Programmation

Cryptographie ...

Christian Quest - cquest@cquest.org

Jean-Pierre Milliet

Qui suis-je ?

- Auteurs de plugins 4D
  - Internet-ToolKit (ITK)
  - Stuff-ToolKit
  - Serial-ToolKit (STK)
  - PGP-ToolKit (expérimental)
- Développements pour ACI
  - Premiers composants réseaux de 4D Server (ADSP et TCP/IP)
  - 4D Remote
- Administrateur de 4d-forum, membre du comité de rédaction de Planète4D
- Expérience en cryptographie grâce à la version SSL d'ITK et à PGP-ToolKit

But de cet atelier

- Découvrir ce qu'est la cryptographie
  - Algorithmes cryptographiques
    - à clés symétriques
    - à clés asymétriques (clés publiques)
  - Signature électronique
  - Fonctions de hachage
- L'utilisation de la cryptographie
- La cryptographie et le développement 4D

Qu'est-ce que la cryptographie ?

- Vient du grec «kryptos» = «caché»
- Cacher des données, les rendre illisibles à celui qui n'a pas la «clé» pour les lire
- Cela va plus loin que simplement cacher des données:
  - Les signatures électroniques
  - Authentification

La cryptographie dans la vie courante

- Les distributeurs de billets
- Les téléphones mobiles
- Les cartes à puces
- Le commerce électronique
- Les numéros de série
- etc...

Terminologie

- Alice et Bob
- Décryptage(Cryptage(texteClair)) = texteClair
- La Cryptanalyse
  - «Casser» des messages cryptés
- Les clés: symétriques ou asymétriques
  - Symétriques (même clé utilisée pour crypter et décrypter)
  - Asymétrique (clé différente pour crypter et décrypter)

Algorithmes cryptographiques à clés symétriques et asymétriques

- Les clés symétriques
  - Elles correspondent à des «mots de passe»
  - La même clé est utilisée pour crypter et décrypter
  - La clé doit être échangée
- Les systèmes à clés asymétriques (ou "publiques"
  - Des clés différentes sont utilisées pour crypter et décrypter
  - Les clés ont deux parties, l'une "publique", l'autre "privée"
  - Seulement la partie "publique" des clés a besoin d’être échangée

Exemple d'utilisation : clé symétrique

- Alice crypte un message avec une "clé"
- Elle envoie ce message crypté à Bob
- Bob décrypte le message à l'aide de la même clé
- Alice a donc dû transmettre la clé à Bob
- N'importe qui ayant la clé peut décrypter le message, la clé étant transmise, le sécurité est assez faible

Principe des systèmes à clés publiques

- On génère une paire de clés (l'une privée, l'autre publique) liées entre elles mathématiquement
- Les clés privées servent à décrypter les messages et doivent donc rester confidentielles
- Les clés publiques sont rendues publiques pour permettre d'encrypter un message.

Exemple d'utilisation de clé publique

- Bob a une paire de clés (privée + publique)
- Bob donne sa clé publique à Alice
- Alice crypte un message destiné à Bob en utilisant la clé publique de Bob, Alice sait que seul Bob (qui a la clé privée correspondante) pourra décrypter le message
- Bob décrypte le message à l'aide de sa clé privée.

Algorithmes à clé symétrique

- Ils utilisent la même clé pour crypter et décrypter des données:
  - DES, TripleDES = Data Encryption Standard
  - RC2, RC4, RC5, RC6 (mis au point par RSA Labs)
  - IDEA = International Data Encryption Algorithm
  - Blowfish, Twofish
  - CAST (définis par des RFC)
  - Enigma (utilisé par la commande « crypt » sous Unix)
  - TEA = Tiny Encryption Algorithm
  - Le futur AES...

AES : Advanced Encryption Standard

- Le futur remplaçant du DES
- En cours de sélection par le NIST (National Institute of standards and Technology)
- Algorithmes en lice:
  - MARS (IBM)
  - RC6 (RSA Labs)
  - Rijndael
  - Serpent
  - Twofish

Algorithmes à clés asymétriques (publiques)

- Ils utilisent une paire de clés privées/publiques
  - RSA = Rivest Shamir Adelman
  - Diffie-Hellman (utilisé pour l ’échange de clés)
  - ECC = Elliptic Curve Cryptosystems
  - DSA/DSS = Digital Signature Standard
  - El Gamal
  - LUC (basé sur les fonctions de Lucas)

Signatures électroniques

- Les signatures électroniques sont l'une des utilisations des systèmes à clés publiques
- Deux buts:
  - Certifier que des données n'ont pas été modifiées (authentification du contenu)
  - Certifier que des données proviennent bien du «signataire» (authentification de la provenance)
- Les données ne sont pas cryptées
- N'importe qui peut vérifier la signature

Exemple de signature électronique

- Alice a une paire de clés (privée + publique)
- Alice signe un message en utilisant sa clé privée
- Bob vérifie (grâce à la clé publique d'Alice) que c’est bien Alice qui avait signé le message

Standards de signature électronique

- DSA/DSS = Digital Signature Algorithm
- PGP = Pretty Good Privacy

Cryptage + signature électronique

- On peut combiner le cryptage d'un message et la signature électronique.
- L'émetteur **identifiera** le destinataire
- Le destinataire **authentifiera** l'émetteur

Exemple

- Alice a une paire de clés (privée + publique)
- Bob a une autre paire de clés (privée + publique)
- Alice et Bob s'échangent leurs clés publiques

- Alice signe un message en utilisant sa clé privée et le crypte avec la clé publique de Bob

- Bob décrypte le message à l'aide sa clé privée et vérifie (grâce à la clé publique d'Alice) que c’est bien Alice qui avait signé le message

Identification et Authentification

- Lorsque Alice signe son message elle le **signe** à l'aide de sa clé privée.
- Lorsque Bob décrypte son message, il est **identifié** à l'aide de sa clé privée
- Bob peut **authentifier** la provenance du message grâce à la clé publique d'Alice
- Les clés publiques peuvent être stockées sur des serveurs accessibles à tous, encore faut-il pouvoir être sûr de leur provenance...

Confiance dans les clés publiques

- Indispensable pour être sûr du propriétaire d'une clé publique
- La provenance d'une clé publique doit être certifiée pour pouvoir lui faire confiance
- Comment «certifier» l'authenticité et la provenance d'une clé publique ?
  - Par une autorité de certification
  - Par un réseau de confiance

Les autorités de certification

- Ces autorités (sociétés, organisations) certifient les clés publiques
- Une clé est certifié si elle a été signé par une autorité de certification reconnue
- Exemples d'autorités de certification
  - Verisign
  - Thawte
  - Equifax
  - GlobalSign
  - Etc...

Certification par réseau de confiance

- Aucune autorité définie
- On a confiance en une clé si elle a été signée par une personne en qui on a confiance, exemple:
  - Alice signe la clé de Bob,
  - Bob envoie sa clé à Chris,
  - Chris fait confiance à cette clé car elle a été signée par Alice et que Chris a confiance en Alice
  - Chris peut maintenant signer la clé de Bob, etc...

Systèmes à clés publiques: résumé

- Une clé contient une partie publique et une partie privée
- La clé privée est utilisée pour signer ou pour décrypter
- La clé publique est utilisée pour vérifier la signature ou pour crypter
- La clé publique doit être certifiée si on veut pouvoir lui faire confiance :
  - Clé signée par une autorité de certification
  - Clé signée par un réseau de confiance

Fonctions de «hachage» Le principe

- C'est la transformation de données de taille quelconque en un résultat de taille fixe (le "digest")
- C'est une opération à sens unique, les données originales ne peuvent pas être récupérées
- Les "digests" sont comme une "empreinte" correspondant aux données d'origine

Fonctions de «hachage» Utilisation

- Utilisées pour la signature électronique pour créer un "digest" du message à signer et garantir qu'il n'a pas été modifié (sorte de "super-checksum")
- Utilisé pour l'authentification (MAC = Message Authentication Code)
  - Par la commande APOP du protocole POP3
  - Par la commande AUTH du protocole SMTP
  - Par HTTP pour les « digest d 'authentification »

Fonctions de «hachage» Les algorithmes

- MD2, MD4, MD5 = Message Digest 2,4,5 (de RSA Labs)
- SHA = Secure Hash Algorithm
- SHA-1 (ou SHS) = version modifiée (corrigée) de SHA
- RIPEMD = RACE Integrity Primitives Evaluation (projet européen)
  - Il existe 2 versions de RIPEMD, RIPEMD128 et RIPEMD160 qui génèrent des digests de tailles différentes (128 ou 160 bits).

Fonctions de «hachage» Niveau de sécurité

- Les digests de 128bits sont considérés comme faibles (MD2, MD4, MD5, RIPEMD128)
- Les digests de 160 bits sont considérés comme sûrs (SHA-1, RIPEMD160)
- SHA et MD4 sont considérés comme faibles à cause de certaines failles dans leur conception.

Algorithmes déposés

- Plusieurs algorithmes sont déposés et font donc l'objet de licences d'utilisation:
  - Algorithmes RSA (protégés jusqu'en Septembre 2000)
  - IDEA
- Algorithmes non déposés ou dont le dépôt est expiré:
  - DES (expiré en 1993)
  - Diffie-Hellman
  - ECC
  - RIPEMD
  - SHA et SHA-1
  - Blowfish, Twofish, CAST, MARS, etc...

Utilisation de la cryptographie

- Chiffrage (Cryptage/Décryptage)
  - Stocker des données cryptées
  - Crypter des communications
- Signature électronique et fonctions de hachage
  - Authentification de la provenance de données
  - Contrôle de l'intégrité des données (authentification du contenu)
  - MAC (Message Authentication Code)

Internet et la cryptographie

- Transmissions sécurisées
  - SSL, TLS
  - VPN (Virtual Private Networks) = Réseaux Privés Virtuels
- Authentification
  - Utilisé par POP3, SMTP, HTTP, S-HTTP, HTTPS, etc...
- Sécurisation des contenus
  - S/MIME (email sécurisé)
  - PGP et OpenPGP

Transmissions sécurisées

- SSL: Secured Socket Layer
  - Popularisé à l 'origine par Netscape
  - Fournit une communication sécurisée au niveau « transport »
  - Utilisation de certificats pour identifier chaque extrémité (le serveur, et parfois le client)
  - Basé sur les algorithmes RSA (problème légaux)
  - Deux versions SSLv2 et SSLv3
- TLS: Transport Layer Security (RFC#2246)
  - Un remplacement à SSL
  - Basé sur SSLv3
  - Moins de problèmes légaux (ne dépend pas des algorithmes déposés par RSA)

Authentification & Identification

- Authentification des sites Web sécurisés par HTTPS (HTTP sur SSL) à l ’aide de certificats
- Identification possible des clients web par HTTPS et des certificats côté client
- Utilisation de SHA et MD5 dans les protocoles POP3, SMTP
- Identification des utilisateurs par HTTP (Digest Authentication)

Contenu sécurisé

- Encryptage des fichiers
  - PGP
- Encryptage des courriers électroniques
  - S/MIME
  - PGP et OpenPGP

Qu'est-ce que PGP ?

- Ce n'est pas un algorithme de cryptographie
- C'est un format "standardisé" de messages cryptés
- PGP utilise des algorithmes "standards" comme DES, TripleDES, CAST, Diffie-Helman, etc…

La cryptographie dans 4D

- Propriétaire ou standard ?
- Algorithmes de chiffrage écrits avec 4D
- Plugin de cryptographie
- Signature électronique
- Fonction de hachage

Propriétaire ou standard ?

- Les algorithmes propriétaires de cryptographie sont généralement peu fiables mais suffisants pour des besoins simples
  - Exemple: stockage de mots de passe cryptés -> pas besoin d 'utiliser un algorithme standard
- Les algorithmes standards sont nécessaires si votre code ne se trouve pas « à l'autre bout »
  - Exemple: envoi d'un email crypté -> vous devrez respecter un standard si vous voulez que le destinataire puisse décrypter votre message !

Algorithmes de cryptage en code 4D

- Beaucoup de calcul sur des entiers
- Respect strict du typage des variables (ne surtout pas mélanger «réels» et entiers).
- Exemple: Tiny Encryption Algorithm
  - Algorithme conçu par l'Université de Cambridge
  - Disponible sous forme de code 4D
  - Malheureusement algorithme pas vraiment «standard»
- Utilisation de la compression comme cryptage
  - La compression de données rend les données illisibles.

Plugins de cryptographie

- Xcrypt (support de DES)
- DataProtector (SHA-1 et DSA)
- PGP-ToolKit (support de PGP)
- Internet-ToolKit v2 (MD2,MD4,MD5, SHA, SHA-1, RipeMD)
- Internet-ToolKit v2.5 (SSL+TLS, DES, TripleDES, CAST, etc)

Plugins de cryptographie Xcrypt (Telekynetics/C4i)

- Plugin offrant le support du DES classique
- Clés de seulement 56bits
- Peut être adapté pour faire du triple DES
  - EEE3 = encryption avec 3 clés = 3x56 = 168 bits
  - EDE3 = Encrypt/Decrypt/Encrypt avec 3 clés

Plugins de cryptographie DataProtector (Highwinds Trading Company LLC)

- Supporte le hachage « SHA-1 »
- Supporte la signature électronique de type DSA/DSS

Plugins de cryptographie
PGP-ToolKit (Ch. Quest)

- Nouveau plugin (encore expérimental)
- Support direct de PGP dans 4D
- Cryptage/Décryptage
- Signature/Vérification de signatures

- DEMO

Plugins de cryptographie Internet-ToolKit v2 (Ch. Quest)

- Fonctions de hachage
  - MD2, MD5
  - SHA, SHA-1
  - RIPEMD
- Peut être utilisé pour le support des authentifications dans certains protocoles Internet:
  - Commande APOP de POP3 (RFC#1939)
  - Commande AUTH de SMTP (RFC#2554)
  - Authentification par « digest » HTTP (RFC#2617)
  - Répartition de charge via QuickDNS Pro

Plugins de cryptographie Internet-ToolKit v2.5 (Ch. Quest)

- Support de SSL et TLS
  - Versions 40 et 128bits
  - Modifications minimes du code actuel utilisant ITK
  - Peut fonctionner comme serveur ou client (permet d'accéder à un serveur sécurisé comme un centre de paiement)
  - Ne dépend pas du futur support SSL de 4Dv6.7 (fonctionne donc avec 4Dv5.5, 4Dv6, 4Dv6.5)
- Nouvelles routines de cryptographie (DES, TripleDES, CAST, etc)
- DEMO

Questions/Réponses