Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms - By David Chaum

Présenté par Aurélien Monnet-Paquet

www.inria.fr

19 Mai 2016

Problème ?

- Comment garder confidentiel qui parle avec qui et quand ?

Problème ?

- Comment garder confidentiel qui parle avec qui et quand ?
- Comment se protéger de l'analyse du trafic réseaux ?

Notation

Une solution basée sur la cryptographie à clé publique

- K : Clé publique
- Inv(K) : Clé privée
- X : Message en clair

Notation

Une solution basée sur la cryptographie à clé publique

- K : Clé publique
- Inv(K) : Clé privée
- X : Message en clair
- $Inv(K)(K(X)) = K(Inv(K)(X)) = X$

Notation

Une solution basée sur la cryptographie à clé publique

- K : Clé publique
- Inv(K) : Clé privée
- X : Message en clair
- $Inv(K)(K(X)) = K(Inv(K)(X)) = X$
- R : Une chaîne de bits aléatoire

Notation

Une solution basée sur la cryptographie à clé publique

- K : Clé publique
- Inv(K) : Clé privée
- X : Message en clair
- $Inv(K)(K(X)) = K(Inv(K)(X)) = X$
- R : Une chaîne de bits aléatoire
- Le chiffrement du message par la clé publique K :

$$K(R, X)$$

Hypothèses

Hypothèse 1

Personne ne peut déterminer quoi que ce soit sur les correspondances entre un ensemble d'éléments chiffrés et l'ensemble des éléments non chiffrés, ou créer des contrefaçons sans la chaîne aléatoire appropriée ou la clé privée.

Hypothèses

Hypothèse 1

Personne ne peut déterminer quoi que ce soit sur les correspondances entre un ensemble d'éléments chiffrés et l'ensemble des éléments non chiffrés, ou créer des contrefaçons sans la chaîne aléatoire appropriée ou la clé privée.

Hypothèse 2

Chaque serveurs connaît le serveur d'origine du message qu'il reçoit, ainsi que le/les destinataires. Chaque serveurs peuvent injecter, modifier, supprimer des messages.

Mail system

Alice souhaite envoyer un message à Bob :

- Alice chiffre le message M avec la clé publique (Ka) de Bob
- Alice ajoute l'adresse (A) de Bob au résultat
- Alice chiffre le tout avec la clé publique (K1) du premier serveur

Alice souhaite envoyer un message à Bob :

- Alice chiffre le message M avec la clé publique (Ka) de Bob
- Alice ajoute l'adresse (A) de Bob au résultat
- Alice chiffre le tout avec la clé publique (K1) du premier serveur

$$K1(R1, Ka(R0, M), A) \rightarrow Ka(R0, M), A.$$

Alice souhaite envoyer un message à Bob :

- Alice chiffre le message M avec la clé publique (Ka) de Bob
- Alice ajoute l'adresse (A) de Bob au résultat
- Alice chiffre le tout avec la clé publique (K1) du premier serveur

$$K1(R1, Ka(R0, M), A) \rightarrow Ka(R0, M), A.$$

$$K1(R1, Ka(R0, M), A) \rightarrow Ka(R0, M), A.$$

- Le serveur déchiffre ce qu'il reçoit avec sa clé privée
- Puis enlève la chaîne R1 et délivre le reste

But d'un tel serveur :

- Cacher les correspondances entre les éléments en entrée/sortie
- Cacher l'ordre d'arriver en fournissant des éléments de taille uniforme en sortie
- Traiter une et une seule fois un lot

Si R1 contient un horodatage, le serveur ne conserve pas de copie des lots

Distribution en cascade

Comment ?

- Découper le message en parties égales
- Envoyer les morceaux aux serveurs

Kn(Rn, K<n-1>(R<n-1>, ... , K2(R2, K1(R1, C , A))...))

Avec C = Ka(R0, M)

Comment répondre en gardant l'adresse d'Alice anonyme ?

Former une adresse anonyme de retour :

$$K1(R1, Ax), Kx$$

- Ax est l'adresse réelle d'Alice

Comment répondre en gardant l'adresse d'Alice anonyme ?

Former une adresse anonyme de retour :

$$K1(R1, Ax), Kx$$

- Ax est l'adresse réelle d'Alice
- Kx est une clé publique choisie pour l'occasion

Comment répondre en gardant l'adresse d'Alice anonyme ?

Former une adresse anonyme de retour :

$$K1(R1, Ax), Kx$$

- Ax est l'adresse réelle d'Alice
- Kx est une clé publique choisie pour l'occasion
- R1 est une clé / chaîne aléatoire à des fins d'étanchéité

$$K1(R1, Ax), Kx(R0, M) \rightarrow Ax, R1(Kx(R0, M))$$

- $Kx(R0, M)$ est notre message chiffré

$$K1(R1, Ax), Kx(R0, M) \rightarrow Ax, R1(Kx(R0, M))$$

- $Kx(R0, M)$ est notre message chiffré
- $K1(R1, Ax)$ est le chiffrement de l'adresse d'Alice + la clé R1 avec l'adresse publique du premier serveur

$$K1(R1, Ax), Kx(R0, M) \rightarrow Ax, R1(Kx(R0, M))$$

- $Kx(R0, M)$ est notre message chiffré
- $K1(R1, Ax)$ est le chiffrement de l'adresse d'Alice + la clé R1 avec la clé publique du premier serveur
- $R1$ est utilisé pour re-chiffré $Kx(R0, M)$

En utilisant une cascade de serveurs

$$K1(R1, K2(R2, ..., A...)), Kx(R0, M)$$

- Avec A = $K < n-1 > (R < n-1 >, Kn(Rn, Ax))$

Alice peut recevoir un accusé de réception sur le message qu'elle envoie à Bob :

- L'adresse réelle d'Alice est étendu avec une autre adresse anonyme (pour Alice et Bob)

Alice peut recevoir un accusé de réception sur le message qu'elle envoie à Bob :

- L'adresse réelle d'Alice est étendu avec une autre adresse anonyme (pour Alice et Bob)
- Envoi de l'accusé une fois que le dernier serveur a envoyé le message

Alice peut recevoir un accusé de réception sur le message qu'elle envoie à Bob :

- L'adresse réelle d'Alice est étendu avec une autre adresse anonyme (pour Alice et Bob)
- Envoi de l'accusé une fois que le dernier serveur a envoyé le message
  - Adresse de livraison du message
  - Le message
  - Peut être signé par tous les serveurs intermédiaire

Digital Pseudonyms

Définition

Un "pseudonyme" numérique est une clé publique utilisée pour vérifier les signatures effectuées par le titulaire anonyme de la clé privée correspondante.

Digital Pseudonyms

Définition

Un "pseudonyme" numérique est une clé publique utilisée pour vérifier les signatures effectuées par le titulaire anonyme de la clé privée correspondante.

Comment ?

Une autorité est chargé de tenir à jour une liste de pseudonymes.

Digital Pseudonyms

Définition

Un "pseudonyme" numérique est une clé publique utilisée pour vérifier les signatures effectuées par le titulaire anonyme de la clé privée correspondante.

Comment ?

Une autorité est chargé de tenir à jour une liste de pseudonymes.

Pourquoi ?

Dialoguer avec une entité de manière anonyme tout en garantissant qu'on utilise pas plusieurs identités.

Chaque demande d'acceptation :

- Peut être soumis par le system de mail anonyme
- Doit contenir la clé publique et le pseudonyme souhaité
- Peut être pour une liste particulière soumise à un vote
- Peut être acceptée ou refusée par une autorité

Une liste contenant les couples (pseudonyme, clé publique) est publié par cette autorité

L'acceptation dans une liste particulière peut être soumise a un vote :

- Les autres membres de cette liste votent pour/contre
- Un vote est de la forme : $K1(R1, K, Inv(K)(C, V))$

L'acceptation dans une liste particulière peut être soumise a un vote :

- Les autres membres de cette liste votent pour/contre
- Un vote est de la forme : $K1(R1, K, Inv(K)(C, V))$
  - K = pseudonyme du votant
  - V son vote
  - Comptabilisé après avoir vérifié le pseudonyme du votant et la signature du vote

Une liste contenant les couples (pseudonyme, clé publique) est publié par cette autorité

General Purpose Mail Systems

- Chaque message devrait passer par une "cascade" de serveurs
- Les serveurs peuvent fonctionner de manière continue ou périodique
- Les messages (trop) longs sont divisés en plusieurs parties :
  - $Ka(R0, M) = M1, M2, ..., M < l - n >$

Un message qui passe par tous les serveurs intermédiaire a un coût non négligeable

- Manière 1 : sous-ensemble
- Manière 2 : valeur aléatoire
- Une séquence de serveur peut être définie pour un message

Conclusion

Solution efficace pour sécurisé les messages contre l'analyse du trafic réseau.
Permet également d'envoyer / recevoir des messages anonyme avec/sans pseudonyme.