Authentification

- Identification – Authentification – Autorisation - Audit
- Que sécuriser ? SSO
- Facteurs d'authentification

- Technologies

  Username – Password

  Windows – Linux

  Résistance d'un mot de passe, *brute force* → Labo

  Challenge – Response → Labo

  Token SecurID, Activ Card

- S/KEY (en réserve)

Identification – authentification – autorisation

- Identification

  Qui êtes-vous ?

- Authentification

  Prouvez-le !

  Qui prétendez-vous être ?

- Autorisation → chap Windows

  Mécanisme de contrôle d'accès : qui accède à quoi ?

  Utilisateur autorisé à accéder à une ressource, application, …

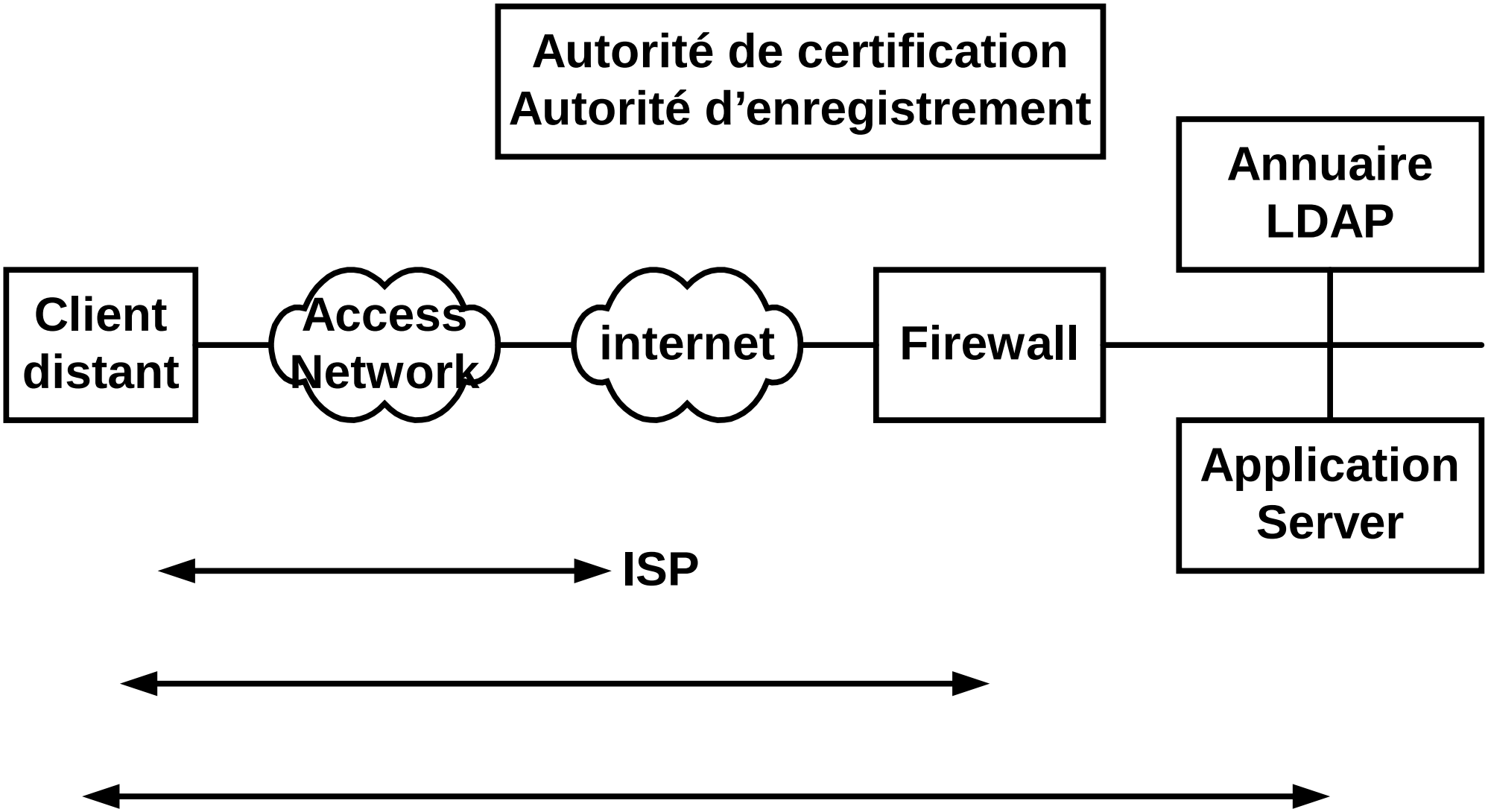
- Audit → chap Windows

  Surveillance – incident dans fichier log

Que sécuriser ?

- Equipements réseaux (routeurs, *switches*, …)
- Systèmes d'accès aux réseaux *(remote access, firewall, ...)*
- Serveurs du système d'information (Unix, Windows, AS400, …)
- Postes de travail (Windows, Mac, …)
- Applications (portail *web,* …)
- …

Diverses authentifications → Single Sign On (SSO)



Les 3 facteurs d'authentification

- *Something you know* — mot de passe, PIN, …
- *Something you have* — *token,* carte à puce, …
- *Something you are* — empreinte, iris de l'œil, …
  → procédés biométriques

Authentification forte si au moins 2 types différents sont utilisés : *token* + PIN, biométrie + *smartcard*, …

Degré de confiance ?

Technologies

- *Username – password*
- *Challenge Response*
- *Token*

Mot de passe : maillon faible ?

L’imposteur obtient le mot de passe :

- par téléphone *(social engineering)*
- sur le *Post-it* collé à l’écran
- avec un logiciel de *crack* utilisant un dictionnaire
- …

Norme BS 7799 → ISO 17799

- §9.2.3 *User password management*

  L'utilisateur s'engage (signe) à ne pas transmettre son mot de passe et à ne pas le mémoriser en clair dans l'ordinateur

- §9.3.1 *Password use – User responsibilities*

  Garder le mot de passe secret, le changer en cas de doute, le choisir en respectant 6 lettres aléatoires au min (xZ7s/1) et le changer régulièrement ou en fonction du nombre d'accès

Username – password

- Identification → *Username* (ID est unique)

  prédéfinis : administrator (Windows), root (Linux), cisco, guest …

  personnels : bob@company.com,

  standardisés : qevtwxs@, myname, …

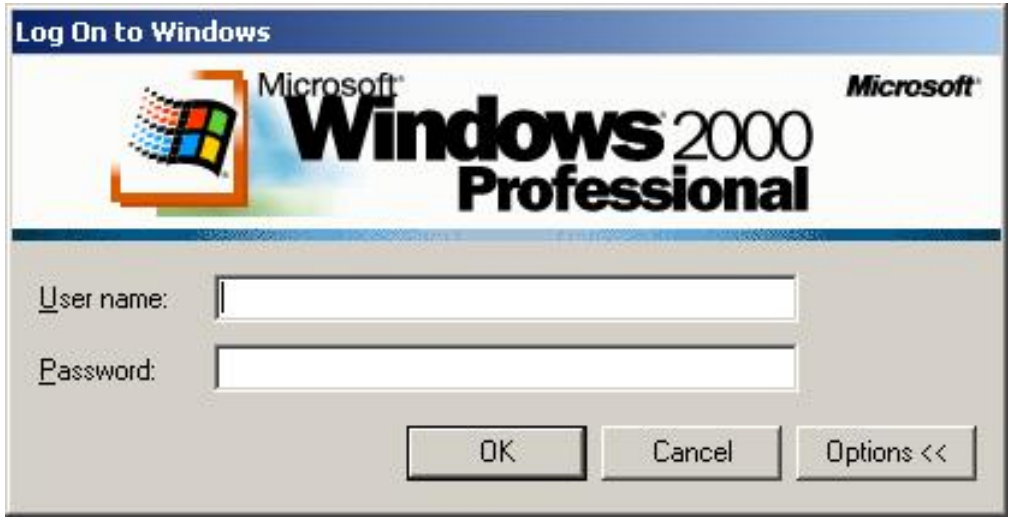
  banalisés : alias, …

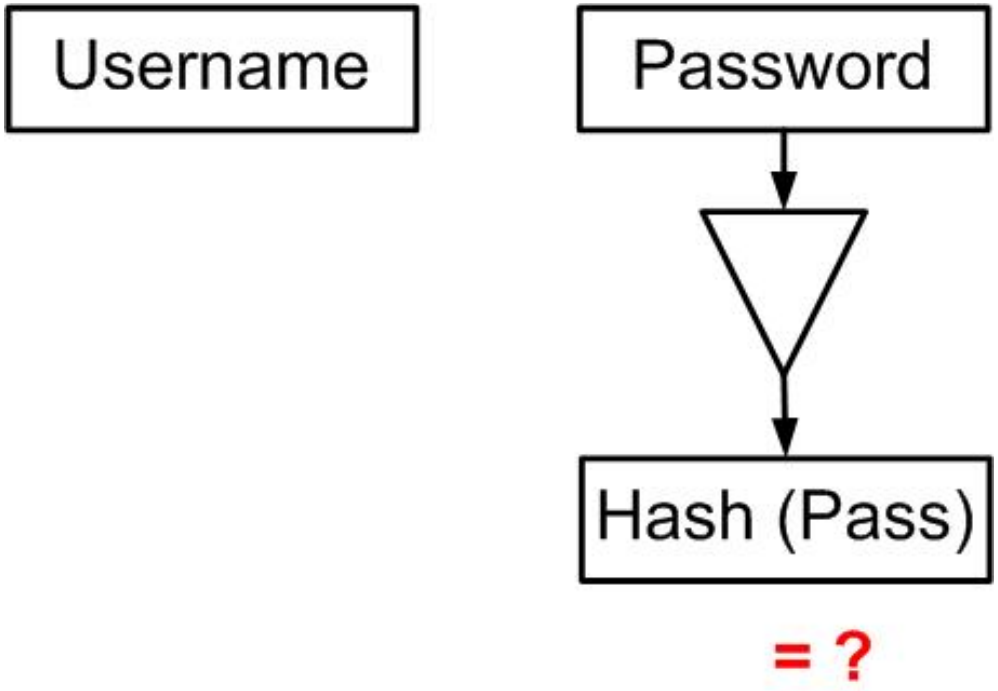
- Authentification → Mot de passe et facteur humain

  Créer des mots de passe efficaces et risquer de ne pas s'en souvenir

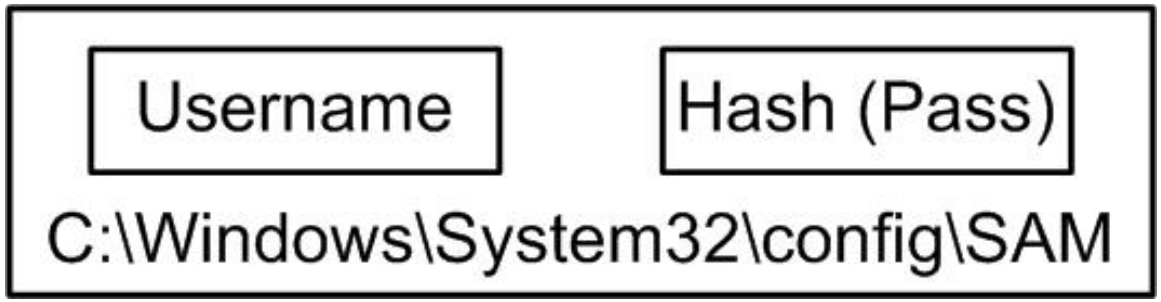
  Créer des mots de passe faciles à mémoriser mais inefficaces (attaques par dictionnaire, permutation, …)

Windows Logon Process (suite dans Windows System)





- Le mot de passe n'est pas stocké en clair (trop dangereux), mais son *hash*
- Fichier SAM contient *Username – hash (password)*
- Démo **`D:\pwdump7`**



Crack d'un mot de passe

- La fonction de hachage étant irréversible, craquer un mot de passe consiste à le deviner

- Il peut se trouver dans un dictionnaire

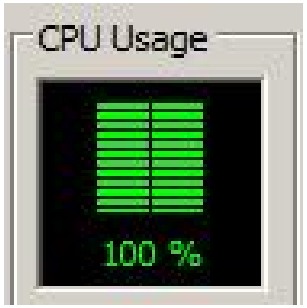
  Exemple = 123, voiture, oiseau, …

  Un bon mot de passe n'est pas dans un dictionnaire !

- Il peut être obtenu à l'aide d'un mot du dictionnaire auquel on ajoute un(des) caractère(s)

  Exemple = voi4ture

- Il peut être obtenu par méthode dite de *brute force* qui va tester toutes les combinaisons possibles de 1,2,3,4,5,6,… caractères. Labo avec Cain (www.oxyd.it)



La méthode Brute Force demande du temps

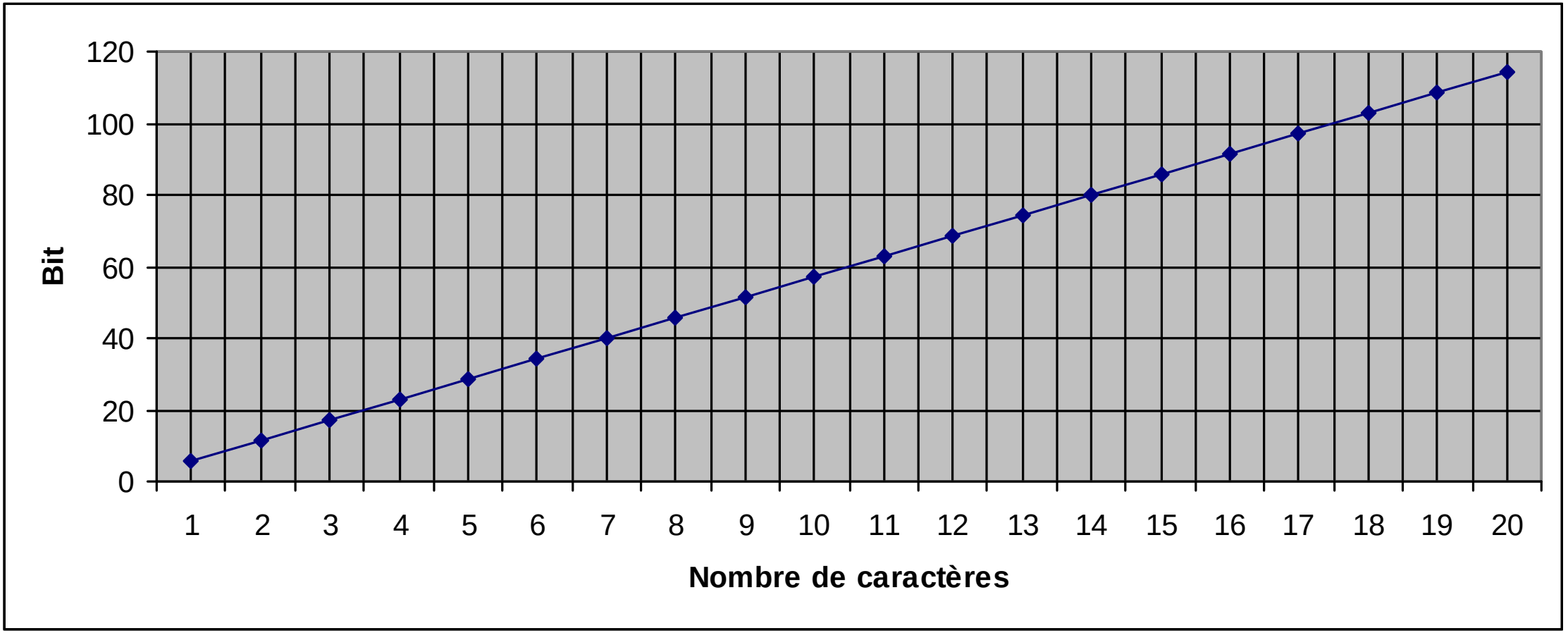
	Jeu	L=6	L=7	L=8
Maj	26	2,6 min	1,1 h	29 h
+Min	52	2,8 h	6 j	309 j
+chiffre	62	8 h	20 j	3,5 a
+spec	92	3,5 j	323 j	81 a

- Nombre de combinaisons = J^{L} = J ^ L
- Hyp : puissance de calcul = 10^{6} [combinaison/s]
- Temps total [minute, heure, jour, année] = J ^ L / P
- Temps moyen = ½ Temps total

Longueur de clé équivalente K

- K = log (nb de combinaisons) / log 2

- Pour un jeu de caract. = 52 a..z A..Z



Exercice

- Quelques puissances de calcul en combinaison / s

10^6	PC utilisant un logiciel standard
10^{10}	10’000 PC mis en parallèle
$2.45\ 10^{11}$	Deep crack ($ 460’000) en jan 99
$5.30\ 10^{14}$	500 ordinateurs les plus puissants

- Déterminer le temps moyen pour casser un mot de passe d’une longueur de 8 caract. utilisant maj, min et chiffres

Lab_Hacking : §1 Username – password (20 min)

Résistance d'un mot de passe

1 Créer 2 comptes utilisateurs

2 Afficher le contenu du fichier SAM avec pwdump

3 Mots de passe d'un dictionnaire

4-6 Outil Cain : SAM, dictionnaire, *brute-force attack*

7 Puissance de calcul de Charly (à domicile)

8 Résistance du mot de passe (à domicile)

9 Longueur du mot de passe (à domicile)

Outil (slide 15) → http://www.tdeig.ch/xxx/CryptMe.xls L

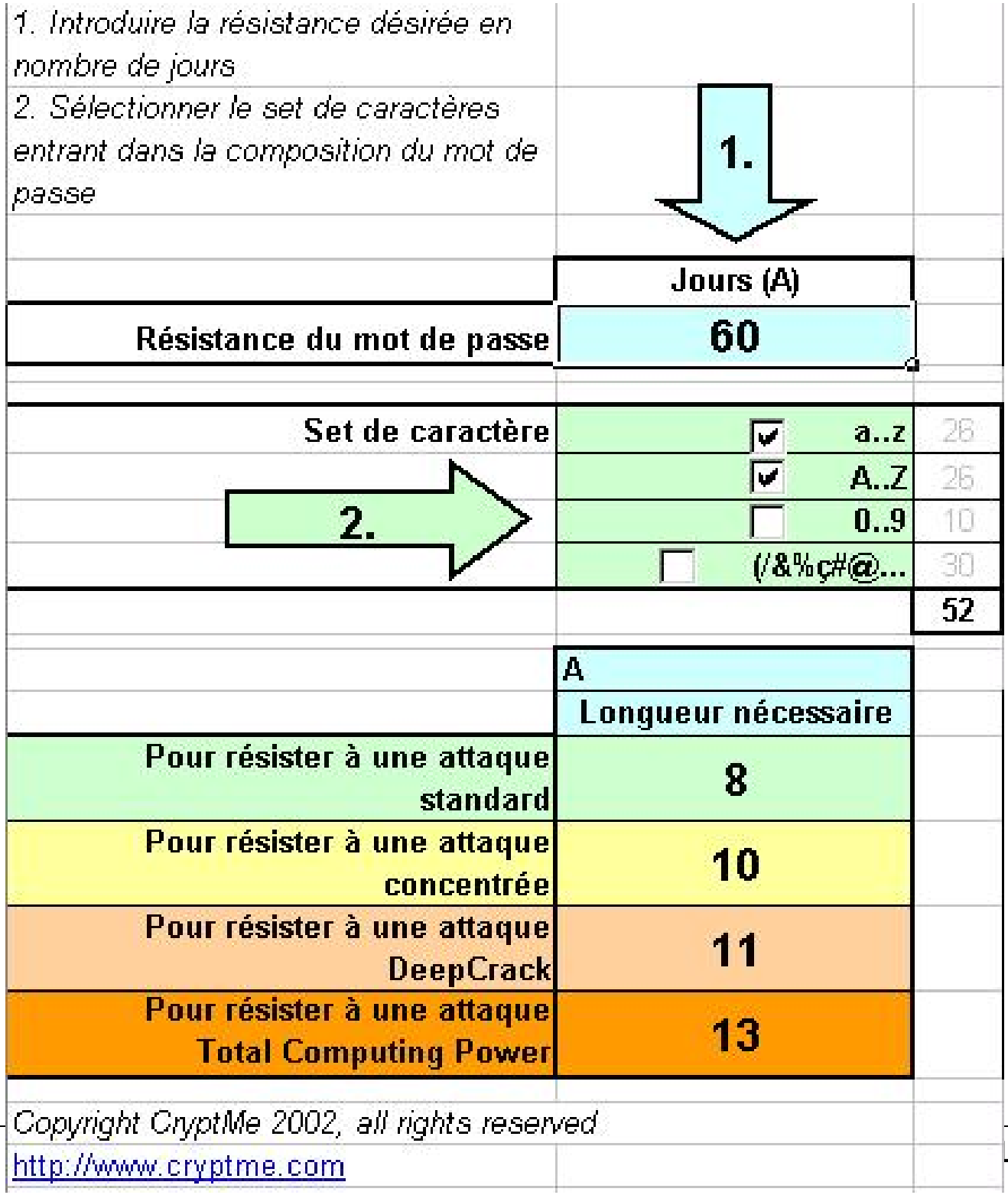
- **Ce travail exige une préparation**

  Paragraphes 2.1, 3.1.1, 3.1.2, 3.1.4, 4.4, 5.3 et 5.4 du document

  http://www.tdeig.ch/windows/Authentification_LM_NTLM.pdf

Résistance

- **Jeu de caractère**
- **Longueur**



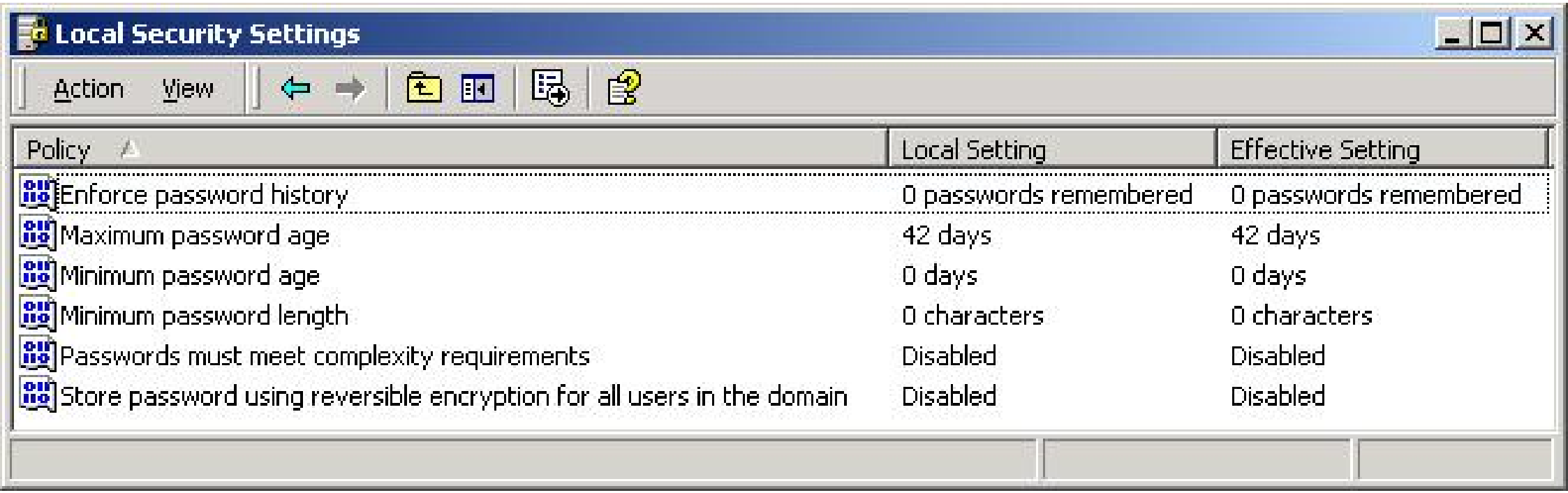
Mots de passe Linux

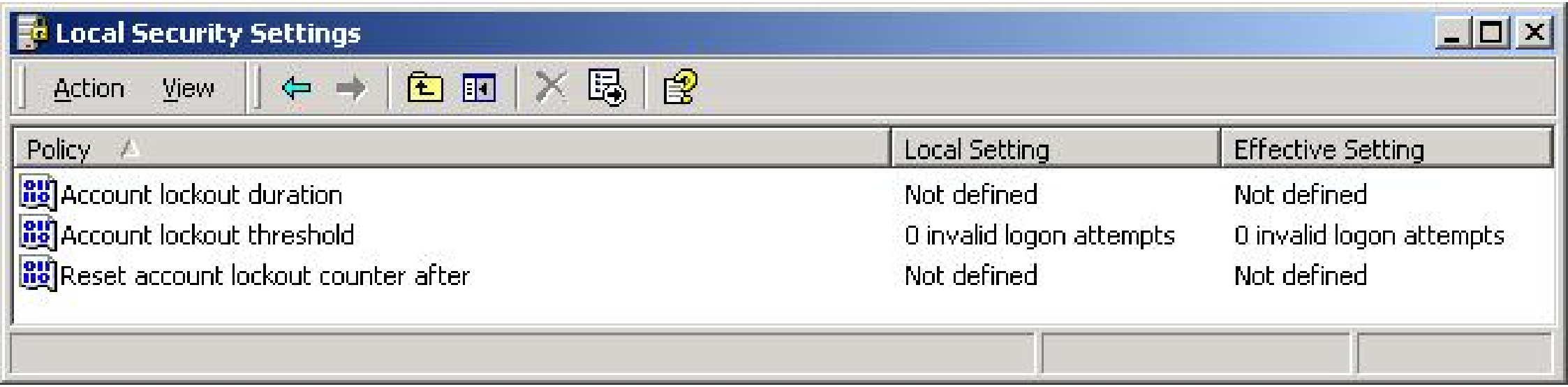
- Linux ajoute du sel *(salt)* au mot de passe pour le rendre plus sûr

- Les mots de passe sont stockés dans /etc/shadow (droit admin)

```
eig:$1$uc9/n2Et$tREFeGoessdiN4IOCPsnp1:
eig = username
   : = separator
    $1$ = hash = MD5
       uc9/n2Et = salt
                $ = separator
                  tREFeGoessdiN4IOCPsnp1 = hash
```

Local Security Policy (suite dans Windows Security)

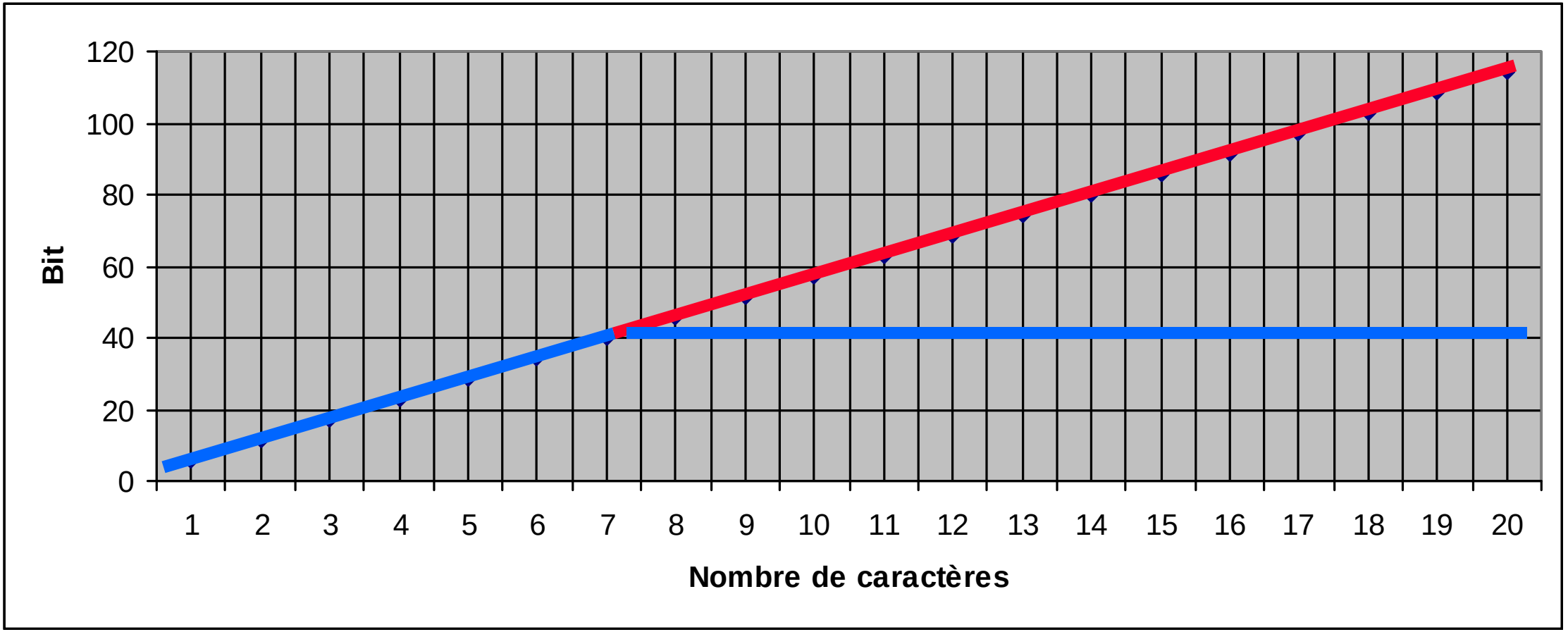




Efficacité des mots de passe

- **Mots de passe générés par l'être humain**

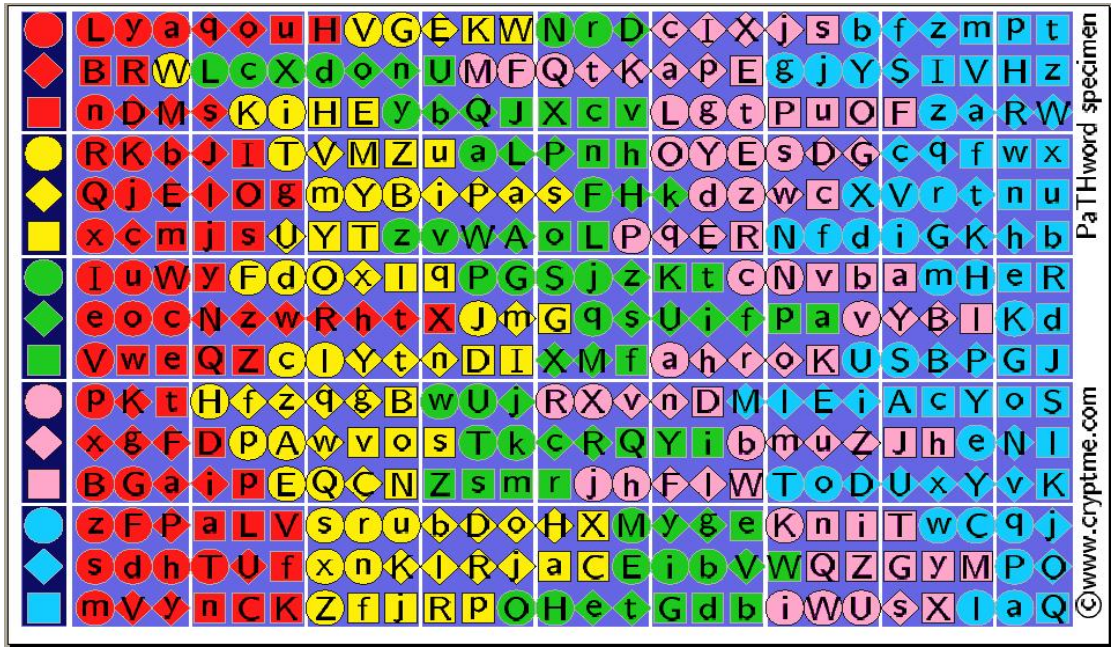
- **Mots de passe entièrement aléatoires**



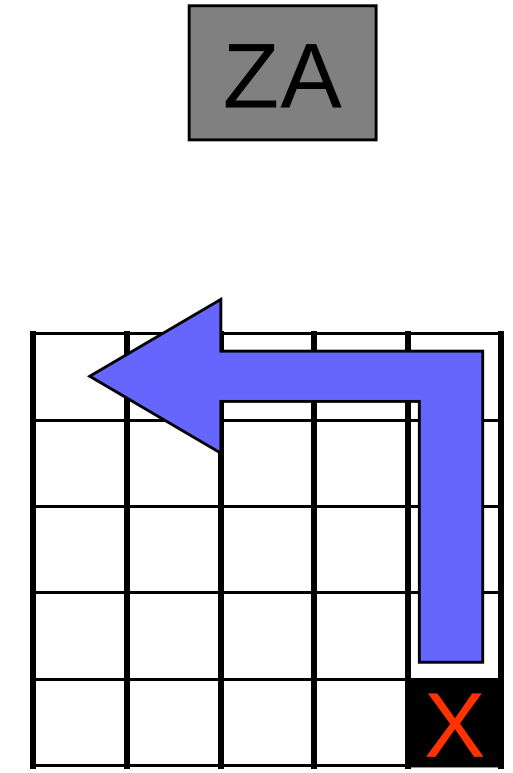
CryptMe PaTHword (1)

Carte unique personnelle

pour 15 systèmes différents

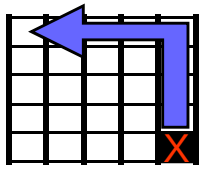


Méthode personnelle



CryptMe PaTHword (2)

PaTHword ™



ZA

Ecouter – espionner (Sniffing)

- *Sniffer* Ecouter le trafic sur le réseau

  Facile sur *shared media network* et *wireless* LAN

  Possible sur *switched network*

  Possible sur interface normalisée (USB, RNIS, …)

- Info sensibles *username-password* en clair

  telnet, ftp, pop, snmp, http, nntp, icq, irc,
  socks, nfs, mountd, rlogin, imap, x11, ICA,

  SMB, Oracle SQL, … :
  email, …

  données de l'entreprise

- Excellent outil Cain utilisé au labo → http://www.oxid.it/

Mots de passe : protection

- Protéger l'accès aux condensés (Windows_SAM, Linux_Shadow)
- Utiliser des mots de passe difficiles (8 caract. au min, maj-min, chiffres et caractères spéciaux)
- Changer de mot de passe régulièrement
- *Challenge – Response*
- *One Time Password*
- …

Authentifications locale & distante

- Authentification locale

  L'utilisateur dispose d'un accès physique à l'ordinateur

  Il utilise clavier et écran

- Authentification distante (réseau)

  L'utilisateur n'a aucun accès physique à l'ordinateur (distant)

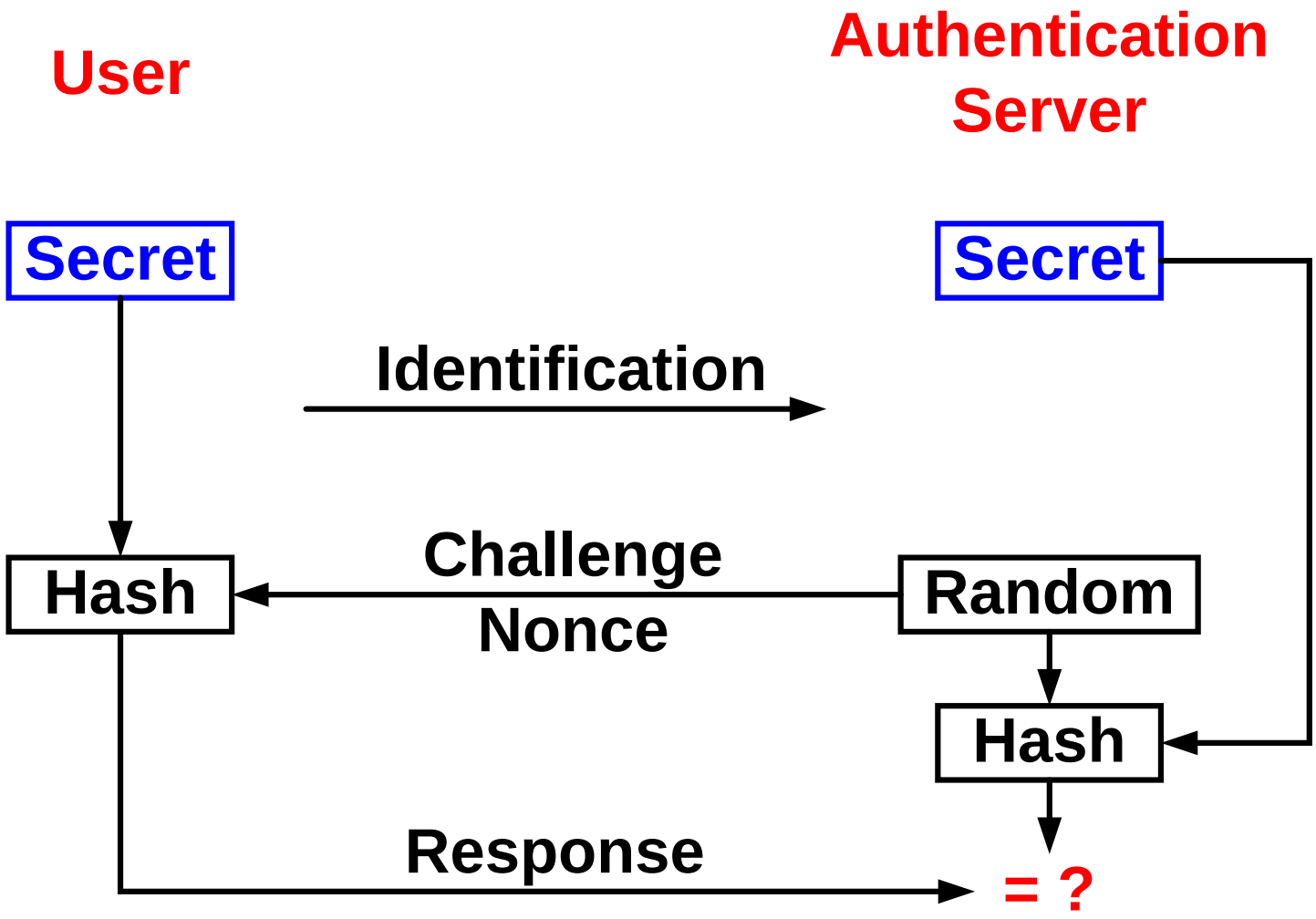
  Il doit disposer d'un ordinateur local à partir duquel il pourra établir une connexion à distance *(remote authentication)* en utilisant un protocole spécifique CHAP, Kerberos, …

- Dans le mode MS :

  la première catégorie fait partie du groupe *INTERACTIVE*

  la seconde catégorie fait partie du groupe *REMOTE INTERACTIVE*

Challenge – Response (1)



Challenge – Response (2)

- Fonctionnement

  Secret partagé

  Valeur aléatoire *(challenge, nonce)*
  Pseudo Random Number Generator (PRNG)

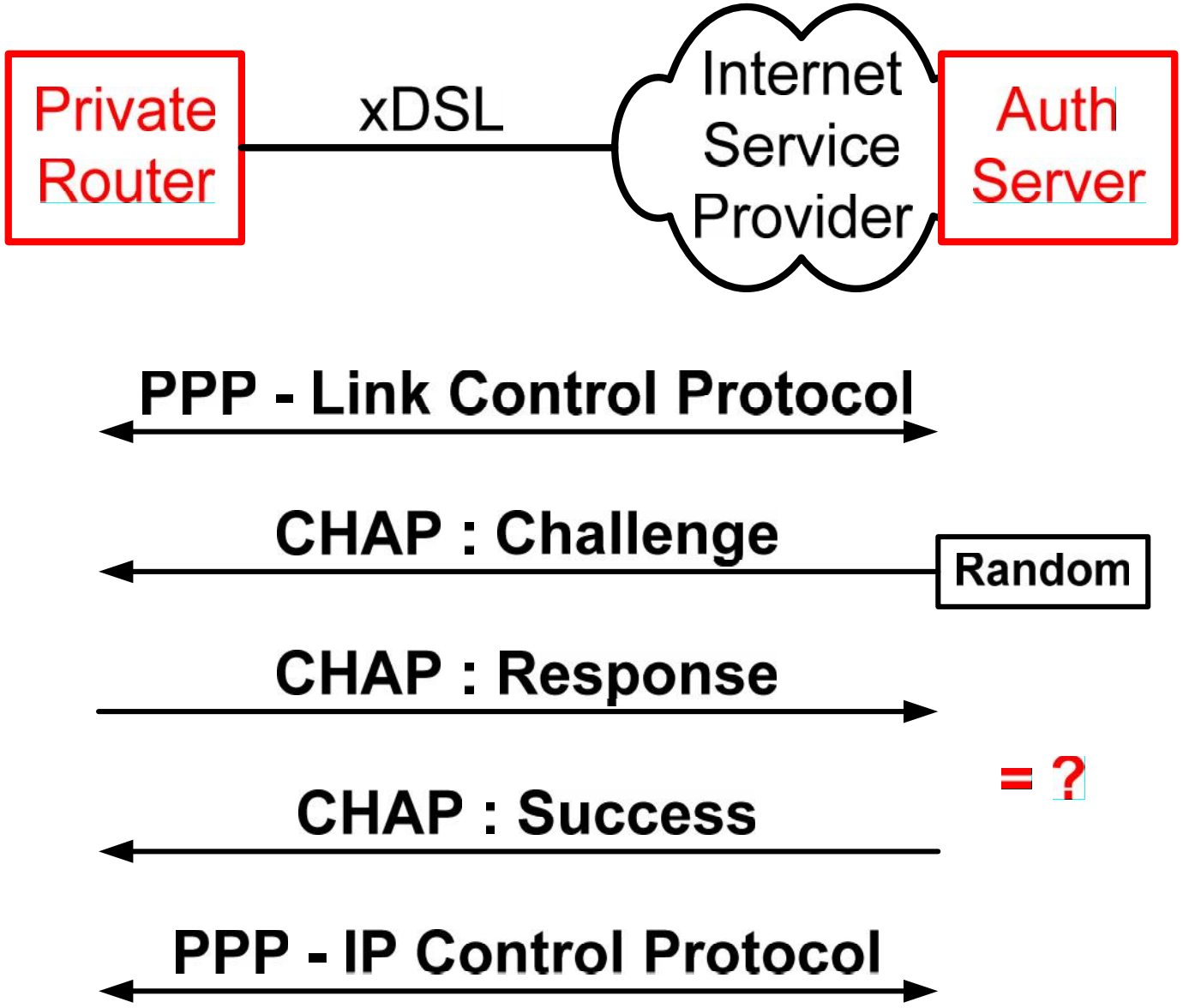
  Protocole CHAP *(Challenge Handshake Authentication Protocol)*
  rfc 1994 (1996)

- Avantages

  Le mot de passe n’est plus transmis sur le réseau

  L’attaque par rejeu *(replay attack)* ne fonctionne pas

ISP Authentication (xDSL) : PPP over Ethernet



Authentification http-digest

→ GET / HTTP/1

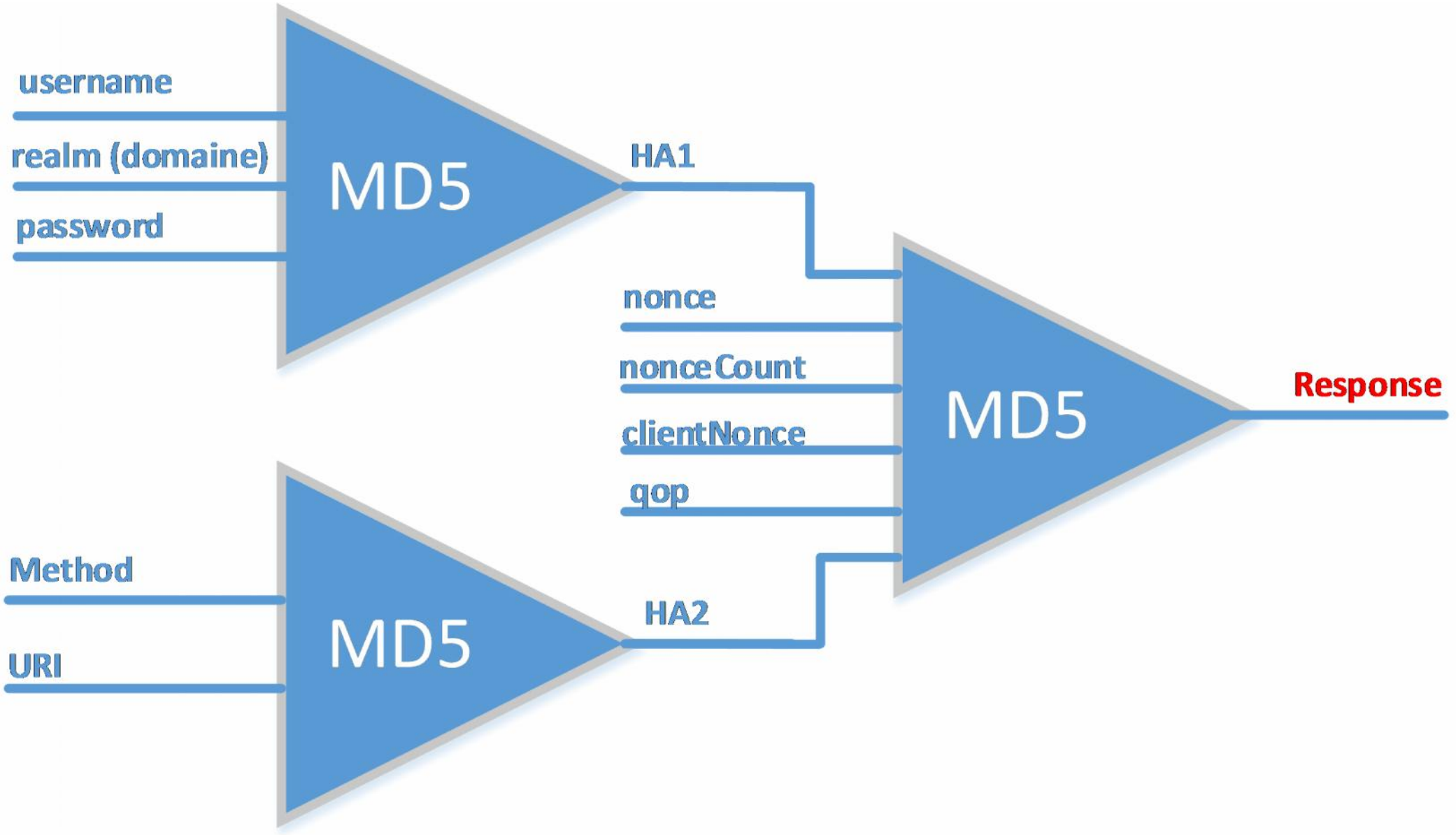
← HTTP/1.1 401 Authorization Required
realm="tdeig", **nonce="TIAw…"**, alg=MD5, qop="auth"

→ GET / HTTP/1
method=GET, URI=/, username="test", realm="tdeig", nonce=
cnonce="2c55…",nc=00000001, alg=MD5, **resp="1e39…"**

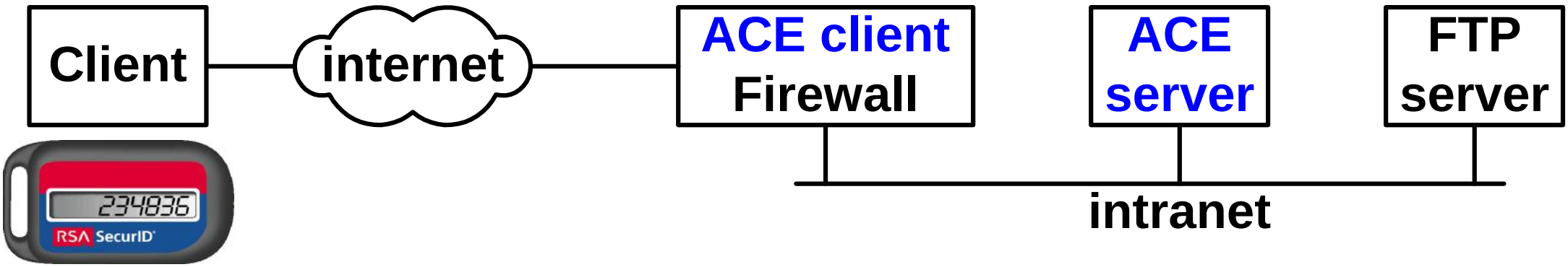
- Démo
- Acquisition Wireshark http-digest
- Voir http://tools.ietf.org/html/rfc2617
- Voir http://en.wikipedia.org/wiki/Digest_access_authentication
- Voir http://www.tdeig.ch/publication/tutorial_http.pdf §11

```php
<?php
$user     = "test";
$realm    = "tdeig";
$nonce    = "TIAwM6XXBAA=318594770a39693e94d6ce71b4d4187743f3816e";
$nc       = "00000001";  //nonce counter
$cnonce   = "2c55e50d8feba4a4880a85c57ab2b44f"; //client nonce
$qop      = "auth"; //quality-of-protection
$dico     = array("abc", "admin", "123", "passpass"); //mini dictionnaire

$i = 0;
while($i < count($dico)) {
    $password = $dico[$i];
    $HA1 = md5("$user:$realm:$password");
    $HA2 = md5("GET:/");

    $response = md5("$HA1:$nonce:$nc:$cnonce:$qop:$HA2");
    if($response == '1e391ae274b41581187747e23ce1c7c3') {
        print("Le mot de passe est : " . $password . "\n");
        break;
    }
    $i++;
}
?>
```

Authentification http-digest



Utilisation d'un jeton SecurID



TCP SYN (FTP) →

← FTP-220 Service Ready "Checkpoint"

USER toto →

← FTP-331 User name ok, need password

PASS 1234@234836 →

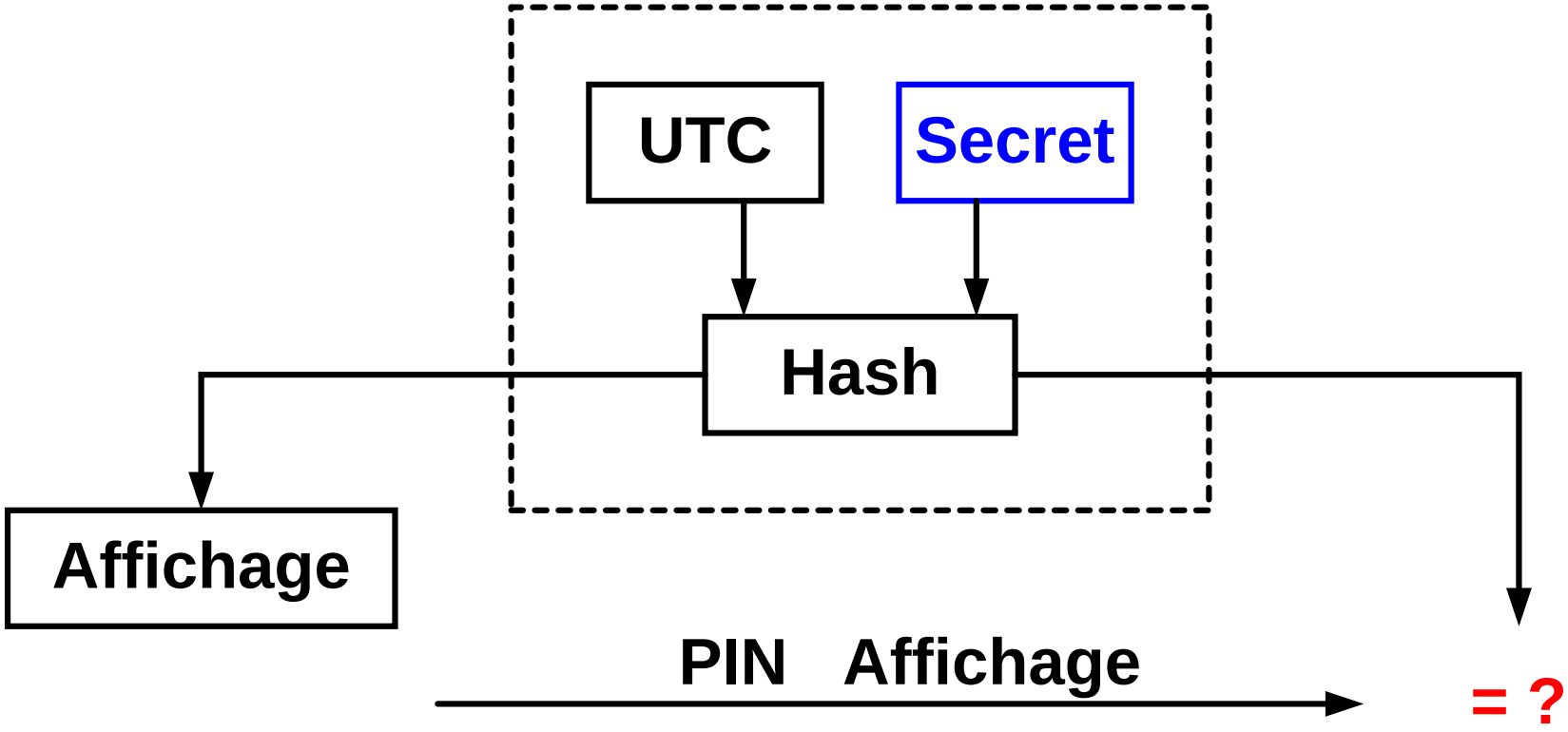
Authentification ↔

← FTP-230 User logged in

Token (SecurID)

User

Authentication Server



- **PIN code transmis (PIN code externe)**
- ***Universal Time Coordinated (Greenwich Mean Time)***

Token

- Fonctionnement

  Secret partagé

  Client & serveur doivent être synchronisés (UTC)

  Nouveau mot de passe généré chaque minute

  → ***One-Time Password***

  PIN code est transmis

- Avantage

  Non vulnérable à l'attaque par rejeu *(replay attack)*

- *Security by obscurity* : ce système a été initialement présenté avec un algorithme secret … qui a depuis été découvert *(reverse engineering)* http://www.linuxsecurity.com/resource_files/cryptography/initial_securid_analysis.pdf

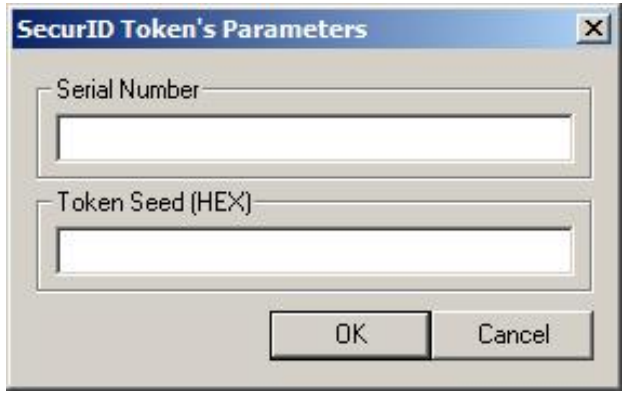
- Actualité → http://www.net-security.org/secworld.php?id=11122

Simuler un jeton SecurID avec Cain (démo)

- Le jeton SecurID est livré avec son fichier XML destiné au serveur

```
- <TKN>
    <SN>52438917</SN>
    <Seed>=1HFeoue3DcUE2LjQC9IzDA==</Seed>
    <Birth>2006/10/13</Birth>
    <Death>2010/02/28</Death>
    <TokenMAC>d2340PXigjqbGwQ0OAWH3A==</TokenMAC>
  </TKN>
```

- Entrer dans Cain

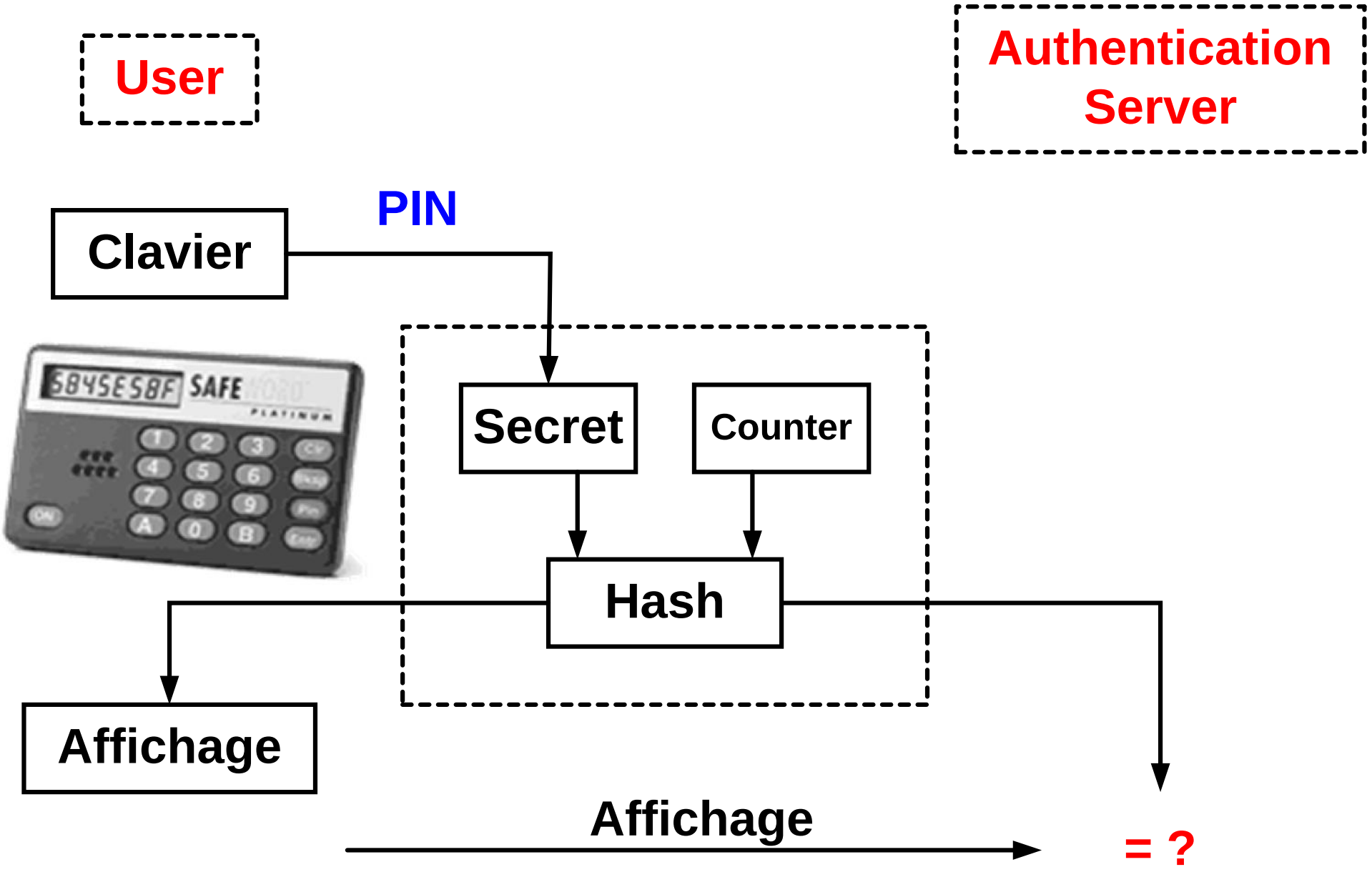


OTP

202466	2011/03/18 - 17:42
806952	2011/03/18 - 17:43
303797	2011/03/18 - 17:44

Serial Number	Key
52438917	3148466656f75653344517246424e6934

Token (Activ Card)



- **PIN code interne**