Sécurité Informatique

Chapitre 4 :

Intégrité des données

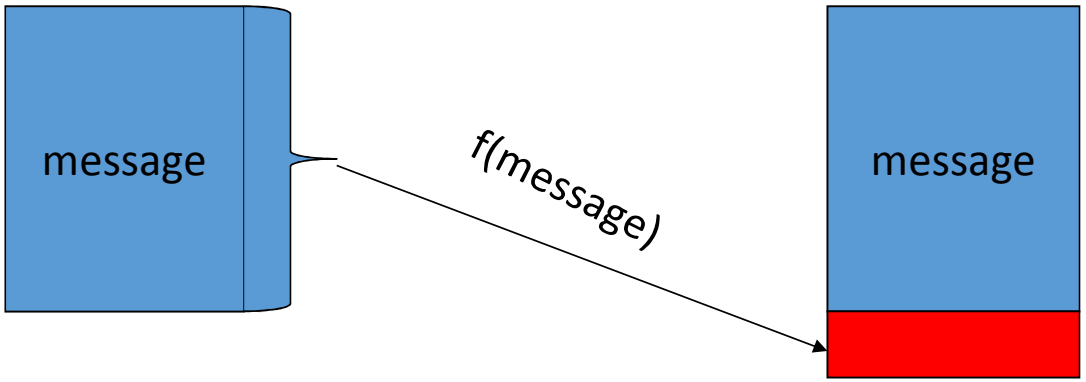
Guellil zouaoui

Propriétés de sécurité

- Confidentialité : S’assurer du caractère secret de l’information, une information ne peut être lue que par des entités habilitées
  - seul le chiffrement peut l’assurer.
- AUTHENTIFICATION :C'est la propriété qui assure la reconnaissance sûre de l ’identité d ’une entité.
- Intégrité : le message reçu est identique à celui émis,
  - la signature électronique est nécessaire.
- Non répudiation : C'est la propriété qui assure que l'auteur d'un acte ne peut ensuite dénier l'avoir effectué.
  - utilisation d’un tiers de confiance.

signature numérique : définition

- **la norme ISO 7498-2 :**« Données ajoutées à une unité de données, ou transformation cryptographique d'une unité de données, permettant à un destinataire de prouver la source et l'intégrité de l'unité de données et protégeant contre la contrefaçon par le destinataire par exemple »



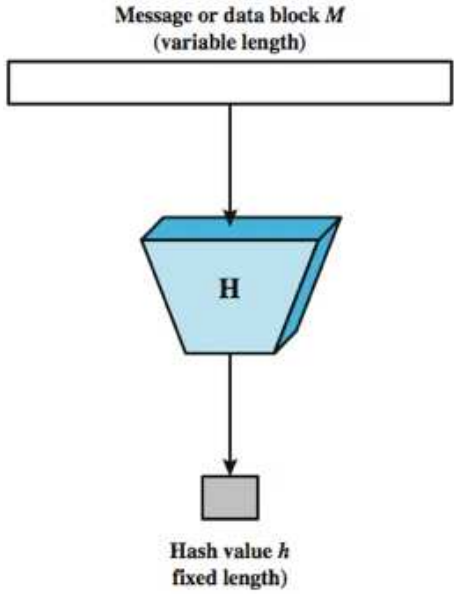
ps: f () est un algorithme de signature numérique.

signature numérique : définition

- La signature numérique permet de reproduire les caractéristiques d'une signature manuscrite (lier un document à son auteur, Rendre la signature difficilement imitable)
- La signature numérique repose sur deux familles d'algorithmes, qui seront utilisés de manière complémentaire :
  - Algorithmes de chiffrement dit « asymétriques » ou à « clef publique ».
  - Fonctions de hachages

Fonctions de hachages

- Les fonctions de hachages sont des fonctions à sens unique et « sans collision », générant une sortie de taille fixe (appelée condensat ou empreinte), caractéristique des données fournies en entrée.



Fonction de hachage : Propriétés

1. H peut être appliquée à un bloc de données de n'importe quelle taille.
2. H produit une sortie de longueur fixe.
3. pour toute donnée x, H(x) doit être facile à calculer.
4. Pour tout code donné h, il est impossible de trouver x tel que H(x) = h
5. Pour tout bloc donné x, il est impossible de trouver y différent de x tel que : H(y) = H(x).
6. il est impossible de trouver un couple (x, y) tel que H(x) = H(y)

Fonction de hachage Simple

	bit 1	bit 2	• • •	bit *n*
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
block *m*	b_{1m}	b_{2m}		b_{nm}
hash code	C_1	C_2		C_n

Figure 3.3 Simple Hash Function Using Bitwise XOR

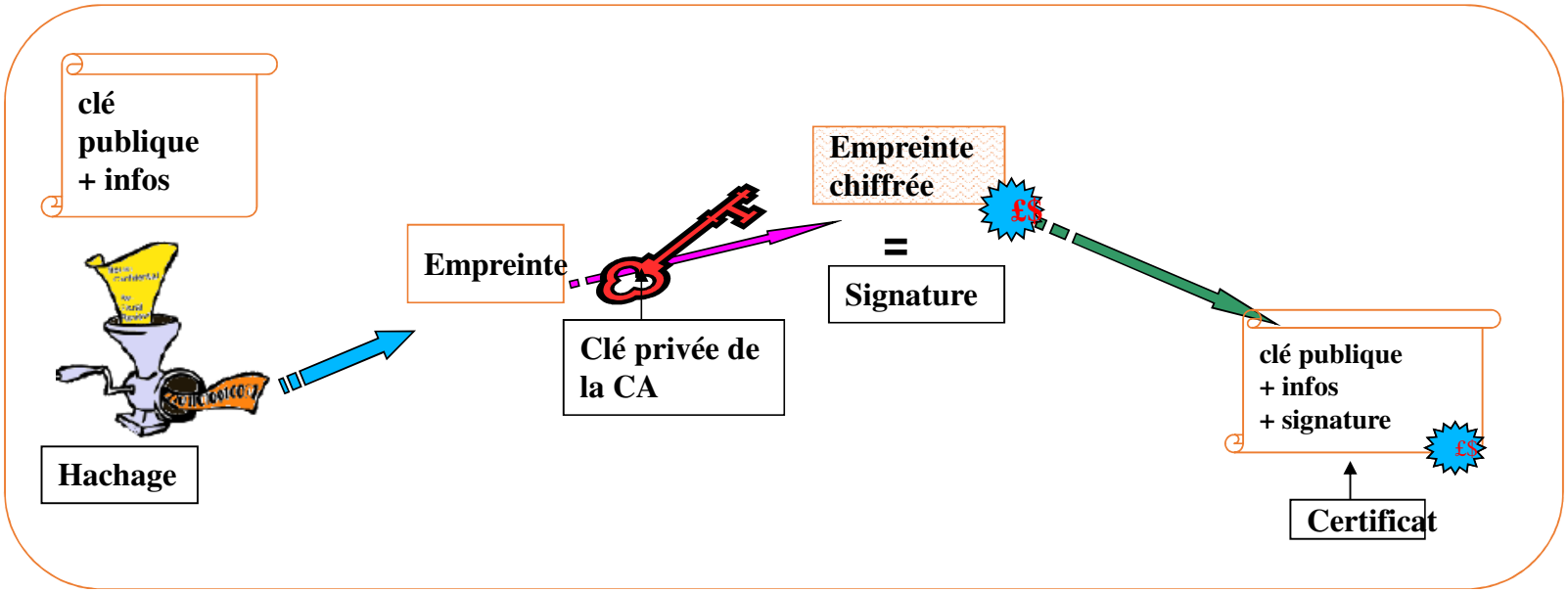
Block1 Xor Block2 Xor Block3 Xor..... Blockn = hash code

Exemple

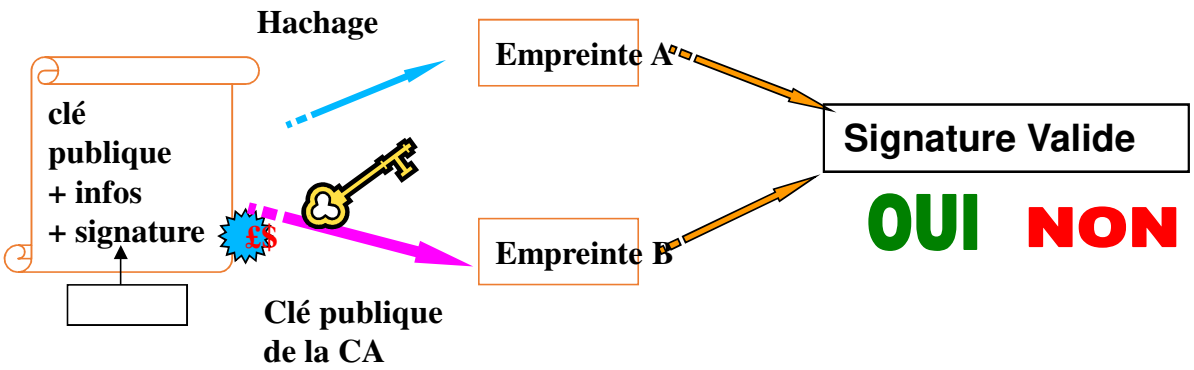
- MD5 (Message Digest 5) : Empreinte de 128 bits
- SHA (Secure Hash Algorithm) : Norme NIST, Empreinte de 160 bits
- SHA-1 (révision publiée en 1994) Considéré comme plus sûr que MD5
- SHA-2 (octobre 2000) Agrandit la taille de l'empreinte.
- SHA-3: Norme NIST octobre 2012. basé sur un principe tout à fait différent de celui des fonctions MD5, SHA-1 et SHA-2 suite des possibilités d'attaques contre ces dernier

Signature d’un document

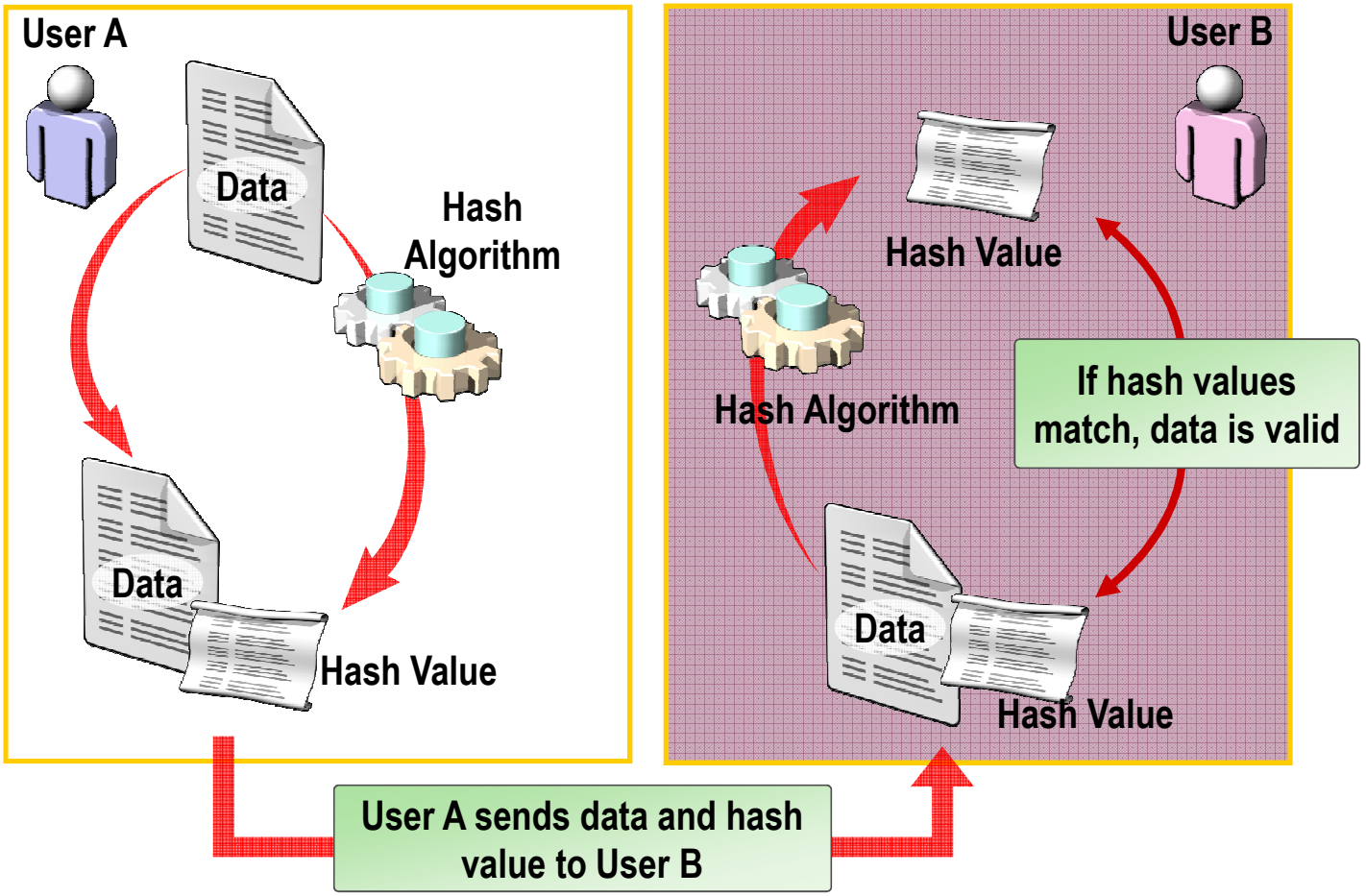
- La Signature est calculée par un algorithme cryptographique et liée aux données de telle façon que les destinataires puissent vérifier que les données n'ont pas été altérées et qu'elles proviennent effectivement de l'expéditeur du message.
- un algorithme de hachage (MD5, SHA..) calcule une empreinte (ou condensat) qui résume le message, cette dernière va être crypté en utilisant un algorithme asymétrique (RSA, DSA..) et la clé privé du propriétaire se qui va donné comme résultat la signature du message.



- Lorsque un récepteur reçoit message,
- il déchiffre sa signature en utilisant la clé publique du propriétaire afin d'obtenir son empreinte (empreinte de référence)
- en même temps il calcule l'empreinte (empreinte candidate),
- s'ils sont les même alors le message est valide.
- Ceci est illustré dans la figure suivante :


Hachage
clé
publique
+ infos
+ signature
£$
Clé publique
de la CA
Empreinte A
Empreinte B
Signature Valide
OUI
NON

Verifying Simple Data Integrity with Hashes (Message Digests)



Message Authentication Code: MAC

- Le concept est relativement semblable aux fonctions de hachage. Il s'agit ici aussi d'algorithmes qui créent un petit bloc authentificateur de taille fixe.
- La grande différence est que ce bloc authentificateur ne se base plus uniquement sur le message, mais également sur une clé secrète.
- Utilisation du chiffrement Conventionnel
  - seuls l'émetteur et le récepteur devraient partager une clé
- Authentification sans chiffrement de message
  - Un champ d'authentication est généré et ajouté à chaque message
- Code d'Authentification de Message (MAC)
  - Calculer le MAC comme une fonction du message M et de la clé K. MAC = F(K, M)

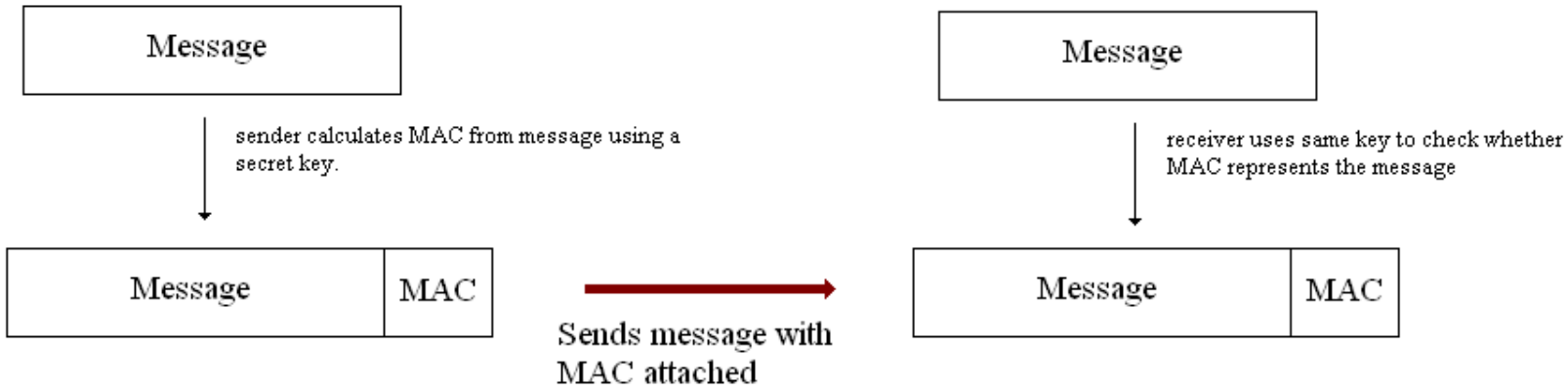
Message Authentication Codes

- Condenses message into a short hash

- For example, encrypt only the MAC with a key known to sender and receiver.

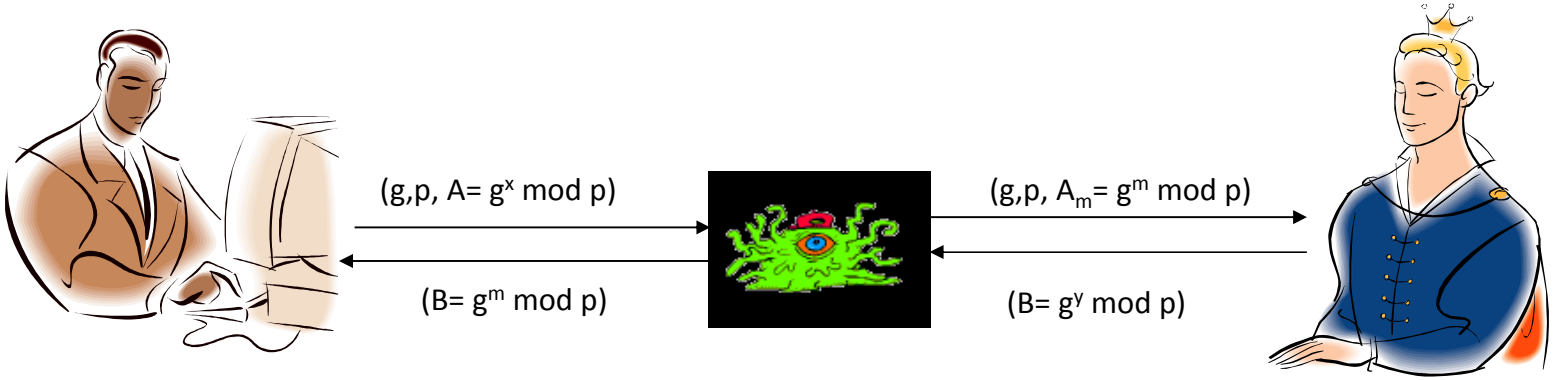
Message Authentication Code

- Alternatively, use a secret key.
- This also provides authentication.



Diffie Hellman: rappel

- Man in the middle attack



Bob

La clés de communication avec Alice est
K= g^{mx} mod p

Man in the Middle

K_{Bob}= g^{mx} mod p

K_{Bob}= g^{my} mod p

Alice

La clés de communication avec Bob est
K= g^{my} mod p

- Les interlocuteurs ne sont pas authentifiés : une attaque par le milieu est toujours possible
- Les échanges ne sont pas intègres : un attaquant peut modifier le secret commun
- Nouveaux schémas d’échange de clé authentifié
- une infrastructure permettant d’authentifier les clés publiques pour échanger des clés de session

Bob

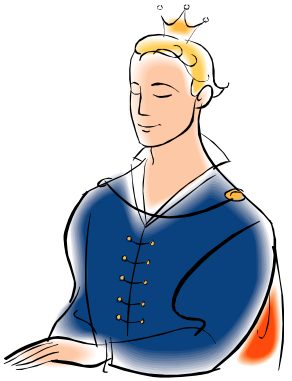
La clé privé :K_{Bs}

Clé publique :K_{Bp}

Bob,g,p, B= g^x mod p

Alice, A= g^y mod p,$sign_{kAs}(g^x,g^y)$

Bob,$sign_{kBs}(g^y, g^x)$



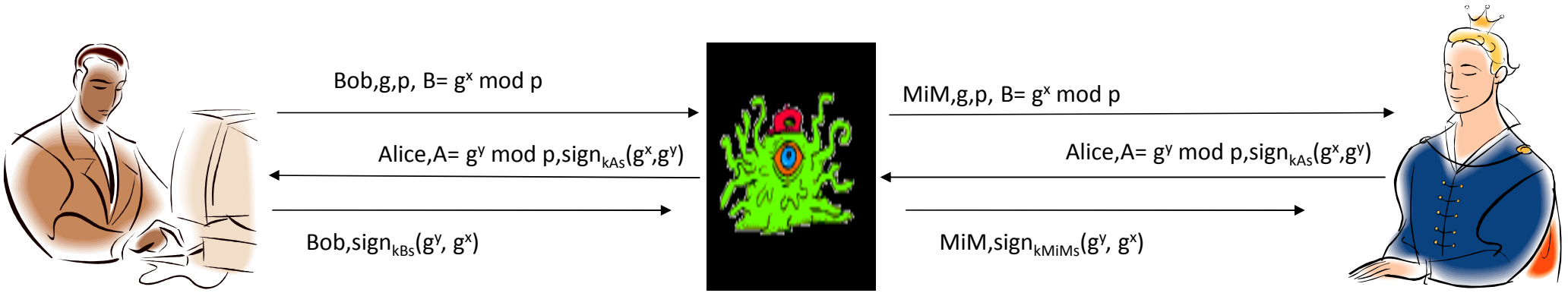
Alice

La clé privé :K_{As}

Clé publique :K_{Ap}

- évite les attaques par le milieu : impossible pour un attaquant de fournir $SIG_{KBs}(g^x, g^y)$
- évite l' attaque par rejeu: la valeur DH du correspondant est aussi signée

Usurpation d'identité



Bob

La clé privé :K_{Bs}

Clé publique :K_{Bp}

Man in the Middle (MiM)

La clé privé :K_{MiMs}

Clé publique :K_{MiMp}

Alice

La clé privé :K_{As}

Clé publique :K_{Ap}

- Man in the middle attack

Schéma SIGMA (SIGn and MAc) proposé par Krawczyk à Crypto 2002

- Schéma SIGMA (SIGn and MAc) proposé par Krawczyk à Crypto 2002
- La clé de MAC K_m et la clé de session partagée K_s sont dérivées de g^{xy}
- Elles doivent être indépendantes calculatoirement

Bob

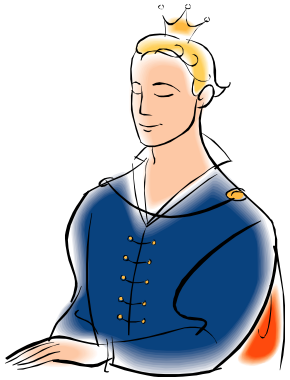
La clé privé :K_{Bs}

Clé publique :K_{Bp}

Bob,g,p, B= g^x mod p

Alice, A= g^y mod p,$sign_{kAs}(g^x,g^y)$,MAC_{Km}(Alice)

Bob,$sign_{kBs}(g^y, g^x)$, MAC_{Km}(Bob)



Alice

La clé privé :K_{As}

Clé publique :K_{Ap}

SIGMA avec protection des identités

- Le chiffrement des deux derniers échanges permet d'assurer la protection des identités,
- Le chiffrement E doit être sûr contre les attaques actives,