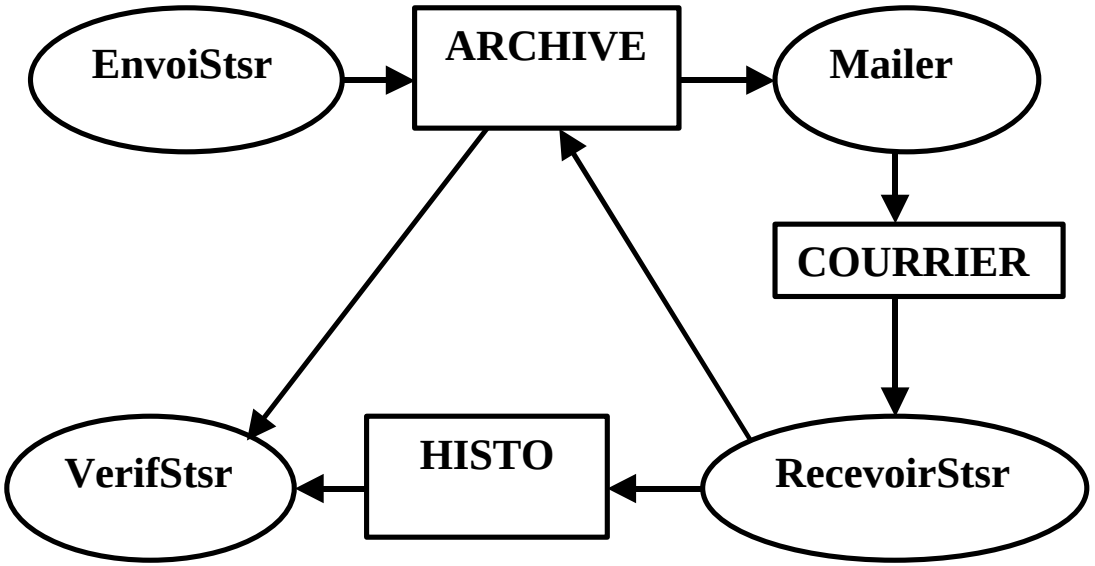
- La politique de sécurité : qui ? quels? quand?
- L'environnement technique : Internet, Mail.
- La fonction principale : contrat d'obligations.
- Les fonctions secondaires : vérifier.
- La fonction d'enregistrement : audit.
- La fonction de non modification de l'ordre : horodatage par chaînage des résumés.

Cas d’utilisation en mode unicast

- Alice et Bob veulent échanger des courriers par Mail
- Ils s'engagent à utiliser le protocole SETSAR
- Ils signent les spécifications du protocole
- En cas de litige, ils fournissent leurs données à une autorité de confiance

Stephen Navarro

Interfaces : structures de données

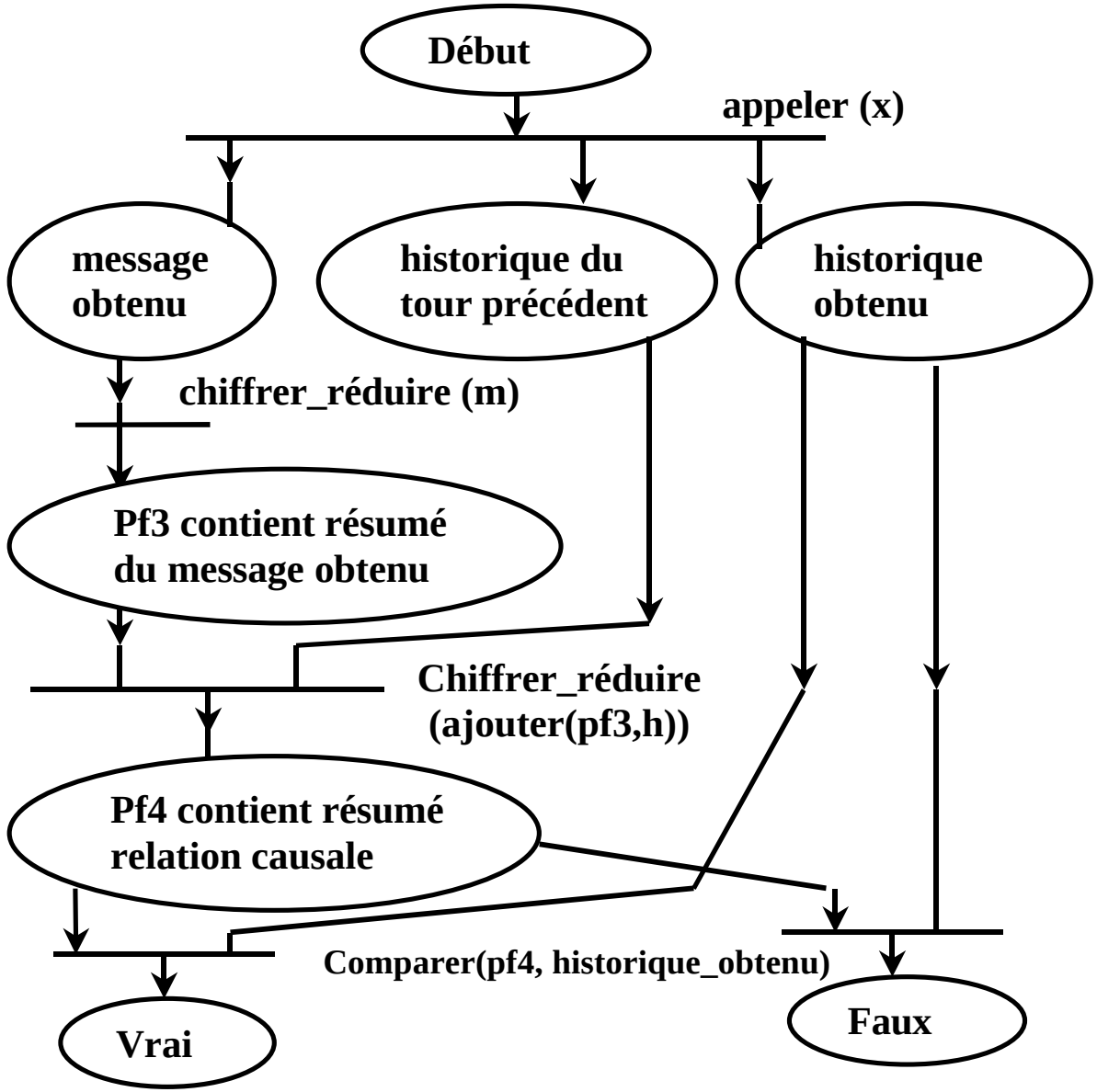


Matrice des droits

	courrier	archive	histo
Auteur	L , E	L , E	L
Destinataire	L	L , E	L , E
Vérificateur		L	L
Juge		L	L
Reste du monde			

Stephen Navarro

Réseau de Pétri de Verifstsr



Stephen Navarro

Preuve statique de VerifStsr

- Utilisation du formalisme des séquents de Gentzen : règles d'inférence.
- Des scénarios d'attaques sont formalisés.
- Une entité reconnaît ou ne reconnaît pas un ensemble de messages et de résumés.
- Ces données sont apparues dans cet ordre.

- La sécurisation de l'ordre causal est une propriété de SETSAR.
- Preuve faible pour la causalité et pour la réception. Preuve forte du protocole pour la relation de causalité.

Application JugementStsr

- En cas de litige entre Alice et Bob.
- Le juge détient l'ensemble des données fournies par chacun des protagonistes.

- Mise en évidence de la tricherie éventuelle.
- Désignation de l'entité qui a triché et des données non conformes.
- Oracle : ordre d'apparition de deux messages.

- Preuve dynamique de l'appli JugementStsr.

Conception détaillée et implantation

- Conception : diagramme des classes.
- Code : java 1.3.1 sous LINUX (3328 lignes).
- Procédures d'exploitation.
- Procédures de test pour Victor.
- Guides d'utilisation pour Alice et Bob.

Conclusion

- **SETSAR : archivage, adressage à soi-même, mécanisme d'enregistrement de l'ordre causal au moyen du chaînage des résumés.**

- **La sécurité fournie par SETSAR possède une propriété de non répudiation par destination.**

- **SETSAR peut servir à sécuriser d'autres protocoles qui existent par ailleurs.**

Sécurité et Temps dans les Systèmes Répartis

Stephen Navarro