Cryptographie dans les bases de données

Benjamin NGUYEN

5A STI – A2S

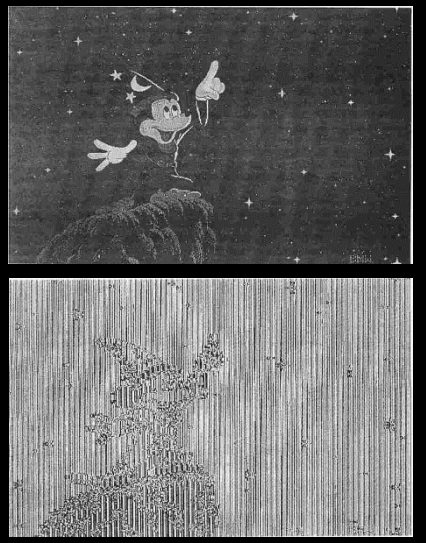
Slides from: N. Anciaux, L. Bouganim, P. Pucheral, A. Canteaut

Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L’approche serveur
- L’approche client
- L’approche matérielle (TPM - TCB)
- Conclusion

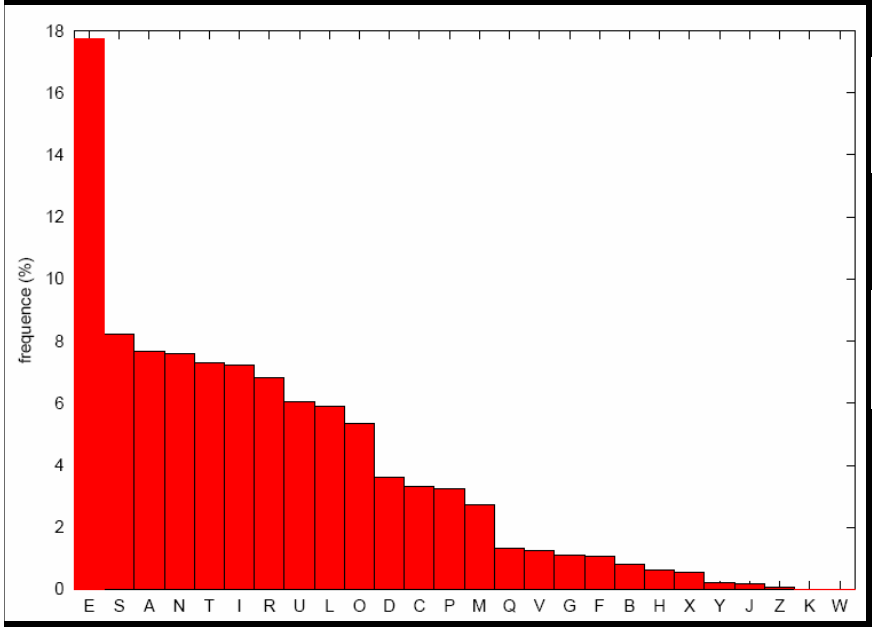
Application du chiffrement au contexte BD

- Si je chiffre une BD avec un algorithme sûr, le résultat est-il sûr ?
  - Non !
  - La robustesse aux attaques des (meilleurs) algorithmes de chiffrement dépend de leur mise en œuvre (mode opératoire/protocole)
  - Exemple : Mode ECB (chiffrement par blocs)
    - ⇒ attaque par analyse de fréquence



- Le contexte BD a des spécificités difficiles à prendre en compte
  - Gros volume de données, performance des requêtes
  - Motifs répétés, distribution qui peuvent être connues
  - Données modifiables
  - Durée de stockage quasi-illimitée
  - Granularité de chiffrement ?

Eviter le chiffrement déterministe
(1) Analyse de fréquence



Fréquence des lettres en français

Dans le texte chiffré

B	N	C	U	X	Q	G	I	W	V
18,7	9,91	7,78	6,90	6,72	6,37	5,84	5,84	5,30	4,60

En français

E	S	A	N	T	I	R	U	L	O
17,8	8,23	7,68	7,61	7,30	7,23	6,81	6,05	5,89	5,34

svxlegi avxw s atxsew ues dvttes r ehxqaaye
aweggegi res aueaiwvs lasies vqseaxz res tews
hxq sxqlegi qgrvuegis mvtaaygvgs re lvsaye
ue galqwe yuqssagi sxw ues yvxoowes atews
a aeqge ues vgi qus reavses sxw ues auagmdes
hxe mes wvqs re u akxw tauarwvqis ei dvgiexz
uaqssegi aqiexsetegi uexws ywagres aques euagmdes
mvtte res alqwvgs iwaqgew a mvie r exz
me lvsayexw aque mvtte qu esi yaxmde ei lexue
uxq gayxewe sq eeax hx qu esi mvtqhxe ei uaqr
u xg ayame svg eem alem xg ewxueyxexue
u axiwe tqte eg evqiagi u qgoqwte hxq lvuaqi
ue aveie esi seteuaeue ax awqgme res gxees
hxq dagie ua ietaeie ei se wqi re u awmdew
ezque sxw ue svu ax tquqex res dxees
ses aques re yeagi u etaemdegi re tawmdew

Conclusion

B ⟶ E
N ⟶ S
C ⟶ A

(2) Analyse de fréquence: bigrams

Dans le texte chiffré

ES	UE	GI	RE	EG	EX	IE	SE	QU	TE	UA	EW	AG	AQ	HX	XW
25	17	13	12	9	8	8	8	8	8	8	7	7	7	7	7

En français

ES	LE	EN	DE	RE	NT	ON	ER	TE	SE	ET	EL	QU	AN	NE	OU	AI

U ⟶ L
R ⟶ D
G ⟶ N
Q ⟶ I

Dans le texte chiffré

ES	LE	NI	DE	EN	EX	IE	SE	IL	TE	LA	EW	AN	AI	HX	XW
25	17	13	12	9	8	8	8	8	8	8	7	7	7	7	7

En français

ES	LE	EN	DE	RE	NT	ON	ER	TE	SE	ET	EL	QU	AN	NE	OU	AI

I ⟶ T

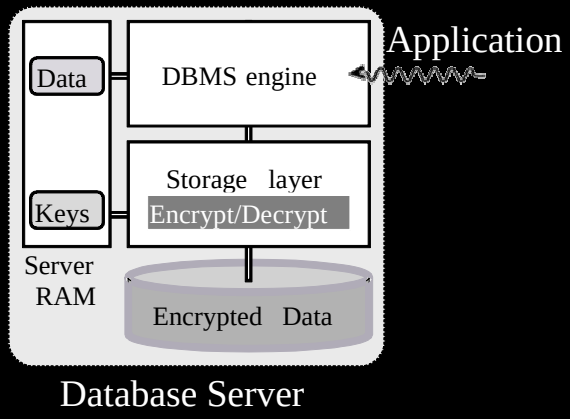
(3) En complétant par des noms communs

svxlent avxw s atxsew les dvttes d ehxiaaye
awennent des aleatwvs lastes viseaxz des tews
hxi sxilent indvlents mvtaaynvns de lvsaye
le naliwe ylissant sxw les yvxoowes atews
a aeine les vnt ils deavses sxw les alanmdes
hxe mes wvis de l akxw taladwvits et dvntexz
laissent aitexsetent lexws ywandes ailes elanmdes
mvtte des aliwvns twainew a mvte d exz
me lvsayexw aile mvtte il est yaxmde et lexle
lxi nayxewe si eeax hx il est mvtihxe et laid
l xn ayame svn eem alem xn ewxleyxexle
l axtwe tite en evitant l inoiwte hxi lvlait
le avete est setelaele ax awinme des nxees
hxi dante la tetaete et se wit de l awmdew
ezile sxw le svl ax tiliex des dxees
ses ailes de yeant l etaemdent de tawmdew

indvlent vnt	V ⟶ O
oiseaxz	X ⟶ U Z ⟶ X
a aeine	A ⟶ P
leuws	W ⟶ R
taladroits	T ⟶ M
yrandes	Y ⟶ G

Chiffrer à quel niveau ? (1/3)

- Chiffrement niveau OS (chiffrement fichiers/stockage)

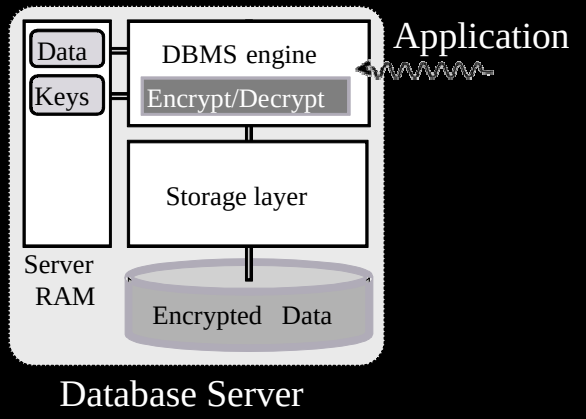


- ✚ Transparent pour le SGBD et l'application
- ━ … mais chiffrement non sélectif ➔
  - Granularité = fichier
  - Chiffrement partiel proscrit (=> performances ?)
- ━ … et ne résiste pas aux attaques du DBA

Hors Sujet pour ce cours !

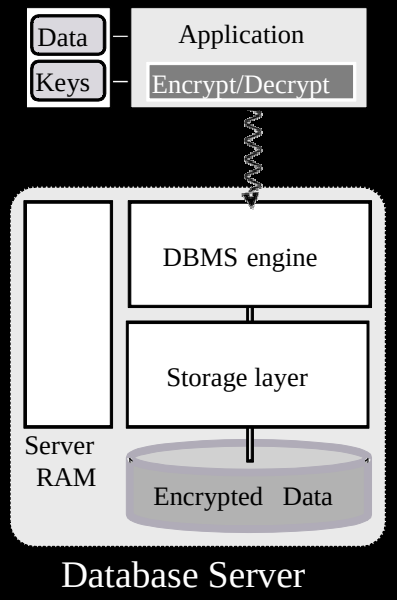
Chiffrer à quel niveau ? (2/3)

- Chiffrement niveau SGBD/serveur
  - ✚ Chiffrement sélectif (spécifique)
    - Chiffrer selon les privilèges utilisateur
    - Chiffrer les données les plus sensibles
      - au niveau table, ligne, colonne…
      - … de façon conditionnelle (salaire >10K)
  - ✚ Transparence pour l'application
  - ━ … mais mécanismes internes SGBD à revisiter
    - Evaluation de requête + indexation sur des données chiffrées impossible
      - sauf chiffrement à propriétés particulières (préservant égalité ou l'ordre => dangereux pour la sécurité)
    - Surtout dans un contexte où le serveur n'est pas de confiance (approche client)
  - ━ … et problème de performance en perspective
  - ━ … et ne résiste toujours pas aux attaques du DBA



Chiffrer à quel niveau ? (3/3)

- Chiffrement niveau application/client
  - + Résistance aux attaques internes - DBA
  - – Aucune transparence pour l'application
    - L'application pilote de chiffrement/déchiffrement
      - Et gère les clés…
    - Le SGBD ne peut traiter les données déchiffrées
      - Au risque d'attaques internes
  - – Dégradation importante des performances
    - L'application prend à sa charge une (grande) partie de l'évaluation de requête
  - – Le client peut attaquer les droits d'accès
    - Les données et les clés sont en clair sur le client

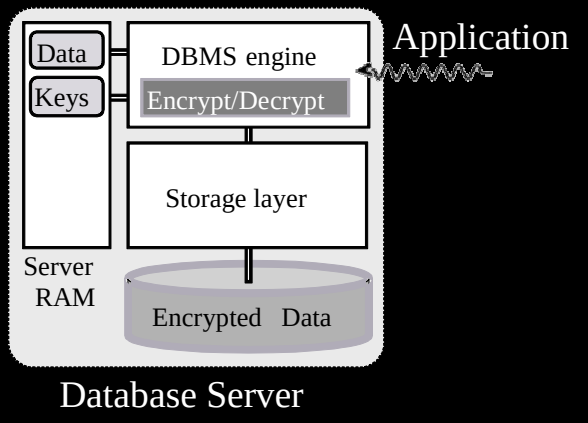


Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L’approche serveur
- L’approche client
- L’approche matérielle (TPM - TCB)
- Conclusion

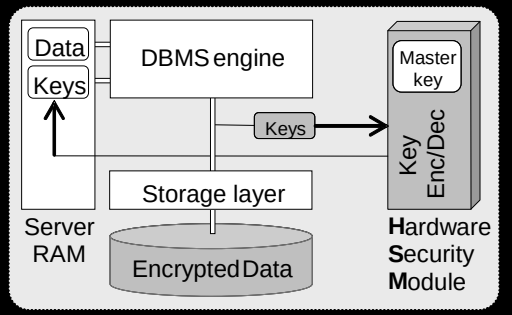
L'approche chiffrement serveur : principe

- Le serveur chiffre/déchiffre/hache les données
  - Les clés sont détenues par le serveur
  - Les données sont déchiffrées lors de l'évaluation des requêtes
- La BD est protégée sur le support de stockage persistant
  - Le niveau de protection des données n'excède pas celui des clés, stockées sur le serveur
  - Solution Oracle: les clés sont chiffrées par une *master key* chiffrée elle-même avec un mot de passe administrateur (*wallet*)
- Faiblesses : attaques internes
  - Attaque administrateur (le DBA a tous les droits)
    - Peut accéder aux clés/données et utiliser la base
    - Ceci sans laisser de traces

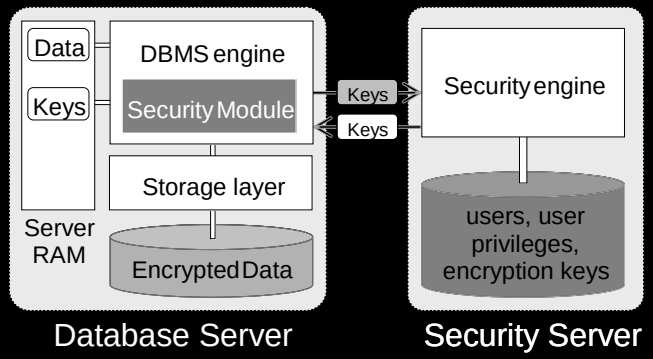


L'approche serveur : gestion des clés

- Techniques réduisant au maximum
  - l'exposition des clés
  - Et les droits de l'administrateur
- Introduction d'un dispositif HW sécurisé (HSM, Hardware Security Module)
  - ✚ Les clés ne sont accessibles que pendant l'exécution
  - Elles sont chiffrés avec une « master key », dans le HSM
  - ━ … mais les clés restent exposées (brièvement)
  - … et les données sont en clair lors de l'exécution



- Introduction d'un serveur de sécurité (serveur distinct)
  - Un 2ème serveur gère les clés et droits d'accès
    - 2ème Admin : Grant/revoke/create user…
  - ✚ Résiste « mieux » aux attaques internes
    - Le DBA/trojan ne peut pas observer l'empreinte de la BD
  - ━ Mais les clés/données restent en clair à l'exécution



Approche serveur: solutions commerciales (1)

- L'exemple d'Oracle (attention indisponible sur la version XE)
  - DBMS Obfuscation Toolkit (8i / 1999) :
    - Solution de chiffrement niveau application
    - À base de procédures stockées (chiffrement/déchiffrement/hachage)
  - Transparent Data Encryption (TDE) :
    - Solution de chiffrement niveau SGBD
    - SQL étendu à la gestion du chiffrement
    - Master key : chiffrée par un mot de passe (admim) ou stockée dans un fichier ou un HSM
    - Chiffrement niveau attribut (10g / 2003)
      - Colonnes sensibles chiffrées : dans les tablespaces (même temporaires), SGA (mémoire côté serveur), logs/backups…
      - Indexation des prédicats d'égalité (chiffrement NO_SALT) ➔ attaque par analyse de fréquence !
    - Chiffrement niveau Tablespace (11g / 2007)
      - Tablespaces complets chiffrés sur disque, déchiffré en SGA
      - Indexation classique (indexes fabriqués sur le clair)
      - Simplifie la gestion du chiffrement (on chiffre toute la table)
- SQL server 2008 TDE : similaire à Oracle TDE / chiffrement niveau Tablespace

Approche serveur: solutions commerciales (2)

- Protegrity Database Protector
  - Basé sur une solution de chiffrement serveur
    - Pour tout SGBD : Oracle, SQL Server, IBM DB2 …
  - Ajout d'un serveur de sécurité
    - Secure.Manager : gestion des utilisateurs, droits et clés
    - Secure.Server : module de chiffrement intégré au noyau SGBD
  - … et 2 personnes physiques différentes …
    - Database Administrator (DBA) / Security Administrator (SA)
- IBM DB2 Data Encryption Expert : similaire à Protegrity

L’approche serveur : conclusion

- L’empreinte de la BD est chiffrée
  - Attention aux concessions sur la sécurité pour les performances (indexation)
- Exposition des clés de chiffrement
  - HSM => exposition brève lors de l’exécution
- Réduction des droits de l’administrateur
  - Serveur de sécurité => le DBA n’a plus « tous » les droits
- Mais :
  - Données en clair lors de l’exécution
  - Clés détenues ou transmises au serveur
- Limite de l’approche serveur
  - Le serveur n’est pas un site de confiance
  - Vulnérable aux attaques internes (administrateur / insider / trojan)
- A utiliser : quand vous avez confiance dans le SGBD et que vous voulez vous protéger contre la perte / vol / piratage d’un disque.

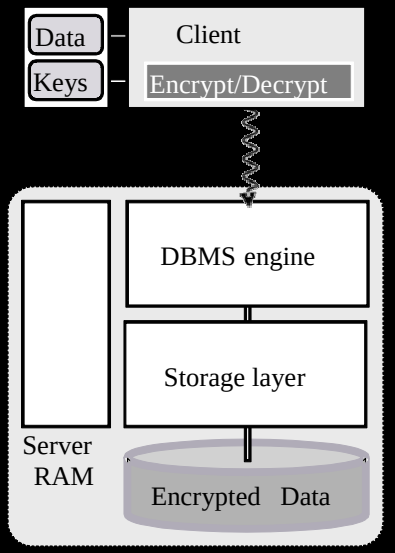
Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L’approche serveur
- L’approche client
- L’approche matérielle (TPM - TCB)
- Conclusion

L'approche chiffrement client : principe

- Chiffrement coté client
  - Pas de transmission du clair ni des clés au serveur
  - Le traitement s'effectue sur le client (pire cas)
- La BD est protégée coté serveur
  - Le serveur résiste aux attaque internes
  - Mais… dégradation très importante des performances

  ➔ Problème : déporter la majeure partie du traitement sur le serveur (données chiffrées), sans perte de sécurité
- Limites de l'approche client :
  - Gestionnaire de droit côté client
  - Données et clés en clair sur le client
  - Or le client n'est pas forcément un site de confiance
  - Donc ne convient pas à une BD partagée (BD privée uniquement)
- A utiliser : quand vous avez plus confiance en votre machine que dans le SGBD (rare ?)



Réponses au problème de performance

- Indexation des données
  - Indexation (index traditionnel chiffré)
    - Ex. Le client maintient et utilise (traverse) un B+-Tree [DDJ+03]
  - Etiquetage des tuples
    - Ex. Le client pose des étiquettes, pour sélectionner/joindre [HIL+02]
- Modèle de chiffrement
  - … indexer des données chiffrées ne sert à rien ☺
  - … traiter directement les données chiffrées est impossible (ou vulnérable aux attaques statistiques) ☺
  - SAUF Si: on dispose de chiffrement à propriétés particulières
    - Ex. Préservant l'égalité [Ora07, BoP02], l'ordre [AKS+04], ou homomorphiques [GeZ07]
    - Compromis sécurité / performance…

Homomorphic Encryption Example

- Homomorphic Encryption is a characteristic of several crypto-systems such as RSA, Paillier, ElGamal, etc.
- *Example :* Consider RSA. Given the RSA public key (e, m), the encryption of a message x is given by :

$$E(p)=p^{e} \bmod m$$

The homomorphic property is :

- $E(p_1) \times E(p_2) = p_1^{e} \times p_2^{e} \bmod m = (p_1 \times p_2)^{e} \bmod m = E(p_1 \times p_2)$

- *Fully* Homomorphic Encrytion means that *all ring* operators are homomorphic (this means + and x).

Fully Homomorphic Encryption [Gen09]

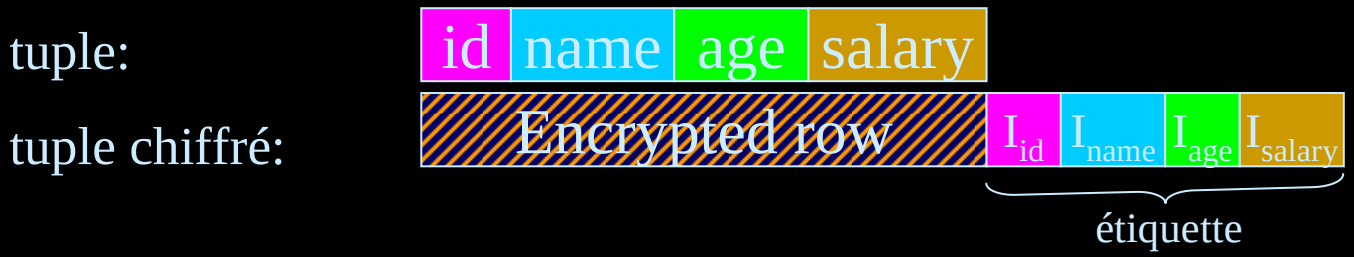
- Why is this a solution ?
  - Any program with bounded input can be transformed into a Boolean circuit
  - Any circuit can be transformed into a polynomial modulo 2
  - Secure computation of a polynomial equates to securely computing any program
  - To securely compute a polynomial, it is necessary and sufficient to securely compute + and x operations.

– We say that E is a fully homomorphic encryption from ({0,1}, +, x) to
– (D, $\oplus$, $\otimes$) if for all c_1, c_2 in D, such that $c_1=E(p_1)$ and $c_2=E(p_2)$
  – $E^{-1}(c_1) \oplus E^{-1}(c_2) = p_1+p_2$
  – $E^{-1}(c_1) \otimes E^{-1}(c_2) = p_1 \times p_2$
– Or more generally $E^{-1}(f_D(c_1,\ldots,c_n))=f_{\{0,1\}}(p_1,\ldots,p_n)$

- A first result was proposed using ideal lattice cryptography in [Gent09], and has been a hot topic since.
- The cost to have good security is (incredibly) high.

Solution par étiquetage [HIL+02]

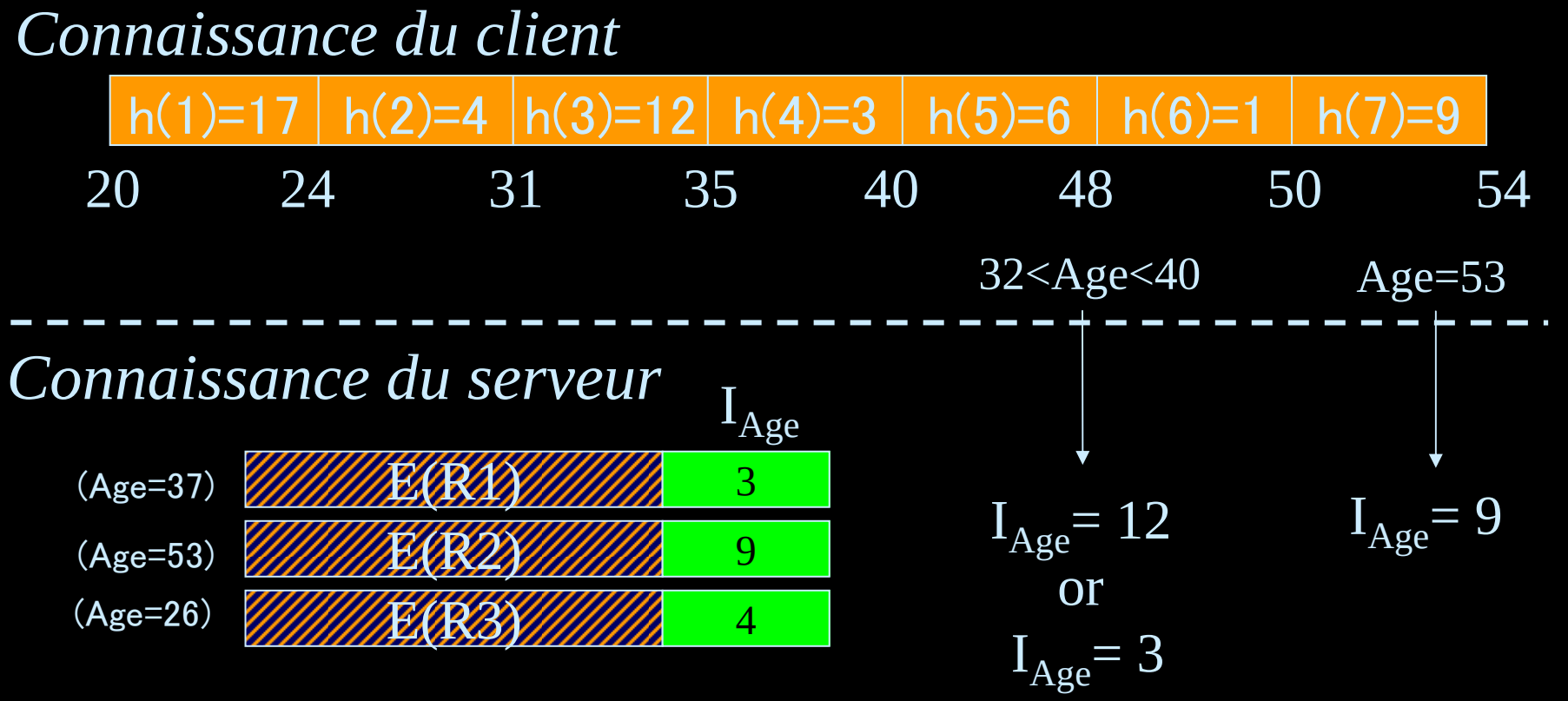
- Granule de chiffrement = tuple
- Ajout d'étiquettes d'attributs
  - Indique l'appartenance d'un attribut d'un tuple à une plage de valeurs
  - Permet des traitements approximatifs sur le serveur
    - Sélection, jointure, groupement



Attributs numériques

- Partitionner le domaine de variation d'un attribut
  - Chaque partition contient un nombre identique de tuples

  → Empêche une attaque en fréquence



Attributs « String »

- Signatures de string (n-grams)

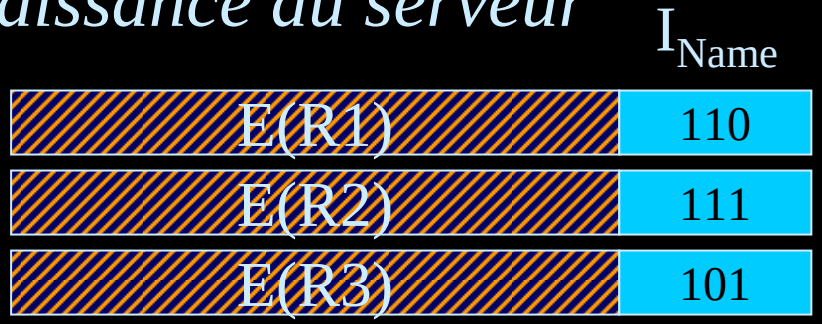
Connaissance du client

N={"g", "re", "ma"}

string	signature
'Greencar'	**110**
'Bigrecordman'	**111**
'Bigman'	**101**

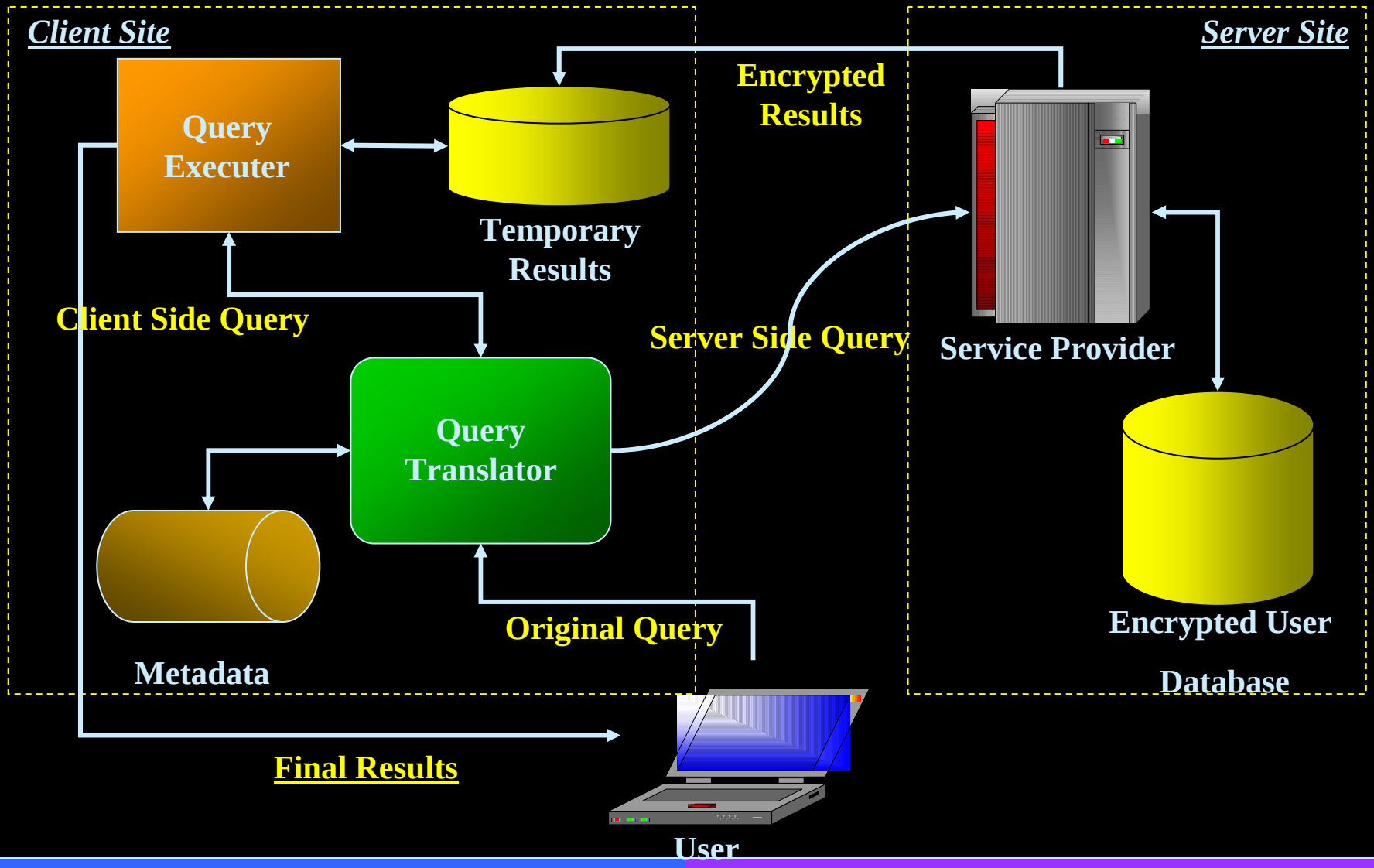
name LIKE '%green%'

Connaissance du serveur



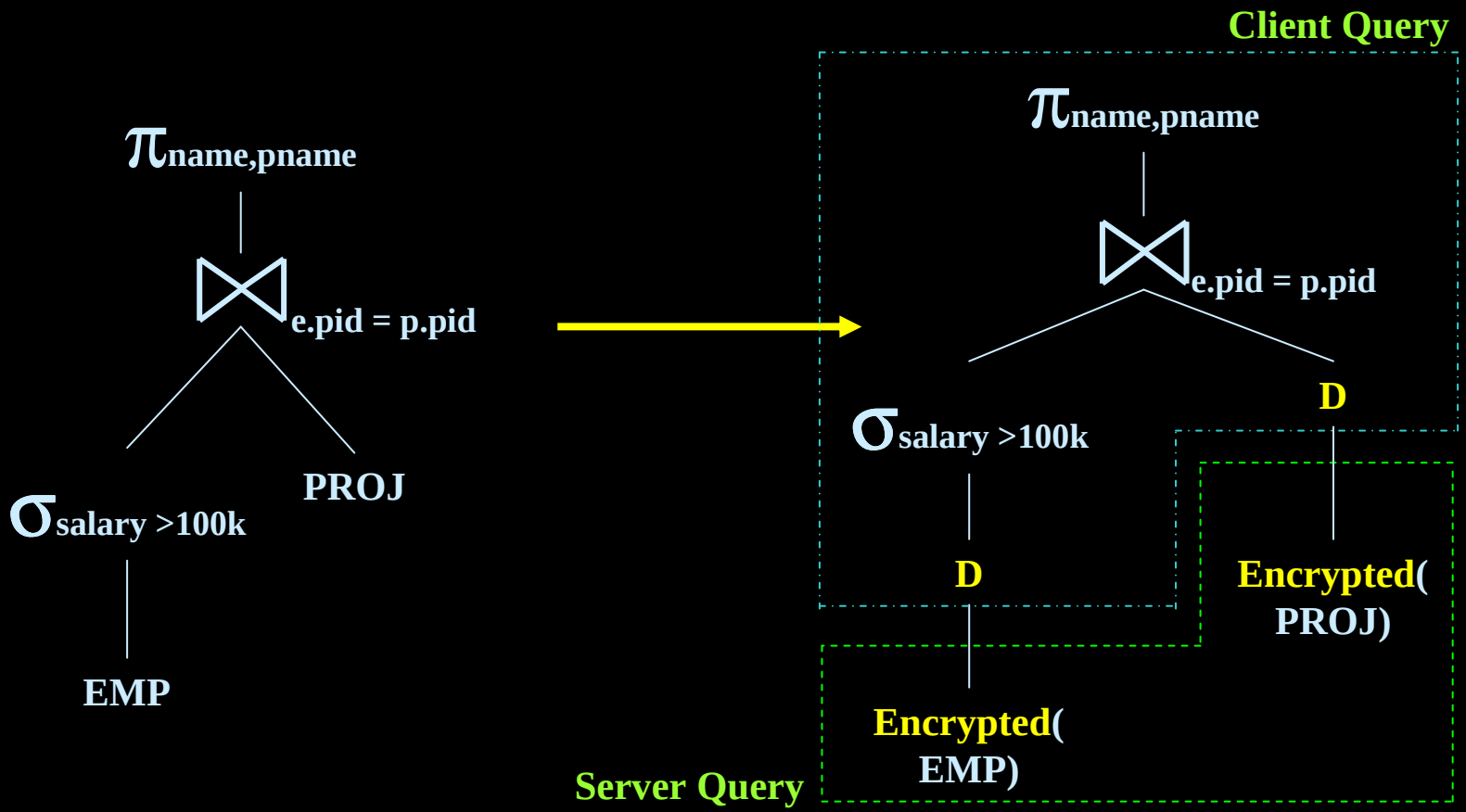
I_{Name} in (110, 111)

Architecture

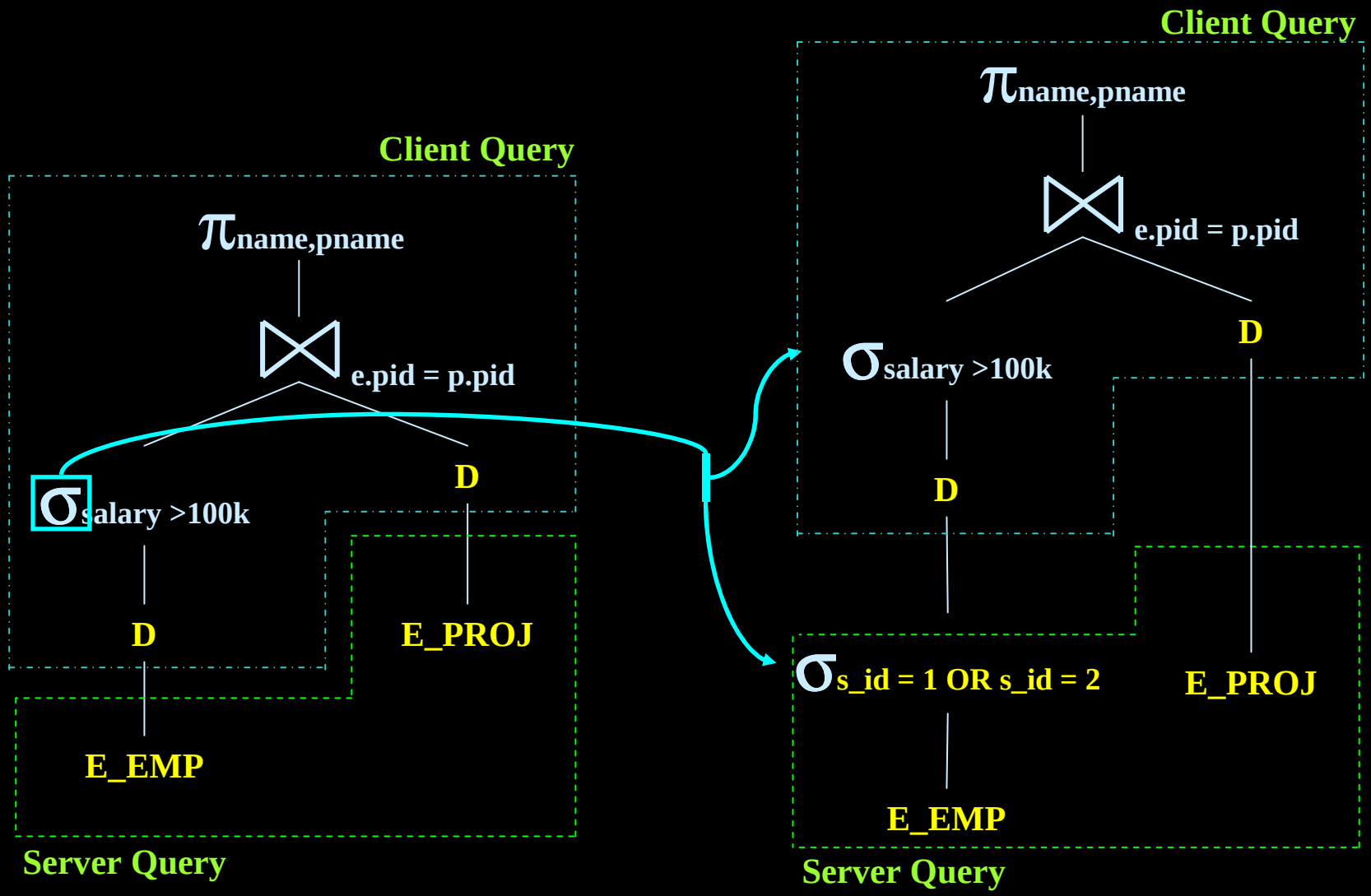


Décomposition de requête (1)

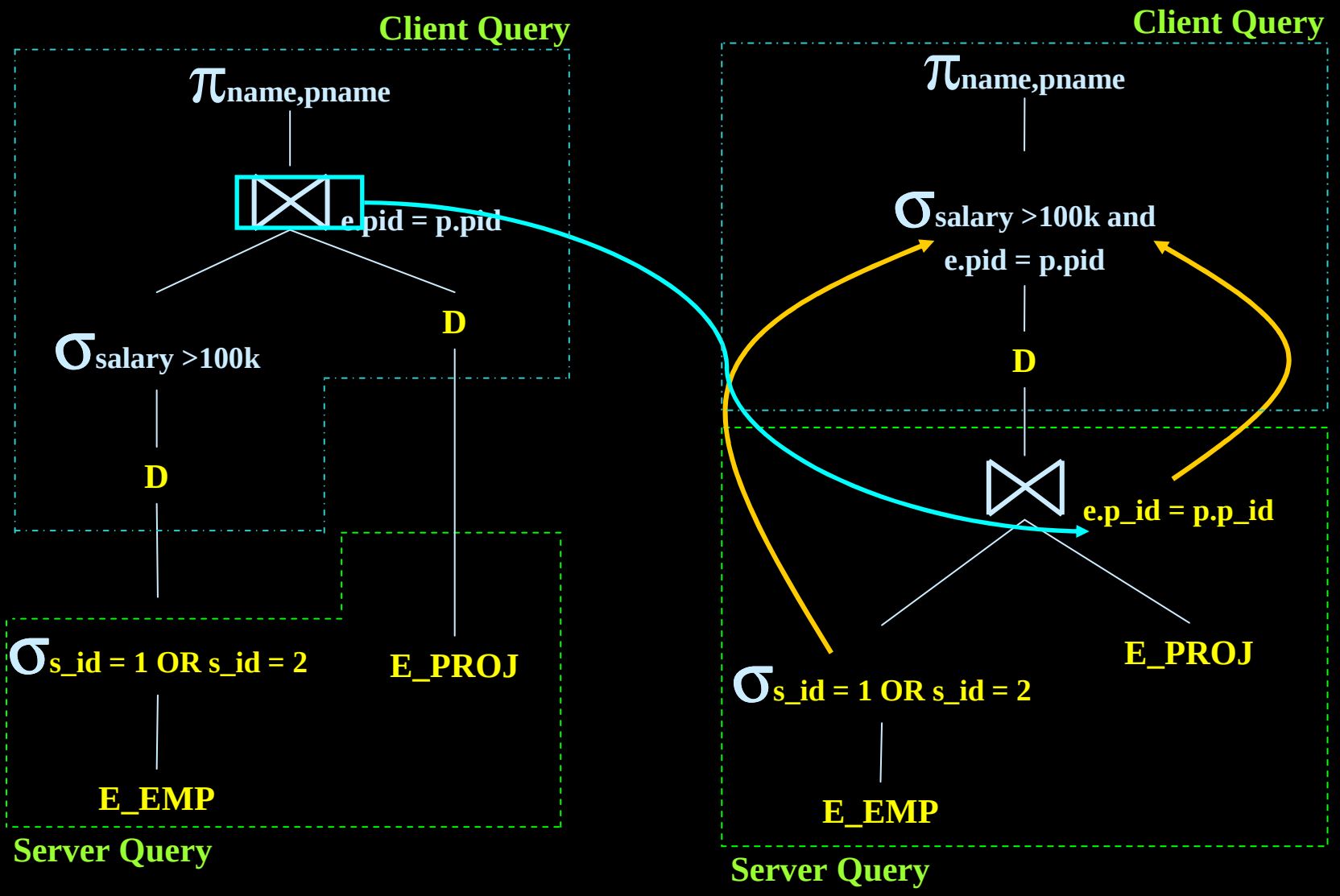
Q: SELECT name, pname FROM emp, proj
WHERE emp.pid=proj.pid AND salary > 100k



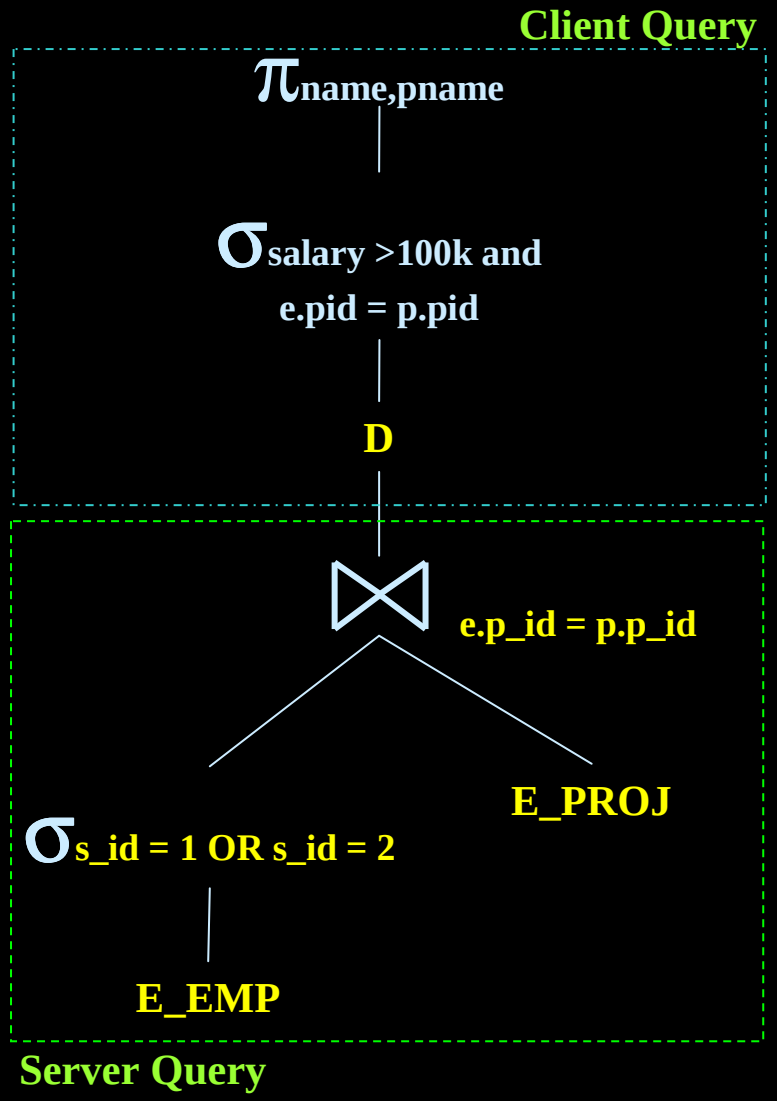
Décomposition de requête (2)



Décomposition de requête (3)



Décomposition de requête (4)



Q: SELECT name, pname
FROM emp, proj
WHERE emp.pid=proj.pid AND
salary > 100k

Q^S: SELECT e_emp.etuple,
e_proj.etuple
FROM e_emp, e_proj
WHERE e.p_id=p.p_id AND
s_id = 1 OR s_id = 2

Q^C: SELECT name, pname
FROM temp
WHERE emp.pid=proj.pid AND
salary > 100k

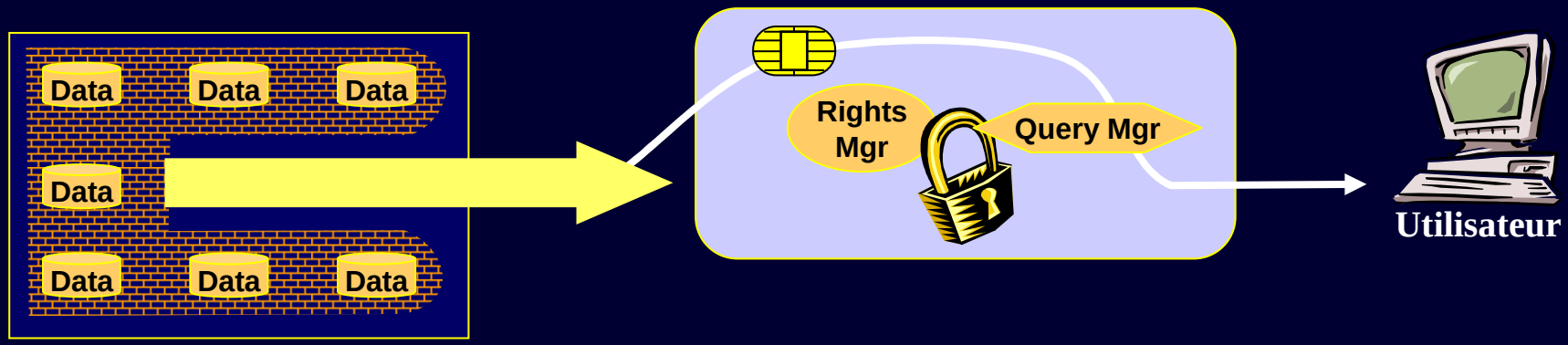
Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L’approche serveur
- L’approche client
- L’approche matérielle (Trusted Platform Module – Trusted Computing Base)
- Conclusion

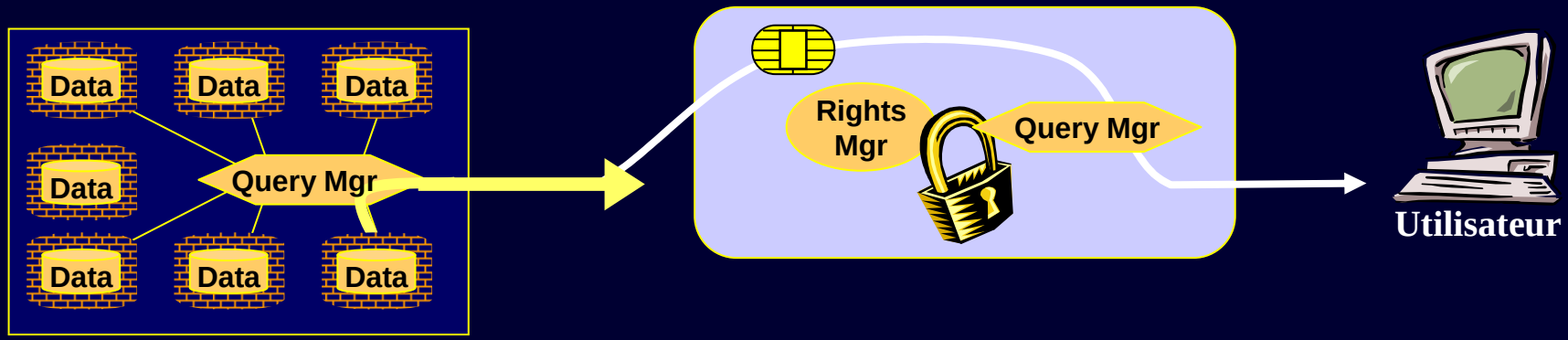
L’approche à base de matériel sûr

- TPM (Trusted Platform Module) sur le client
  - Résiste aux attaques sur le client
    - Gestion sécurisée des droits et des clés
  - Permet un partage sécurisé de la BD chiffrée sur le serveur
- Des instances
  - Chip sécurisé embarqué dans un token USB
  - Co-processeur sécurisé (type IBM 4758, puce Fritz, ARM TrustZone …)
  - Carte SIM + téléphone
  - …

Approche TPM
Tradeoff : confidentialité vs performance



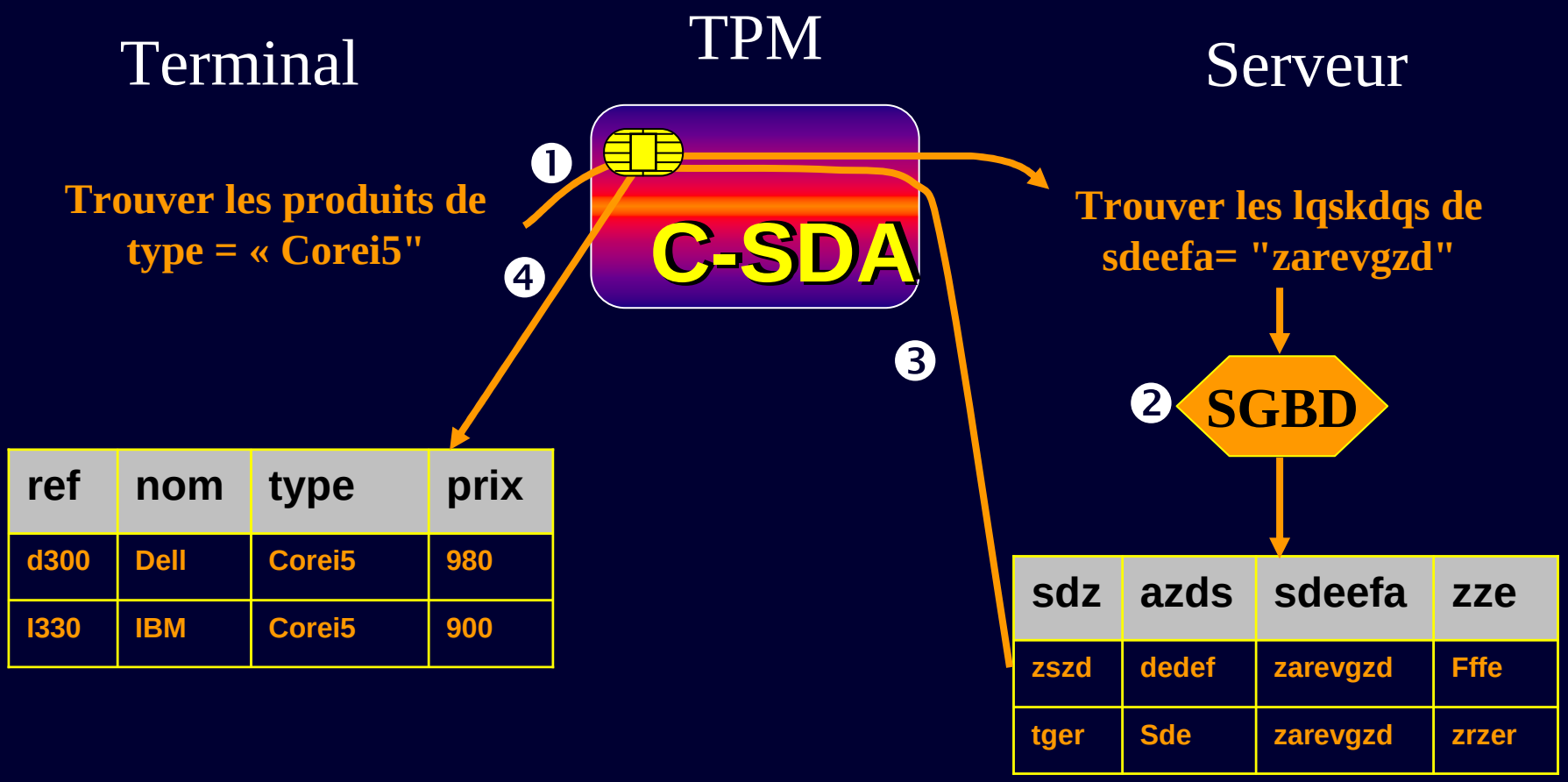
Confidentialité maximale, performances minimales



Confidentialité et performance dépendent du grain et de la méthode de chiffrement

Traitement d'une requête « simple »

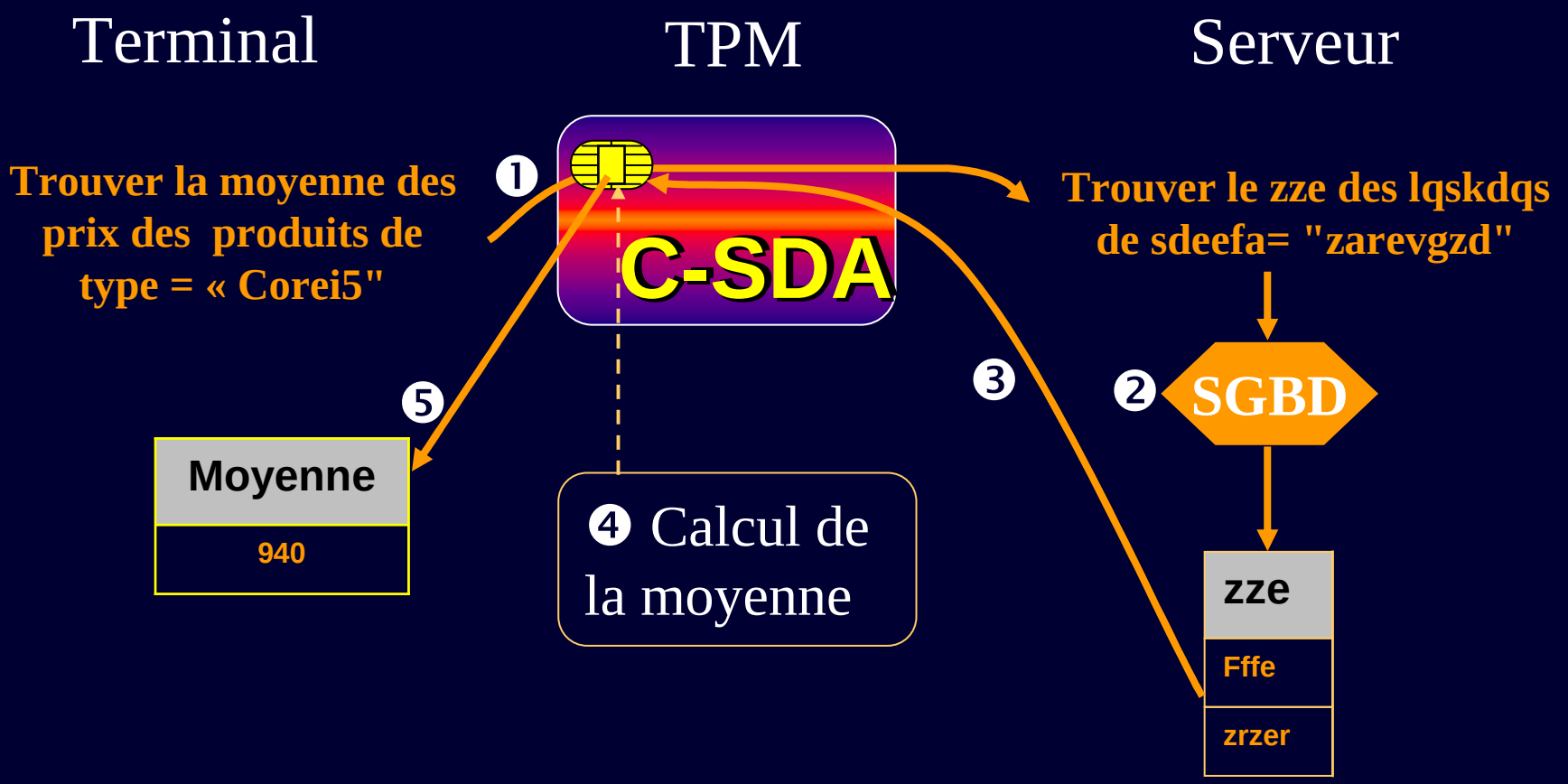
- Le TPM intercepte les requêtes de Select et les traduit



ref	nom	type	prix
d300	Dell	Corei5	980
I330	IBM	Corei5	900

sdz	azds	sdeefa	zze
zszd	dedef	zarevgzd	Fffe
tger	Sde	zarevgzd	zrzer

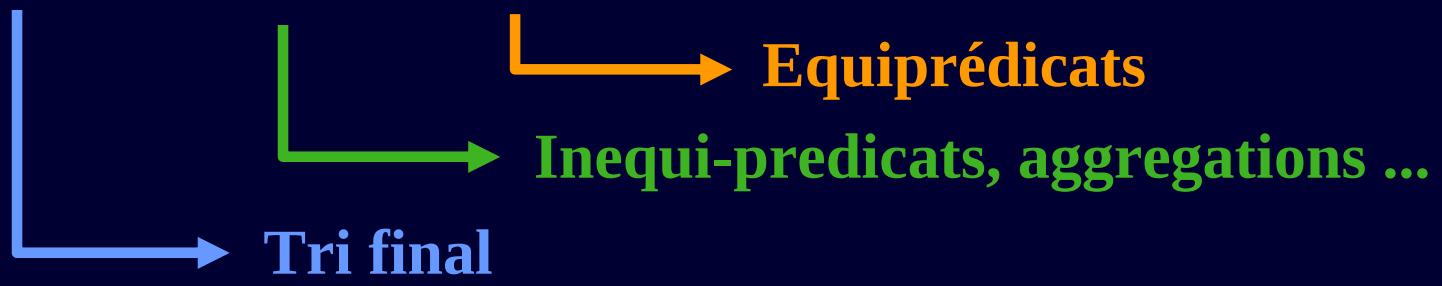
Traitement d'une requête plus complexe



La partie des requêtes non évaluable sur les données chiffrées est évaluée par le TPM (prédicats <, >, fonctions de calcul, etc..)

Décomposition d'une requête

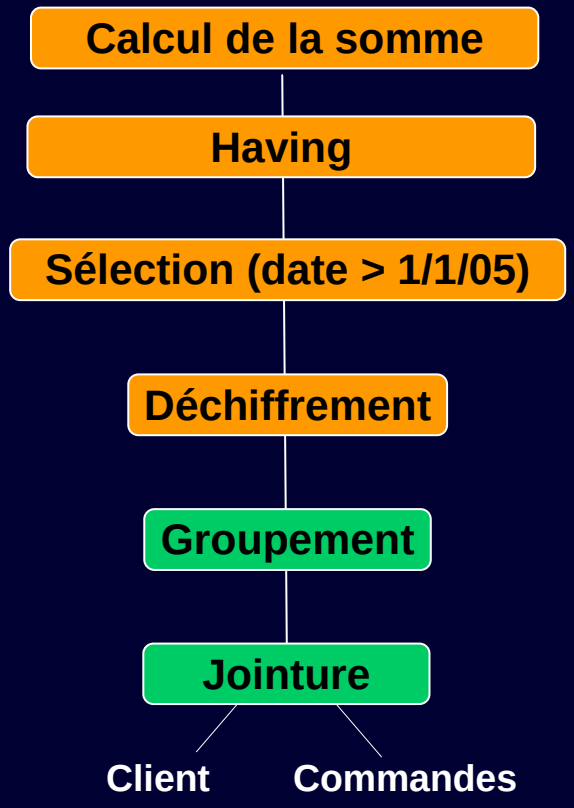
- $Q = Q_{term} \circ Q_{tpm} \circ Q_{server}$

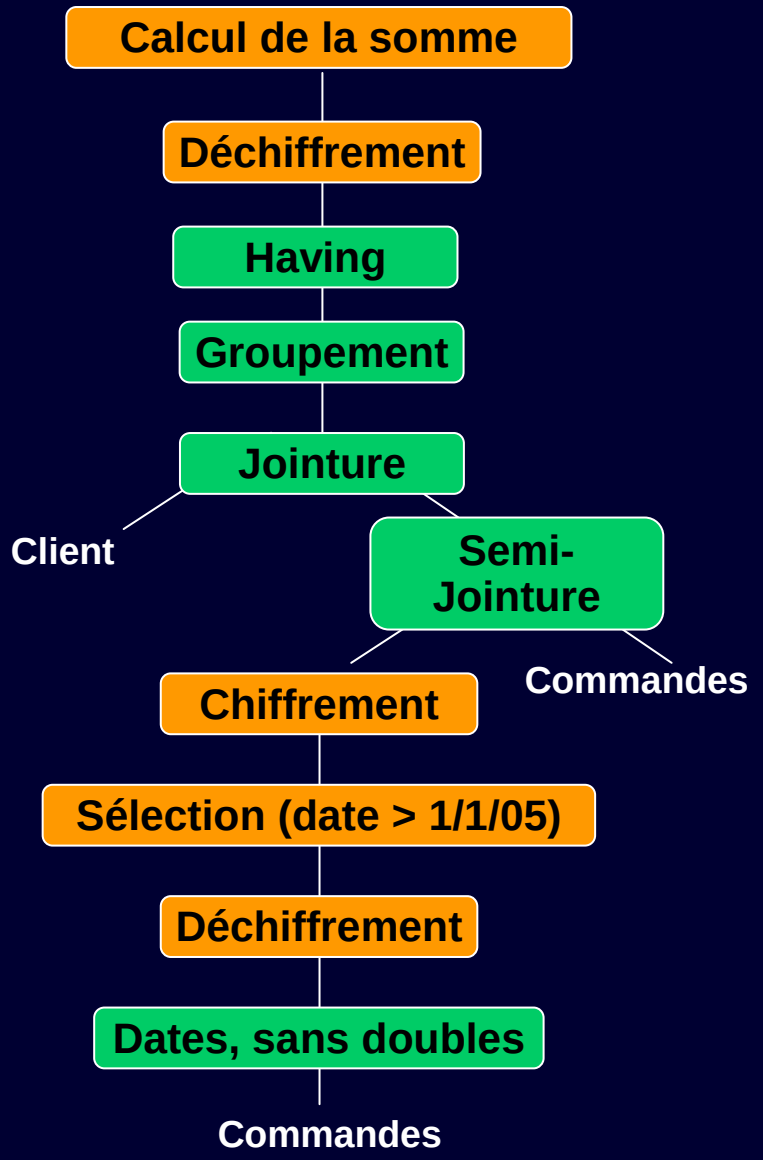



```
SELECT      C.Id, C.name, sum(O.amount)
FROM        Customers C, Orders O
WHERE       C.Id = O.CustId and O.date > 2005
GROUP BY    C.Id, C.name
HAVING      count(*) >= 10
ORDER BY    C.name
```

Exemple d'exécution et optimisation

Nom des clients et sommes à facturer pour les commandes passées après le 1/1/05, pour les clients ayant passé plus de 10 de ces commandes





'*Isoler*' les données les plus *sensibles*

- **Isoler des données sans compliquer l'évaluation**
  - Remplacer les données sensibles de la base par des indices dans un "domaine sensible" stocké par le TPM
- **Peut être vu comme un mode de chiffrement *incassable***
- **la propriété a = b ⇔ chiffre (a) = chiffre (b) est préservée**
- ***/!\ Utile si les données sensibles sont uniques ! (comme le code de CB)***

N°	Nom	Code CB
4	Dupond	0454255782
8	Durand	0450500609
9	Dutronc	0456589413
13	Duval	0454547898
15	Dussol	0455121236

N°	Nom	Code CB
4	Dupond	1
8	Durand	2
9	Dutronc	3
13	Duval	4
15	Dussol	5

Code CB
0454255782
0450500609
0456589413
0454547898
0455121236

Gestion des droits

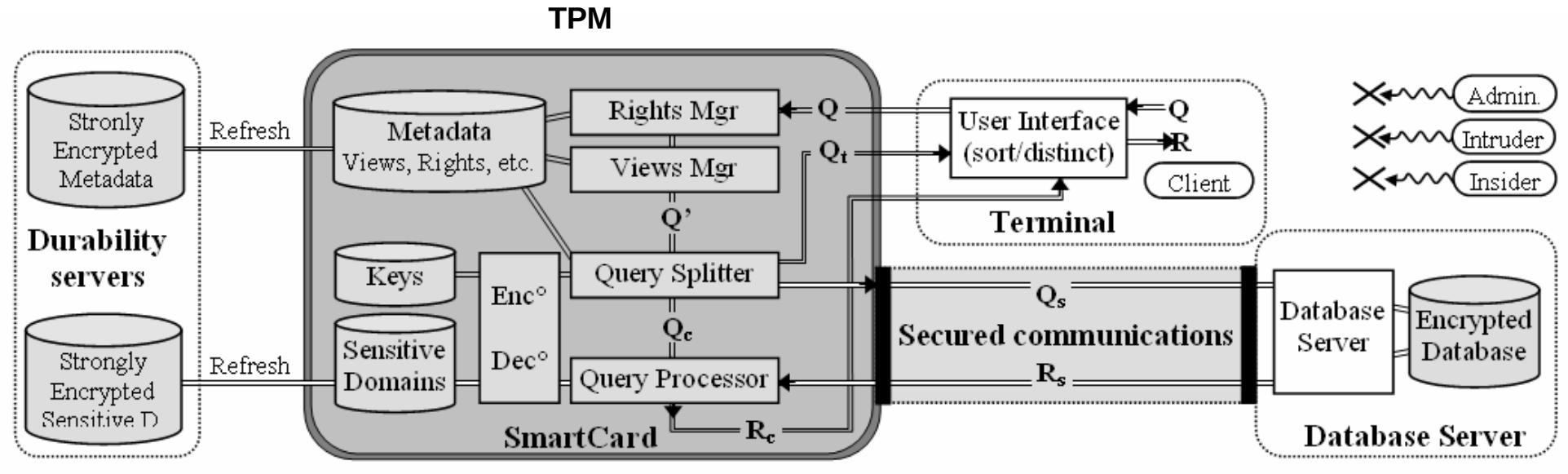
- **Droits BD / droits systèmes**
  - Droits BD : consulter, modifier, supprimer des données
  - Droits système : créer une partition, etc…

- **Les droits BD font partie du noyau de sécurité …**
  - **➔ Les droits BD doivent donc être vérifiés par le TPM !**

- **Les droits BD sont basés sur les vues …**
  - **➔Les vues doivent donc être gérées par le TPM!**

- **Mais les droits et les vues sont des données partagées …**
  - **➔droits et vues (définitions) doivent être stockés sur le serveur !**

Architecture finale



Conclusion

- Approche serveur
  - Solution privilégiée actuellement par les grands éditeurs
  - Non résistant aux attaques internes
    - Mais possibilité de rendre ces attaques complexes
  - Problème de résistance aux attaques externes
    - Indexation des colonnes chiffrées => chiffrement préservant l'ordre
  - Tradeoff sécurité-performance
- Approche client
  - Résiste aux attaques internes sur le serveur
  - Ne résiste pas aux attaques du client => BD privées
  - Problèmes de performance décuplés
  - Nombreux problèmes ouverts
    - Indexation, étiquetage, evaluation de requêtes complexes, mises à jour
- Approche client-TPM
  - Les plus sûres
  - Manquent de généralité et de souplesse (difficulté d'intégration dans un SI)
  - Encore un sujet de recherche !

Plan

- Outils cryptographiques
- Application du chiffrement à une base de données
- L’approche serveur
- L’approche client
- L’approche matérielle (TPM - TCB)
- Conclusion

Cryptography: the characters

Alice: (A)

She wants to communicate with Bob!

Bob: (B)

He wants to communicate with Alice!

Marvin: (M)

Marvin is **M**alicious

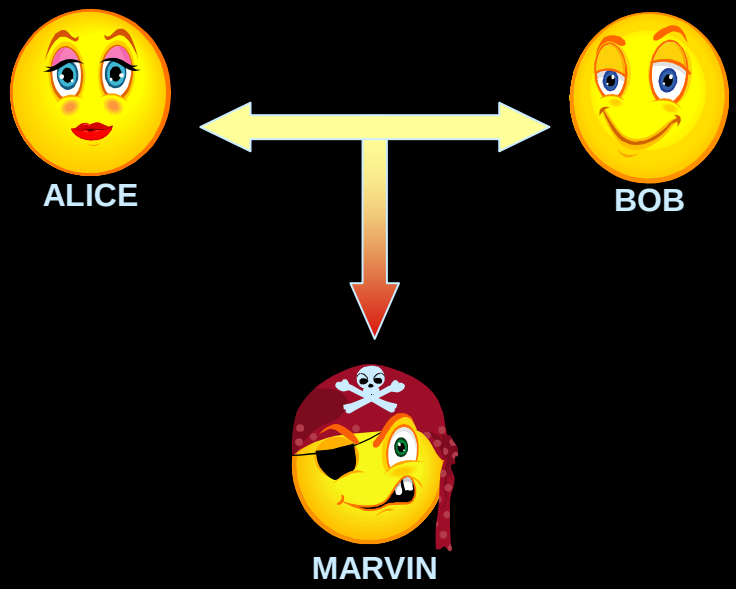
Trent: (T)

Trent is Trusted

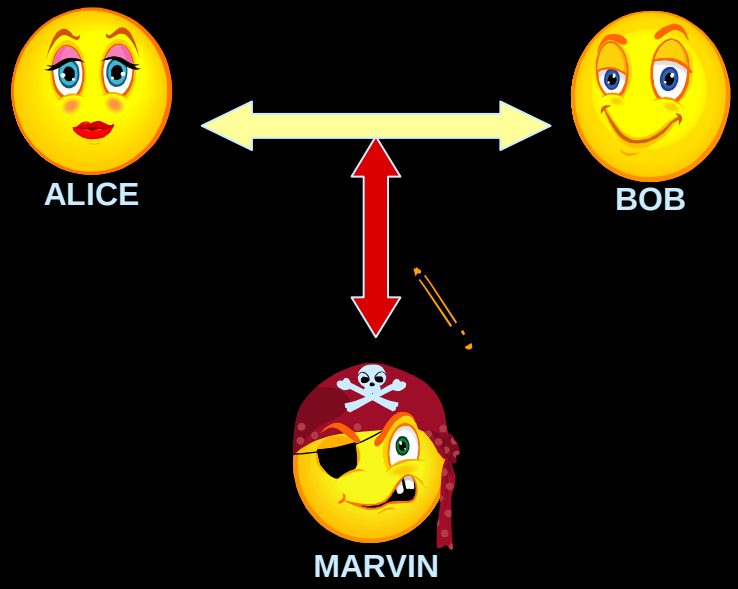
Passive attacks

Active attacks

Marvin has access to unauthorized data



Marvin modifies the transmitted data

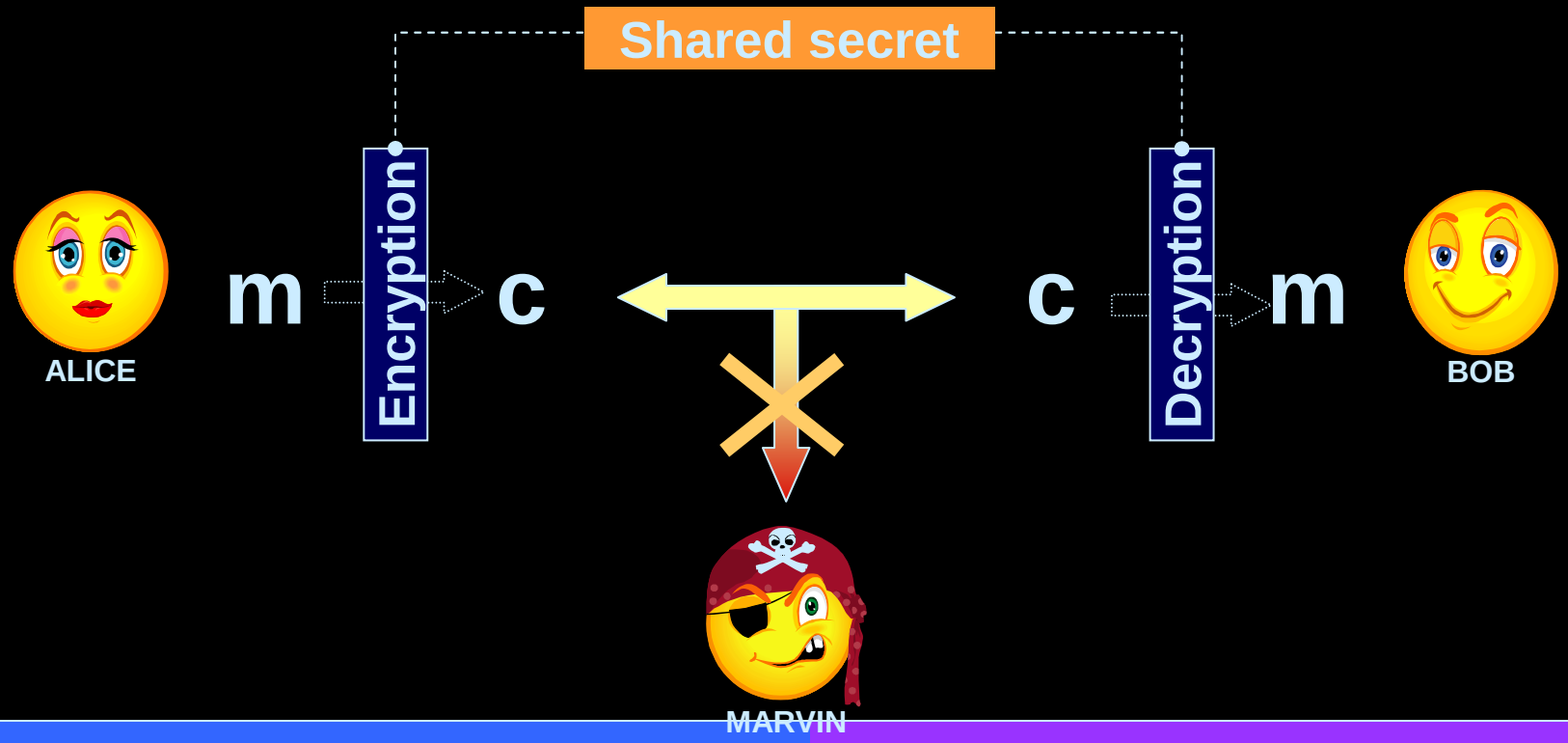


Passive and active attacks

- Passive attacks: threaten **confidentiality**
- Active attacks: threaten **integrity**
  - Identity theft
    - Marvin sends a message to Bob with Alice IDs
  - Data alteration
    - Marvin modifies the content of Alice message
  - Message forging
    - Marvin forges a fake message for Bob or Alice
  - Replay attack
    - Marvin captures a message and sent it to Bob several times
  - Repudiation
    - Alice sends a message to Bob and denies having sent this message
  - Destruction
    - Marvin destroys selectively some messages sent to Bob
  - Delays / reordering
    - Marvin introduces communication delays or reorders messages

Symmetric encryption

- Sender and receiver share a **secret** allowing symmetric encryption and decryption



Lettre de G. Sand à A. de Musset

Je suis très émue de vous dire que j'ai bien compris l'autre soir que vous aviez toujours une envie folle de me faire danser. Je garde le souvenir de votre baiser et je voudrais bien que ce soit là une preuve que je puisse être aimée par vous. Je suis prête à montrer mon affection toute désintéressée et sans calcul, et si vous voulez me voir aussi vous dévoiler sans artifice mon âme toute nue, venez me faire une visite. Nous causerons en amis, franchement. Je vous prouverai que je suis la femme sincère, capable de vous offrir l'affection la plus profonde comme la plus étroite en amitié, en un mot la meilleure preuve que vous puissiez rêver, puisque votre âme est libre. Pensez que la solitude où j'habite est bien longue, bien dure et souvent difficile. Ainsi, en y songeant j'ai l'âme grosse. Accourez donc vite et venez me la faire oublier par l'amour où je veux me mettre.

Kerckhoffs' principle

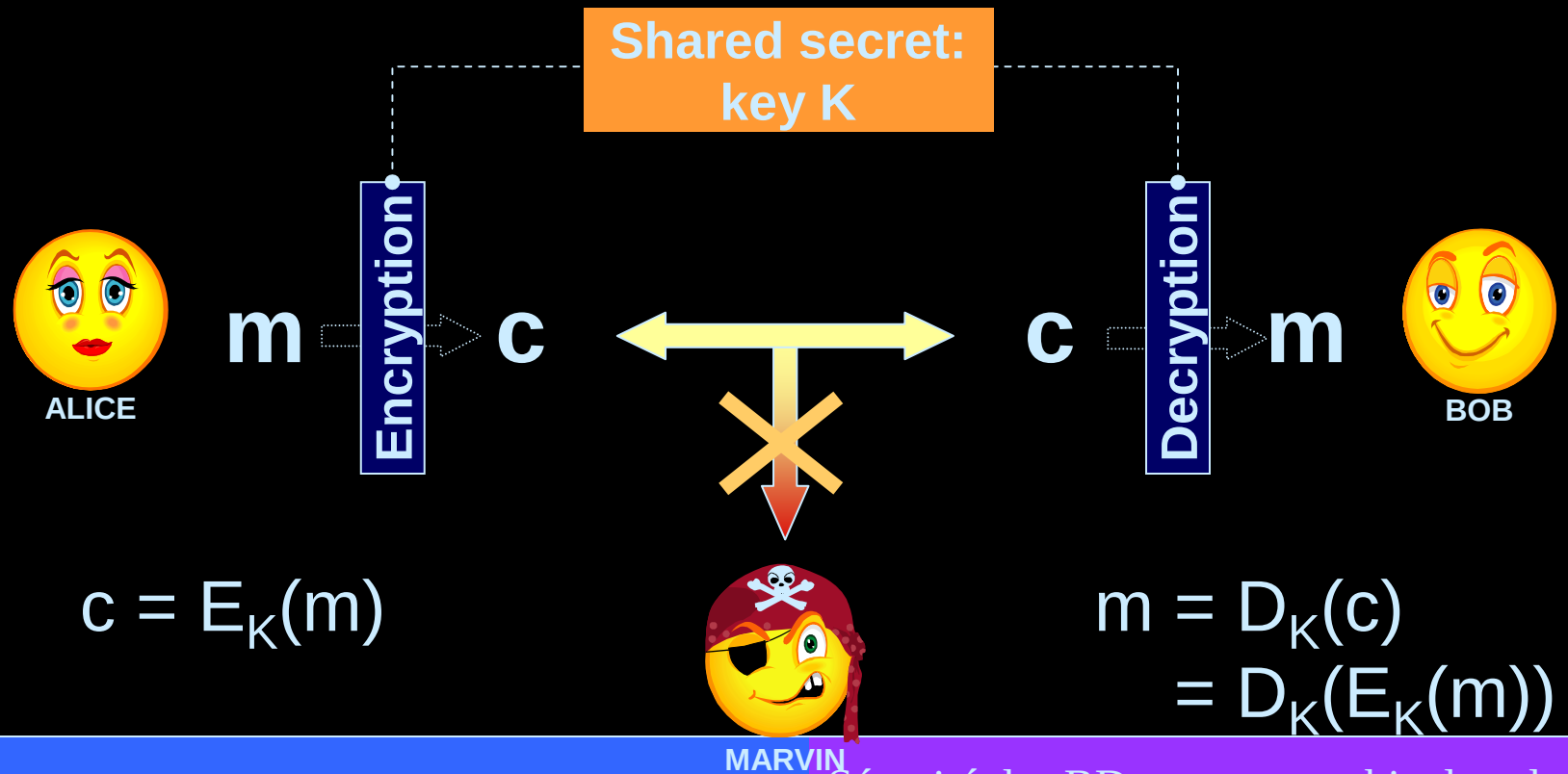
In 1883, the most famous work by Auguste Kerckhoffs was published: La Cryptographie Militaire (military cryptography). This book set forth desiderata for encryption systems…

– The encryption system **must not be required to be secret**, and it must be able to **fall into the hands of the enemy** without inconvenience;

– The system must be **practically**, if **not mathematically**, indecipherable;

Symmetric encryption: Secret key encryption

- All the details of the system, including the encryption and decryption functions are known, **except the key**
- **Security is only based on the secrecy of the key K**



Attacking a symmetric cipher

- The attacker knows the ciphertext **c**

➔Find the plaintext **m**, or better the key **K**

- The attacker knows couples plaintext/ciphertext (**m**, **c)**

➔Find the key **K**, or at least be able to decrypt other messages

Attacking a symmetric cipher (2)

- Example :
  - Caesar cipher (used for military purposes by Julius Caesar)

$$E_n(x) = (x + n) \mod 26.$$
$$D_n(x) = (x - n) \mod 26.$$

  - **Plaintext:** attackatonce
  - **Ciphertext:** exxegoexsrgi

  ➔ Very easy to attack !

- Given the small number of possible shifts (26), the key can be found in 13 operations, in average…

Decryption shift	Candidate plaintext
0	exxegoexsrgi
1	dwwdfndwrqfh
2	cvvcemcvqpeg
3	buubdlbupodf
4	attackatonce
5	zsszbjzsnmbd
6	yrryaiyrmlac
…	
23	haahjrhavujl
24	gzzgiqgzutik
25	fyyfhpfytshj

The problem here is the size of the key! (less than 5 bits !)

Attacking an simple alphabetic substitution

- Assume a **random** alphabetic substitution such as

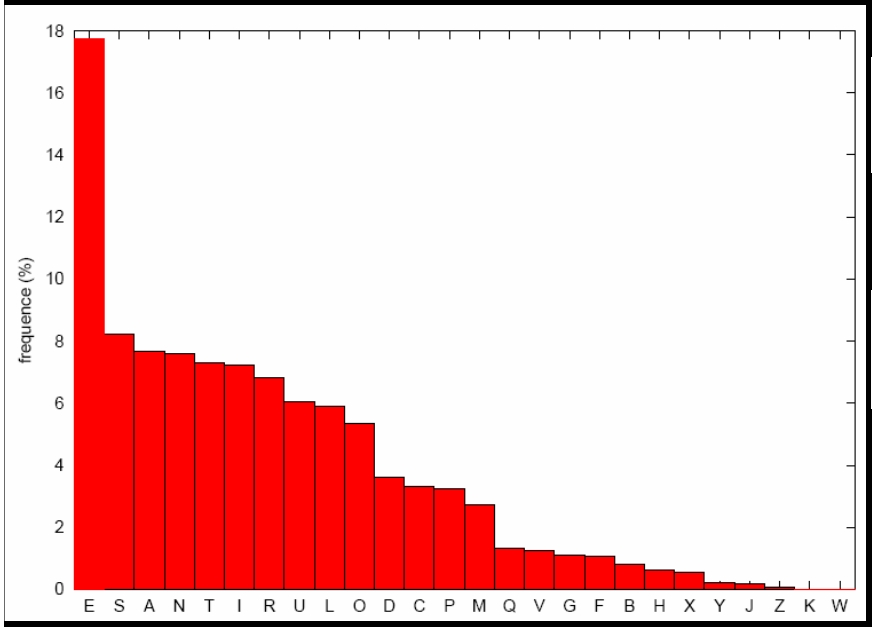
x	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
$f(x)$	C	E	M	R	B	O	Y	D	Q	P	F	U	T	G	V	A	H	W	N	I	X	L	J	Z	S	K

- Can we easily retrieve the key (i.e., the table) knowing a ciphertext ?

(NB: French message)

nvxlbgi avxw n ctxnbw ubn dvttbn r bhxqacyb
awbggbgi rbn cueciwvn lcnibn vqnbcxz rbn tbwn
hxq nxqlbgi qgrvubgin mvtacygvgn rb lvscyb
ub gclqwb yuqnncgi nxw ubn yvxoowbn ctbwn
c abqgb ubn vgi qun rbavnbn nxw ubn aucgmdbn
hxb mbn wvqn rb u ckxw tcucrwvqin bi dvgibxz
ucqnnbgi aqibxnbtbgi ubxwn ywcgrbn cqubn eucgmdbn
mvttb rbn clqwvgn iwcqgbw c mvib r bxz
mb lvscybxw cqub mvttb qu bni ycxmdb bi lbxub
uxq gcyxbwb nq ebcx hx qu bni mvtqhxb bi ucqr
u xg cycmb nvg ebm clbm xg ewxubyxbxub
u cxiwb tqtb bg evqicgi u qgoqwtb hxq lvucqi
ub avbib bni nbteuceub cx awqgmb rbn gxbbn
hxq dcgib uc ibtabib bi nb wqi rb u cwmdbw
bzqub nxw ub nvu cx tquqbx rbn dxbbn
nbn cqubn rb ybcgi u btabmdbgi rb tcwmdbw

(1) Frequency analysis: a single letter



Letter frequency in French

In the cipher text

B	N	C	U	X	Q	G	I	W	V
18,7	9,91	7,78	6,90	6,72	6,37	5,84	5,84	5,30	4,60

In French

E	S	A	N	T	I	R	U	L	O
17,8	8,23	7,68	7,61	7,30	7,23	6,81	6,05	5,89	5,34

svxlegi avxw s atxsew ues dvttes r ehxqaaye
aweggegi res aueaiwvs lasies vqseaxz res tews
hxq sxqlegi qgrvuegis mvtaaygvgs re lvsaye
ue galqwe yuqssagi sxw ues yvxoowes atews
a aeqge ues vgi qus reavses sxw ues auagmdes
hxe mes wvqs re u akxw tauarwvqis ei dvgiexz
uaqssegi aqiexsetegi uexws ywagres aques euagmdes
mvtte res alqwvgs iwaqgew a mvie r exz
me lvsayexw aque mvtte qu esi yaxmde ei lexue
uxq gayxewe sq eeax hx qu esi mvtqhxe ei uaqr
u xg ayame svg eem alem xg ewxueyxexue
u axiwe tqte eg evqiagi u qgoqwte hxq lvuaqi
ue aveie esi seteuaeue ax awqgme res gxees
hxq dagie ua ietaeie ei se wqi re u awmdew
ezque sxw ue svu ax tquqex res dxees
ses aques re yeagi u etaemdegi re tawmdew

Conclusion

B ⟶ E
N ⟶ S
C ⟶ A

(2) Frequency analysis: bigrams

In the cipher text

ES	UE	GI	RE	EG	EX	IE	SE	QU	TE	UA	EW	AG	AQ	HX	XW
25	17	13	12	9	8	8	8	8	8	8	7	7	7	7	7

In French

ES	LE	EN	DE	RE	NT	ON	ER	TE	SE	ET	EL	QU	AN	NE	OU	AI

U ⟶ L
R ⟶ D
G ⟶ N
Q ⟶ I

In the cipher text

ES	LE	NI	DE	EN	EX	IE	SE	IL	TE	LA	EW	AN	AI	HX	XW
25	17	13	12	9	8	8	8	8	8	8	7	7	7	7	7

In French

ES	LE	EN	DE	RE	NT	ON	ER	TE	SE	ET	EL	QU	AN	NE	OU	AI

I ⟶ T

(3) Using some common words

svxlent avxw s atxsew les dvttes d ehxiaaye
awennent des aleatwvs lastes viseaxz des tews
hxi sxilent indvlents mvtaaynvns de lvsaye
le naliwe ylissant sxw les yvxoowes atews
a aeine les vnt ils deavses sxw les alanmdes
hxe mes wvis de l akxw taladwvits et dvntexz
laissent aitexsetent lexws ywandes ailes elanmdes
mvtte des aliwvns twainew a mvte d exz
me lvsayexw aile mvtte il est yaxmde et lexle
lxi nayxewe si eeax hx il est mvtihxe et laid
l xn ayame svn eem alem xn ewxleyxexle
l axtwe tite en evitant l inoiwte hxi lvlait
le avete est setelaele ax awinme des nxees
hxi dante la tetaete et se wit de l awmdew
ezile sxw le svl ax tiliex des dxees
ses ailes de yeant l etaemdent de tawmdew

indvlent vnt	V ⟶ O
oiseaxz	X ⟶ U Z ⟶ X
a aeine	A ⟶ P
leuws	W ⟶ R
taladroits	T ⟶ M
yrandes	Y ⟶ G

(4) and more common words…

soulent pour s amuser les dommes d ehuipage
prennent des aleatros lastes oiseaux des mers
hui suilent indolents mompagnons de losage
le nalire glissant sur les gouoores amers
a peine les ont ils deposes sur les planmdes
hue mes rois de l akur maladroits et donteux
laissent piteusement leurs grandes ailes elanmdes
momme des alirons trainer a mote d eux
me losageur aile momme il est gaumde et leule
lui naguere si eeau hu il est momihue et laid
l un agame son eem alem un erulegueule
l autre mime en eoitant l inoirme hui lolait
le poete est semelaele au prinme des nuees
hui dante la tempete et se rit de l armder
exile sur le sol au milieu des duees
ses ailes de geant l empemdent de marmder

soulent L → V

dommes D → H

ehuipage H → Q

aleatros E → B

mompagnons M → C

etc.

F

Charles Baudelaire

L'Albatros

Souvent, pour s'amuser, les hommes d'équipage
Prennent des albatros, vastes oiseaux des mers,
Qui suivent, indolents compagnons de voyage,
Le navire glissant sur les gouffres amers.

A peine les ont-ils déposés sur les planches,
Que ces rois de l'azur, maladroits et honteux,
Laissent piteusement leurs grandes ailes blanches
Comme des avirons traîner à côté d'eux.

Ce voyageur ailé, comme il est gauche et veule!
Lui, naguère si beau, qu'il est comique et laid!
L'un agace son bec avec un brûle-gueule,
L'autre mime, en boitant, l'infirme qui volait!

Le Poète est semblable au prince des nuées
Qui hante la tempête et se rit de l'archer;
Exilé sur le sol au milieu des huées,
Ses ailes de géant l'empêchent de marcher.

Simple alphabetic substitution

The simple alphabetic substitution cipher is:

- Class of Cipher: Block cipher
  - The message is encrypted by blocks of fixed size (n bits)
- Mode of operation: ECB
  - Electronic Code Book
  - The message is split in n bits blocks, each encrypted separately

- Why the attack was so easy?
  - The algorithm is too weak?
  - The encryption key is too short?
  - The encryption block is too small?
  - Other reasons?

Some symmetric block ciphers

- DES
  - Data Encryption Standard (1976 - 1997)
  - Encrypts 64 bits blocks
  - Encryption and Decryption are the same algorithms
  - The encryption key is 56 bits
- 3DES
  - Used as a replacement for DES between 1997 and 2001
  - Uses three 56 bits keys
  - tripleDES($k_1k_2k_3$, M) = DES(k_3,DES(k_2,DES(k_1,M)))
- RIJNDAEL (AES)
  - Used since 2001 and Encryption standard since 2002
  - Authors: Joan Daemen and Vincent Rijmen ➔ Rijndael
  - Winners of the AES competition

Attacks on DES, 3DES and AES

- DES
  - 1997: 39 days on 10 000 Pentium
  - 1998: Deep Crack breaks a DES key in 56 hours (250 000 US$)
  - 2007: 6.4 days on a $10,000 parallel machine
- 3DES
  - The best attack known on 3-key 3DES requires around 2^{32} known plaintexts, 2^{113} steps, 2^{90} single DES encryptions, and 2^{88} memory !!
  - 3DES is computationally secure (now)
- AES
  - Only Side Channel Attacks were successful on AES
  - See http://www.cryptosystem.net/aes/ for more information
  - AES is computationally secure (now)

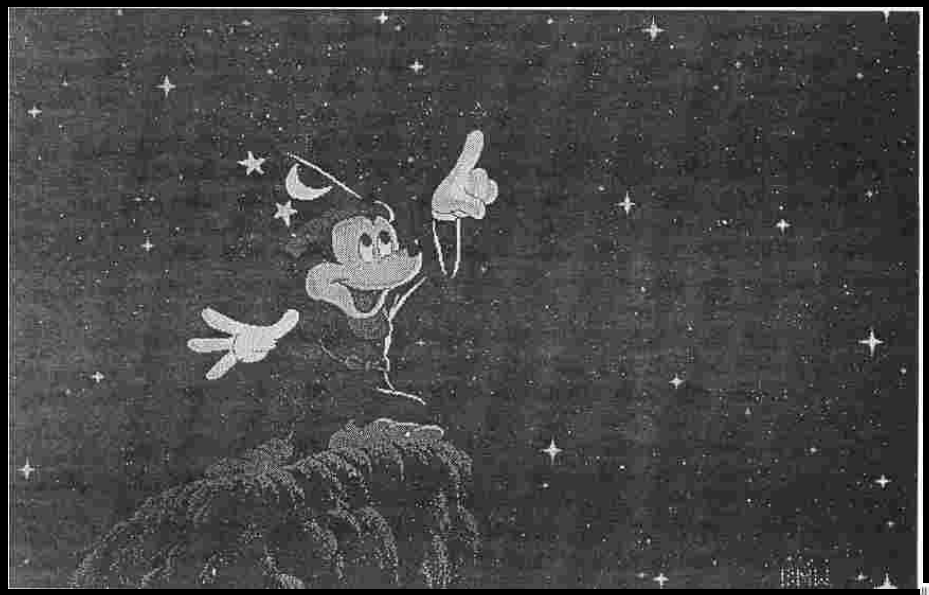
Unconditional vs Computational security

- **Unconditional security**
  - *"the uncertainty in the plaintext, after observing the ciphertext, must be equal to the a priori uncertainty about the plaintext – observation of the ciphertext provides no information whatsoever to an adversary"* [Menezes]

  ➔ The unique possible attack is exhaustive key search
  - To reach unconditional security, the secret key must be as long as the plaintext ! [Shannon 49]

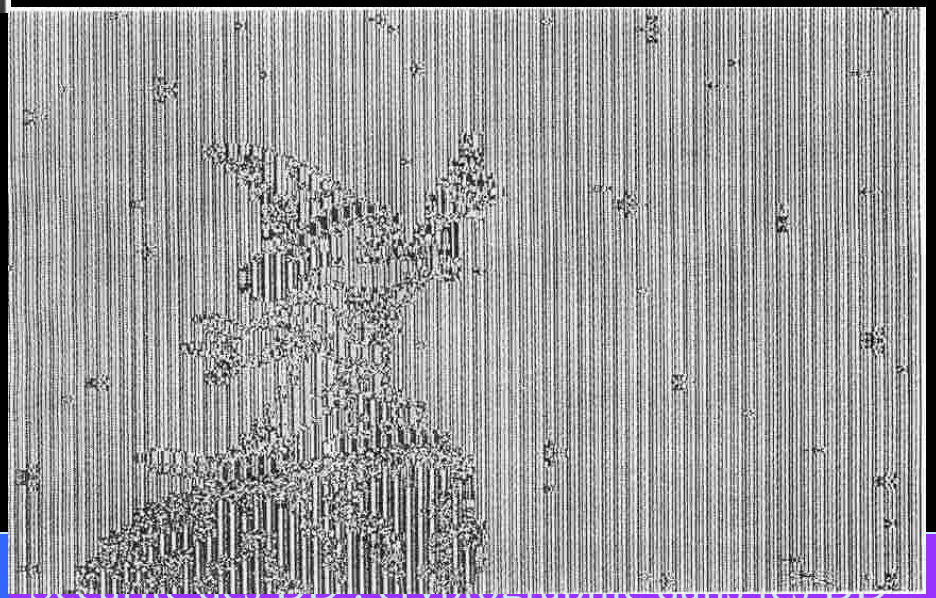
- **Computational security**
  - *"A proposed technique is said to be computationally secure if the perceived level of computation required to defeat it (using the best attack known) exceeds, by a comfortable margin, the computational resources of the hypothesized adversary.*

Triple DES encryption with ECB mode …

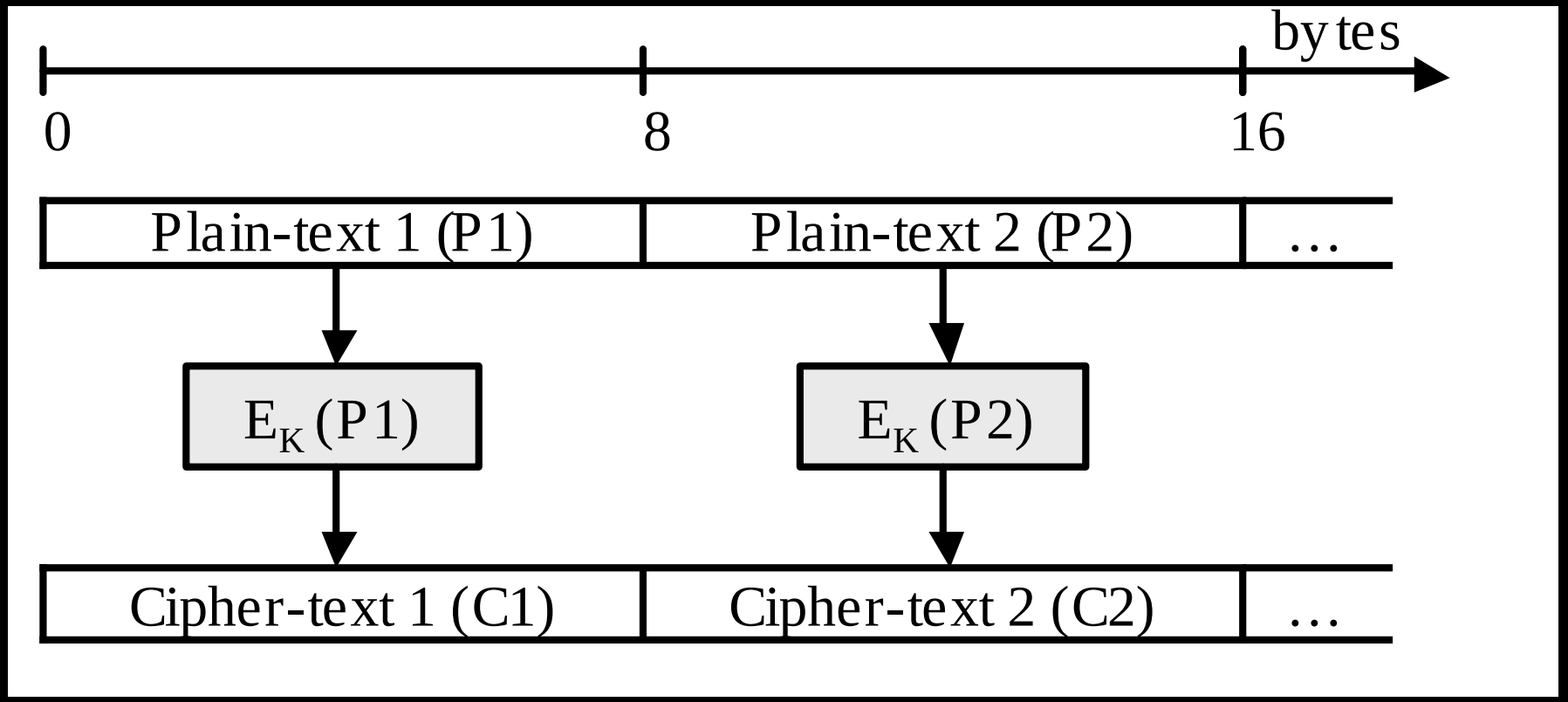


- Can I use 3DES or AES without problems ?

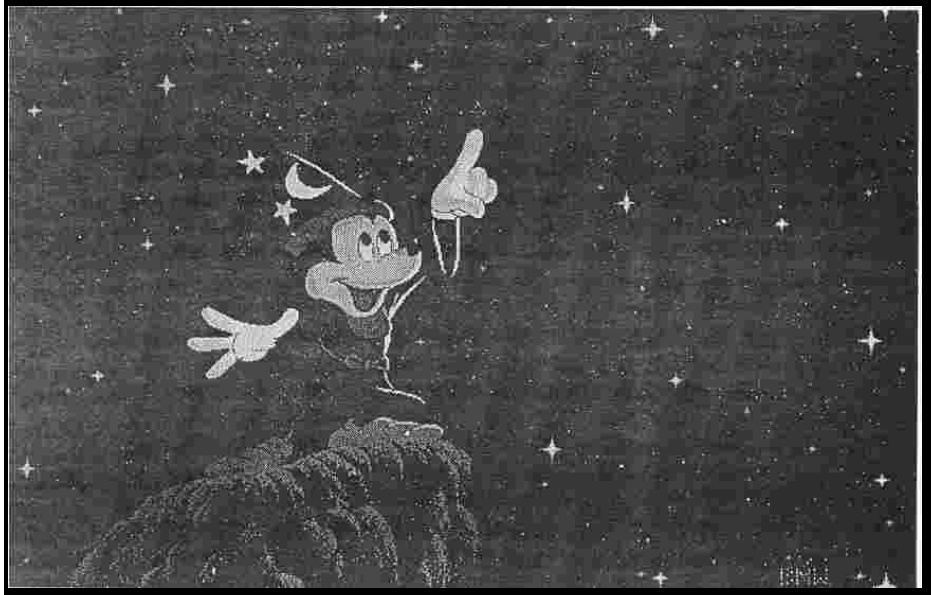
NO!



Encryption mode: Electronic Code Book-ECB

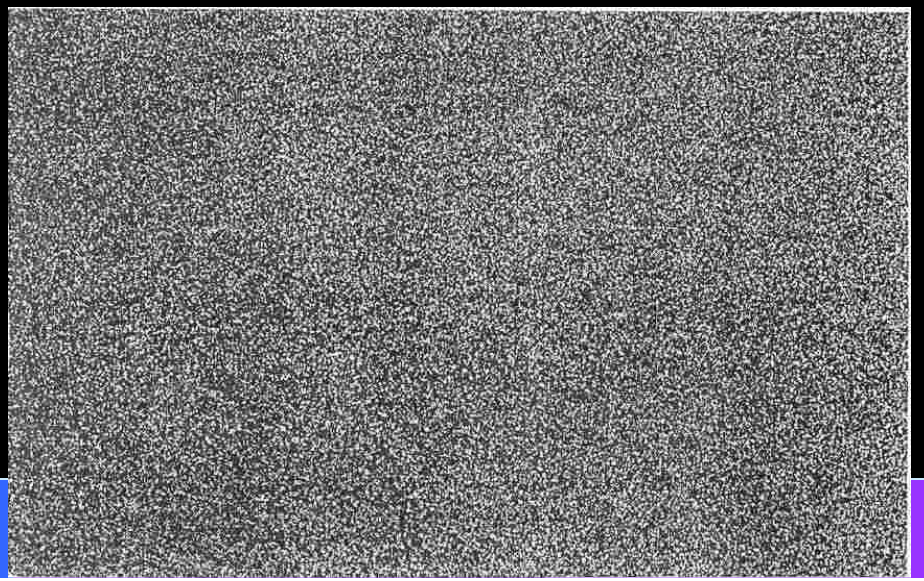


Triple DES encryption with CBC mode …

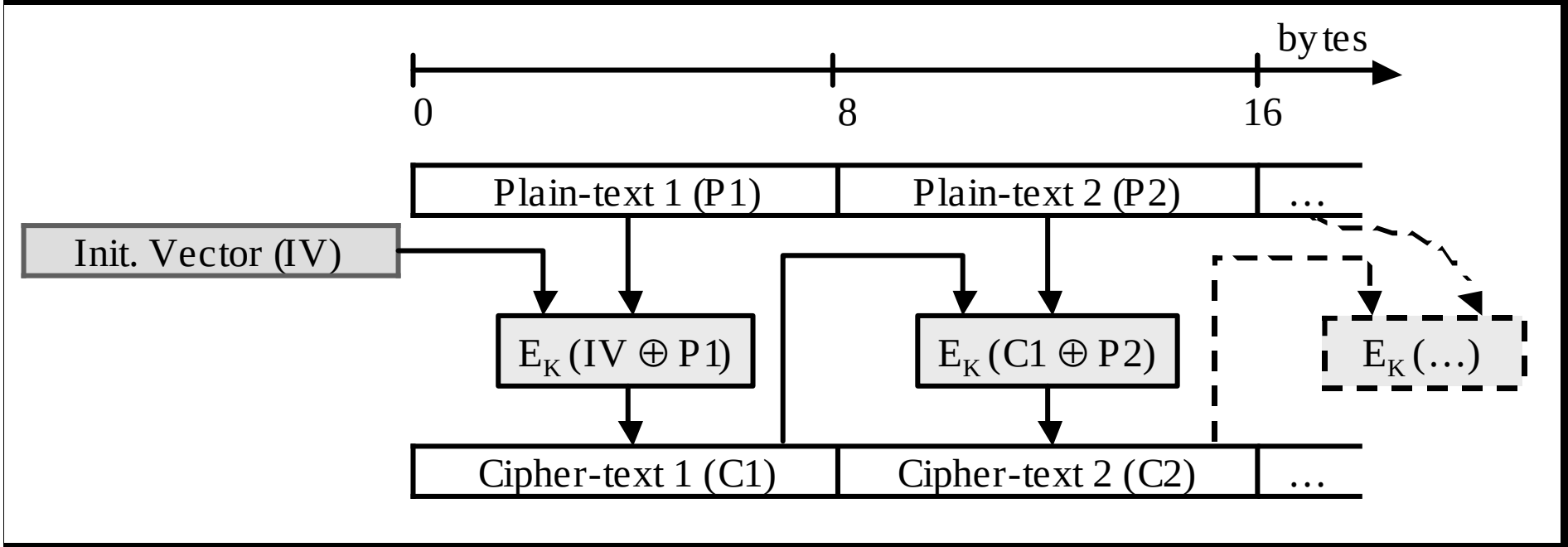


- Can I use 3DES or AES without problems ?

- Yes, if I take care!

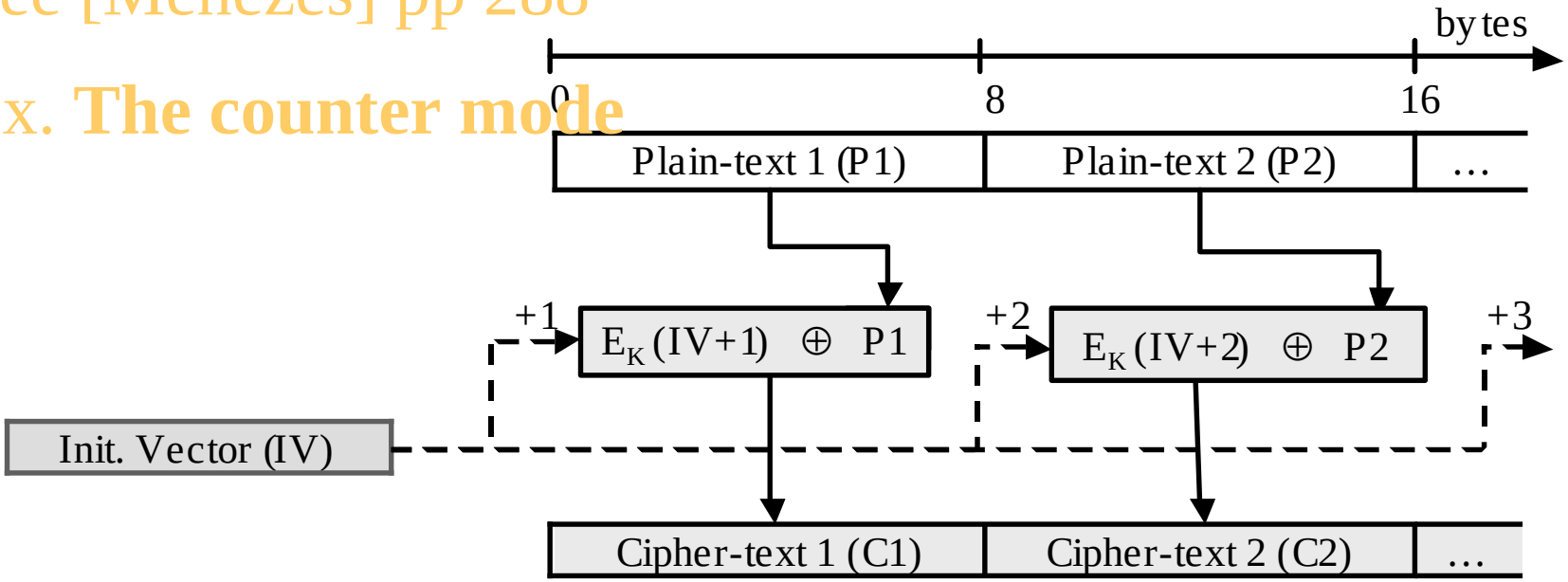


Cipher Block Chaining (CBC)



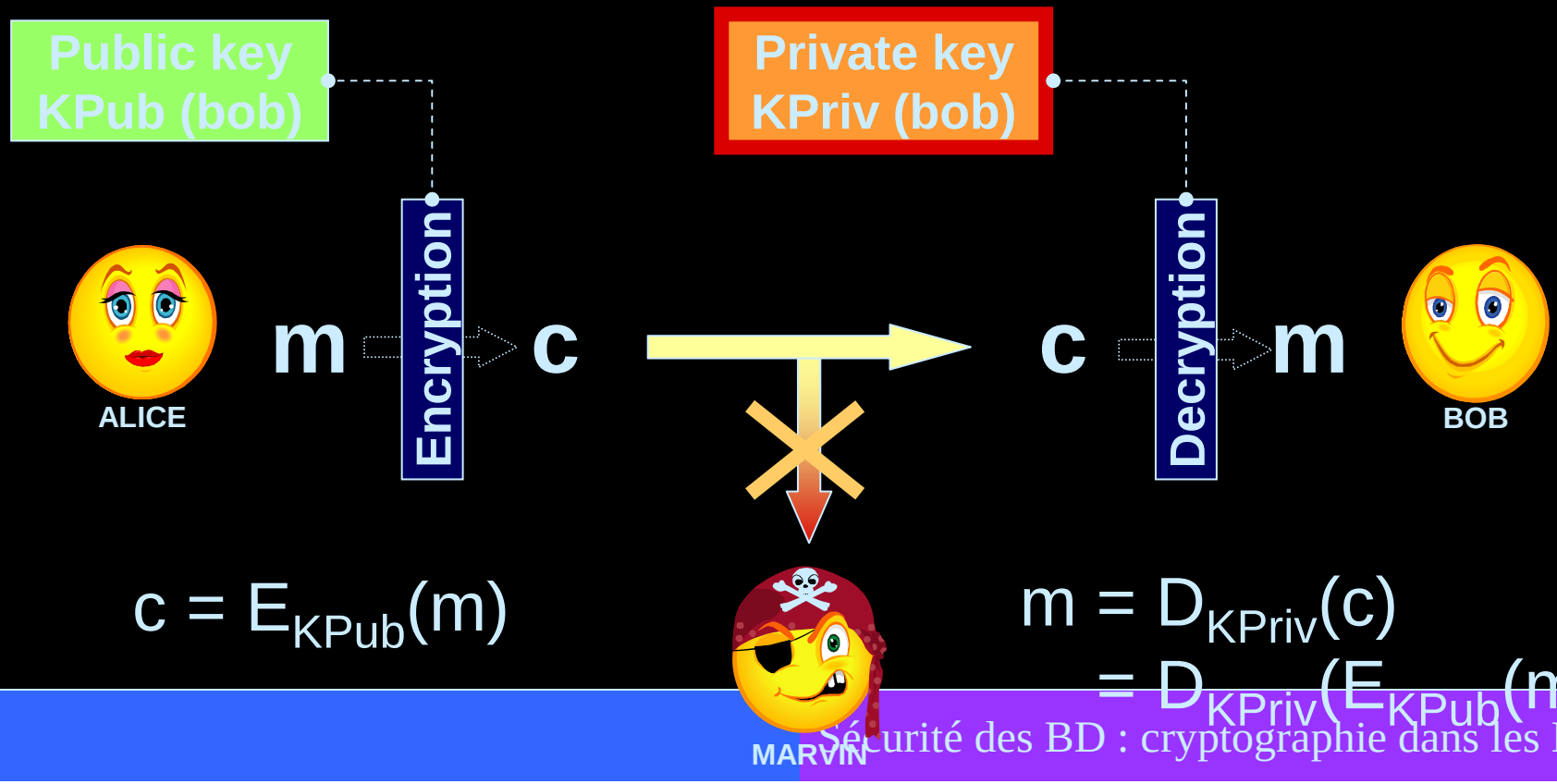
Other modes of operations

- Several modes of operations exists, the adequate one depends on the desired properties of the ciphertext
  - Error propagation
  - Partial decryption
  - Etc.
- See [Menezes] pp 288
- Ex. **The counter mode**



Asymmetric (Public key) encryption

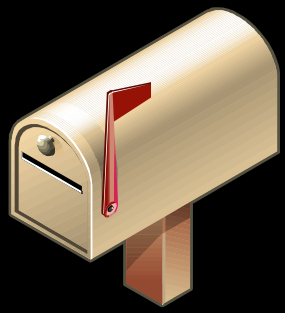
- Each user has
  - a **public key, publicly available** on a directory,
  - **a private key** kept secret
- **No need for a shared secret**



Secret Key vs Public/Private keys

- Alice & Bob have the key
- Alice use the key to deposit a message into the safe
- Bob use the key to retrieve the message from the safe

➔ Only Alice & Bob can exchange messages



- Alice uses the public key (Bob's address) to send a message to Bob
- Bobs uses his private key (the mailbox key) to retrieve the message

➔Anyone can send a message to Bob (his address being public)

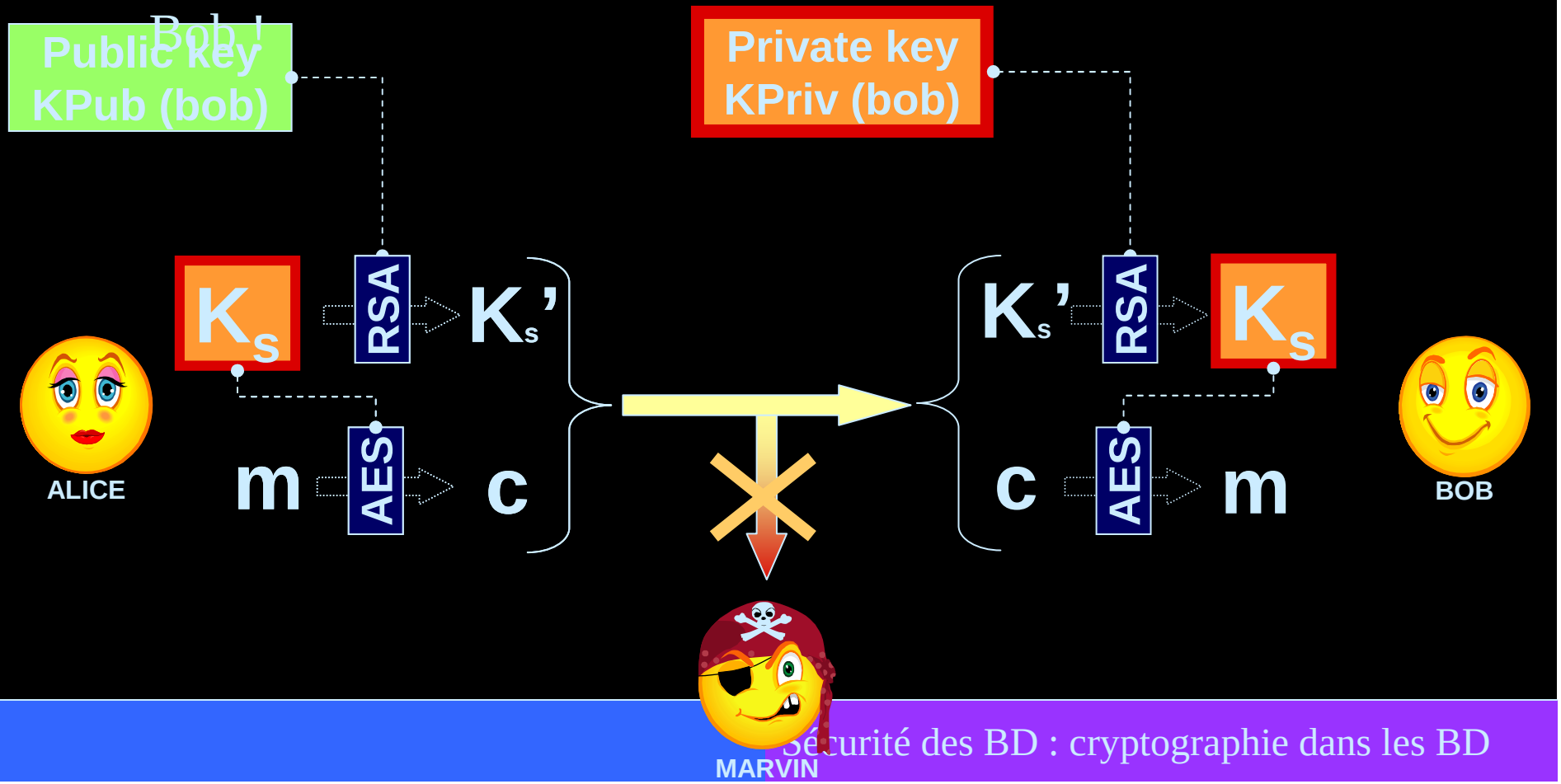
➔Only Bob can retrieve the messages

The RSA asymmetric cipher [Rivest - Shamir - Adleman 78]

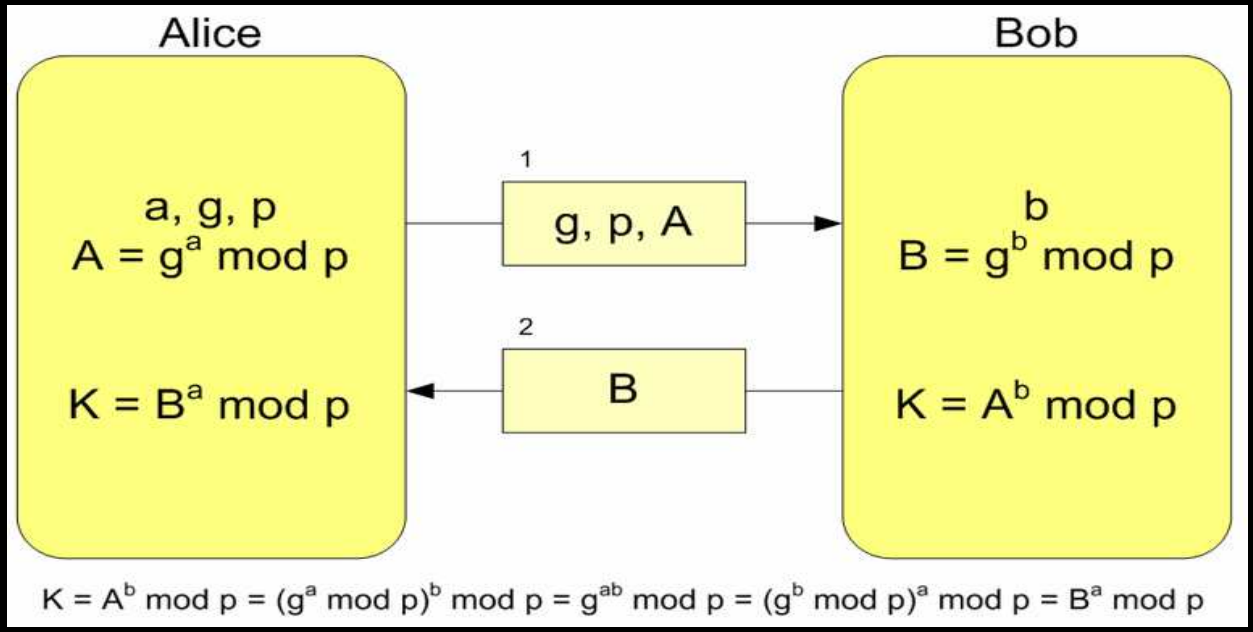
- **Security:** difficulty of factoring large numbers
- **Algorithm**
  - Pick two large random primes p and q
  - Let $N = p \times q$
  - Pick a large integer d relatively prime to $(p-1)\times(q-1)$
  - Find the integer e such that $e*d = 1 \pmod{(p-1)\times(q-1)}$
  - (Public) Encryption key is (e, N): $C = M^e \pmod{N}$
  - Decryption key is (d, N): $M = C^d \pmod{N}$ – ***d must be kept secret!***
- **Performance**
  - Much slower than symmetric algorithm
    - (2-3 orders of magnitude)
  - Generally combined with symmetric algorithms

Hybrid encryption

- Combines the advantages of both encryption methods
  - Performance (symmetric)
  - No shared secret (asymmetric) – anyone can send a message to Bob !



Diffie-Hellmann key exchange : another protocol to exchange a secret



1. Alice and Bob agree to use, e.g., a prime number *p*=23 and base *g*=5.
2. Alice chooses a secret integer ***a*=6**, then sends Bob A = g^a mod *p*
   A = 5^6 mod 23 = 15,625 mod 23 = 8
3. Bob chooses a secret integer ***b*=15**, then sends Alice B = g^b mod *p*
   B = 5^{15} mod 23 = 30,517,578,125 mod 23 = 19
4. Alice computes **s** = B^a mod *p*
   s = 19^6 mod 23 = 47,045,881 mod 23 = 2
5. Bob computes **s** = A^b mod *p*
   s = 8^{15} mod 23 = 35,184,372,088,832 mod 23 = 2

p must be a prime of at least 300 digits, and a and b be at least 100 digits long

Outline

- Are we paranoids ? (no!)
- Cryptographic tools and protocols
  - Basic primitives
    - Encryption
    - Cryptographic hash
  - Combining cryptographic primitives (correctly)
- Overview of database specific tools

Cryptographic hash functions

- Goal: ensure data integrity
  - potentially combined with encryption (see after)
- A hash function *h* must satisfy the following properties
  - maps an input x of arbitrary bit length, to an output h(x) of fixed bit length n.
  - h(x) must be easy to compute
- A hash function h is a cryptographic hash functions if it satisfies (some of) the following properties
  - **preimage resistance:** given a hash value h, it is computationally infeasible to find any x, such that h(x) = h
  - **2nd preimage resistance:** given x and h, such that h(x) = h, it is computationally infeasible to find any second input y, such that h(y)=h(x)=h

Cryptographic hash functions (2)

- Examples:
  - Message Digest 5 (MD5) [Riv92]
  - Secure Hash Algorithm 1 (SHA-1) [NIS95]
- Process the input message by blocks of 512 bits
- The result of the hash function is
  - 160 bits for SHA-1
  - 128 bits for MD5
- Security: SHA-1
  - Find a 2^{nd} preimage requires 2^{160} operations
  - Find a collision requires 2^{80} operations
- Hash functions are publicly known

Outline

- Are we paranoids ? (no!)
- Cryptographic tools and protocols
  - Basic primitives
    - Encryption
    - Cryptographic hash
  - Combining cryptographic primitives (correctly)
- Overview of database specific tools

Combining cryptographic primitives

- Goal: Resist to passive and active attacks
- Passive attacks : threaten **Confidentiality**
- Active attacks: threaten **Integrity**
  - Identity theft
  - Data alteration
  - Message forging
  - Replay attack
  - Repudiation
  - Destruction
  - Delays / reordering
- Approach followed

(1) Confidentiality: Encryption alone

- **Ensures confidentiality** – remind that
  - Symmetric encryption is efficient **but** needs a shared secret
  - Asymmetric encryption is slow **but** does not need a shared secret
    ➔need for a PKI however (Public Key Infrastructure)
  - Combining symmetric and asymmetric is a good option
  - Both must be carefully used
    - Mode of operation: ECB should not be used for messages with repetitive patterns (frequency analysis)
    - Only use known algorithms

- **Encryption does not ensure integrity**
  - This is a common mistake (even encrypted data can be (randomly) altered)

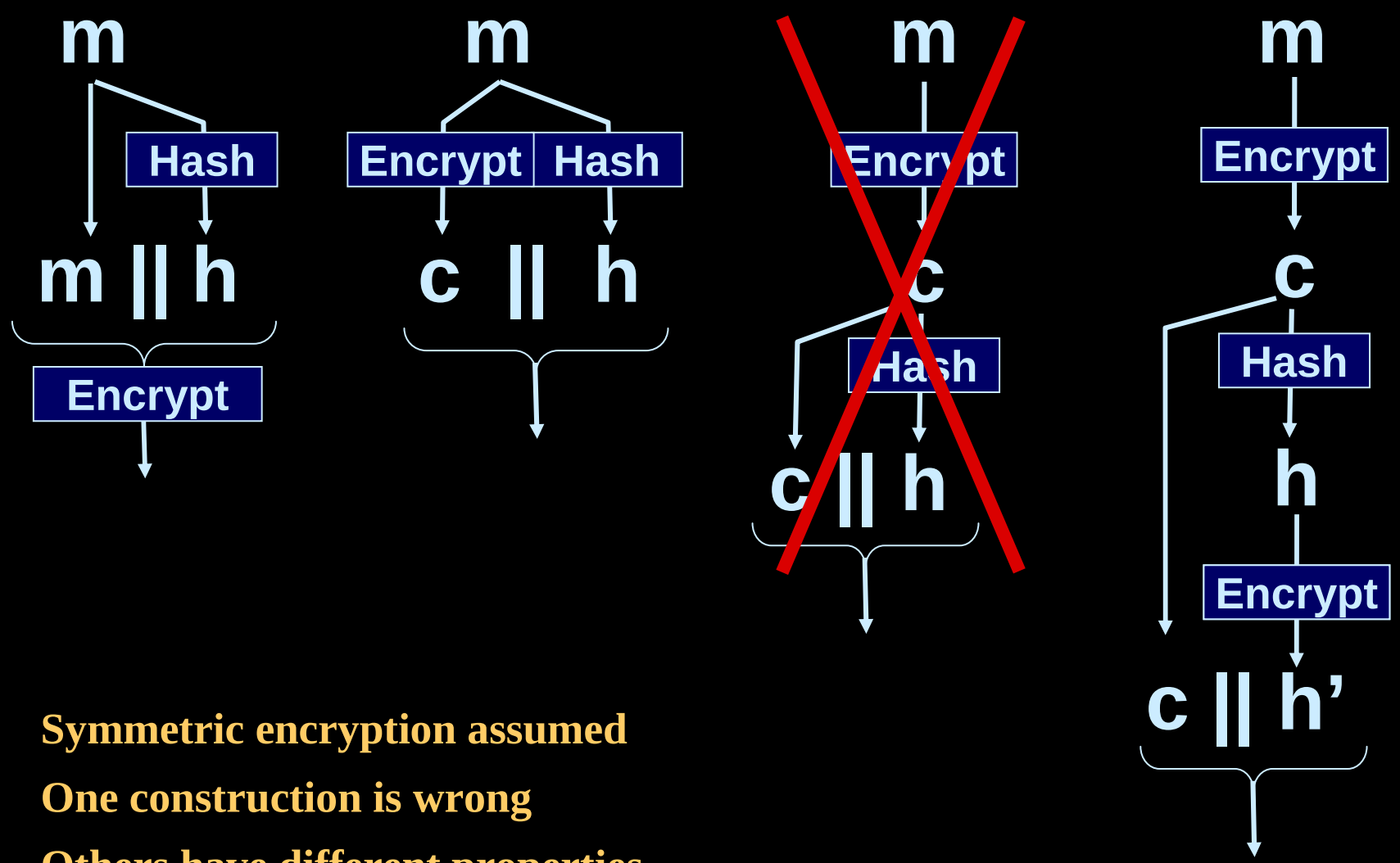
(2) Replay Attack

- Problem:
  - Marvin copies the message and resends it to Bob
- Solution:
  - Includes a unique timestamp (or sequence number) in the message.
  - The receiver keeps timestamps of recently received messages
  - He does not accept a duplicate
- Remark:
  - Obviously, the timestamp integrity must be preserved (see point 4)
- Remark (2):
  - Using sequence number may protect from

(3) Message forging: Nonce

- Problem:
  - Alice and Bob share a session key, K
  - Alice sends a message M1 to Bob and gets M2 back.
  - How can Alice be sure that M2 came from Bob and not from Marvin ?
- Marvin might:
  - send a random string that Alice decrypts (using K) to another random string that looks like a correct response
  - replay an earlier message sent by Bob encrypted with K, that is a possible response (Alice is not a server that maintains a list of timestamps)
- Solution: Include a nonce, N, in M1
  - A random string generated by Alice

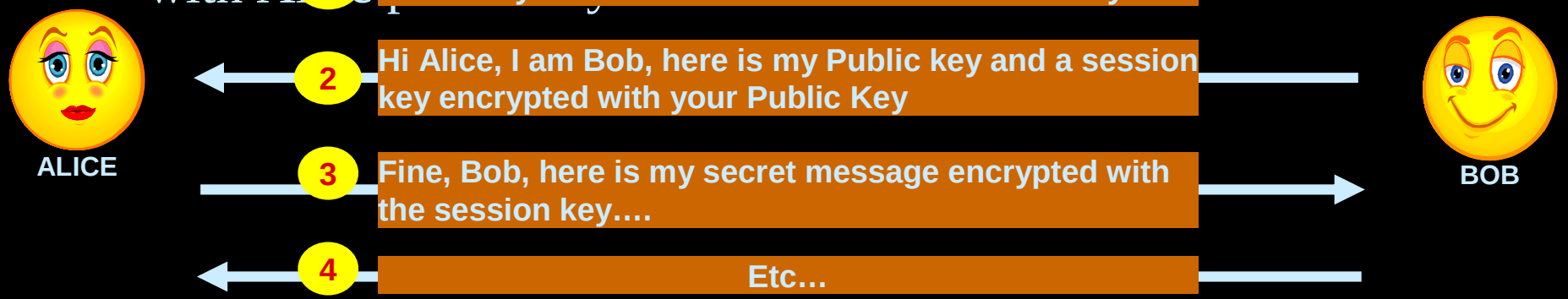
(4) Data alteration + Confidentiality



- Symmetric encryption assumed
- One construction is wrong
- Others have different properties

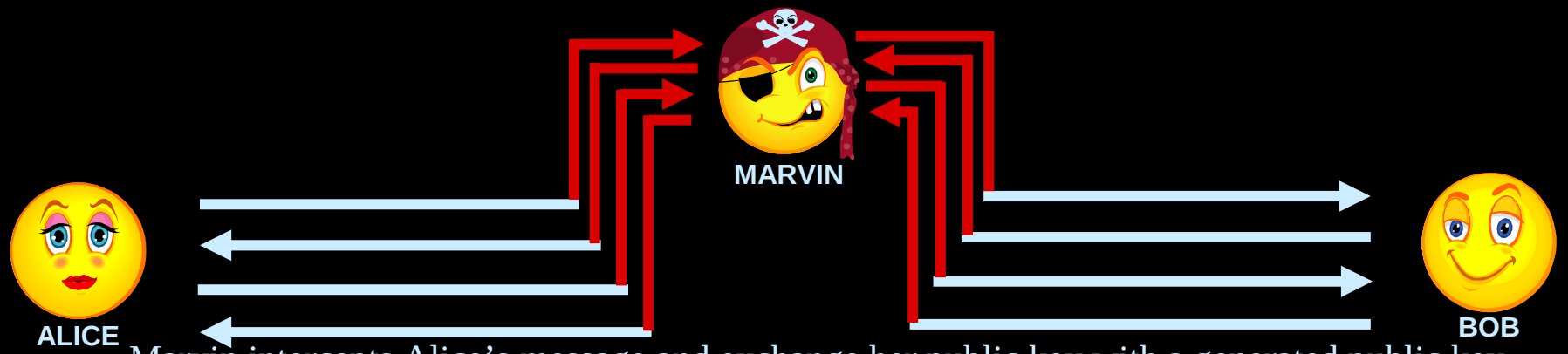
(5) Identity theft: Authentication-1

- Is asymmetric encryption sufficient to ensure authentication ?
  - Alice and Bob have no shared secret but want to establish a secured communication.
  - They exchange their public keys and a session key encrypted with A[illegible]



(5) Authentication-2: attacking the simple protocol

- A “Man in the middle” attack can be conducted !!!



  - Marvin intercepts Alice’s message and exchange her public key with a generated public key (Marvin knows the corresponding private key)
  - Then, he intercepts Bob’s message, decrypt the session key, re-encrypt it with Alice’s public key (kept from 1st message) and exchange Bob’s public key with another generated public key
  - Alice and Bob does not notice anything and Marvin intercepts all messages, can even create fake message !
- A similar attack can be conducted on the Diffie-Helmann protocol
- Need for a trusted party delivering
  - public keys (PKI)
  - or secret keys (e.g., Kerberos)

(5) Authentication-3: trusted party & public keys

- Alice has been previously registered by Trent, showing identity proofs and has sent her public key.
- She received in return, a certificate containing her public key, certified by Trent.
- Trent certificate is "*well known*" (certification hierarchy)

1 – Bobs asks Alice her certificate

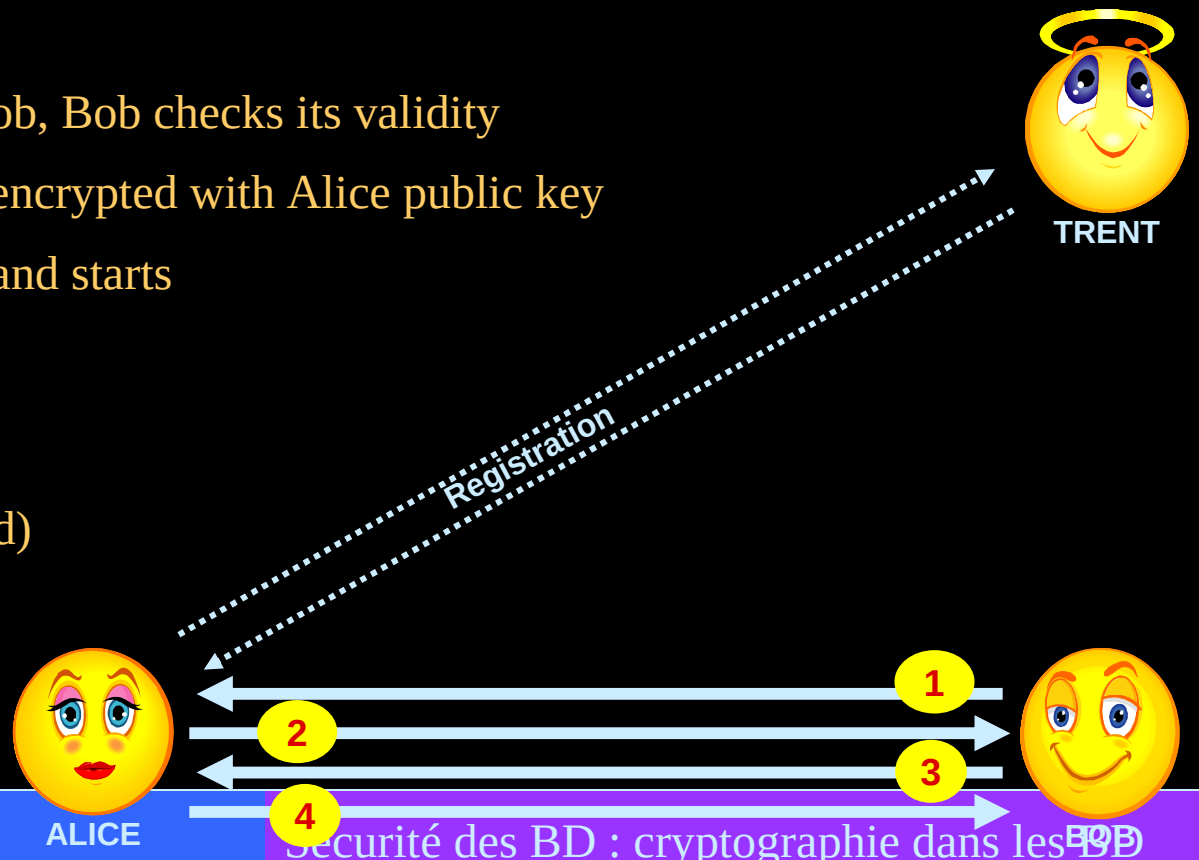
2 – Alice sends her certificate to Bob, Bob checks its validity

3 – Bob sends Alice a session key encrypted with Alice public key

4 – Alice decrypts the session key and starts communicating with Bob

SSL is based on this scheme

(remark: Bob need not be registered)



(5) Authentication-4: trusted party & secret keys

- Alice & Bob have been previously registered by Trent
- During this registration, they exchanged a secret key only known by Trent/Alice (TA) or Trent/Bob (TB)

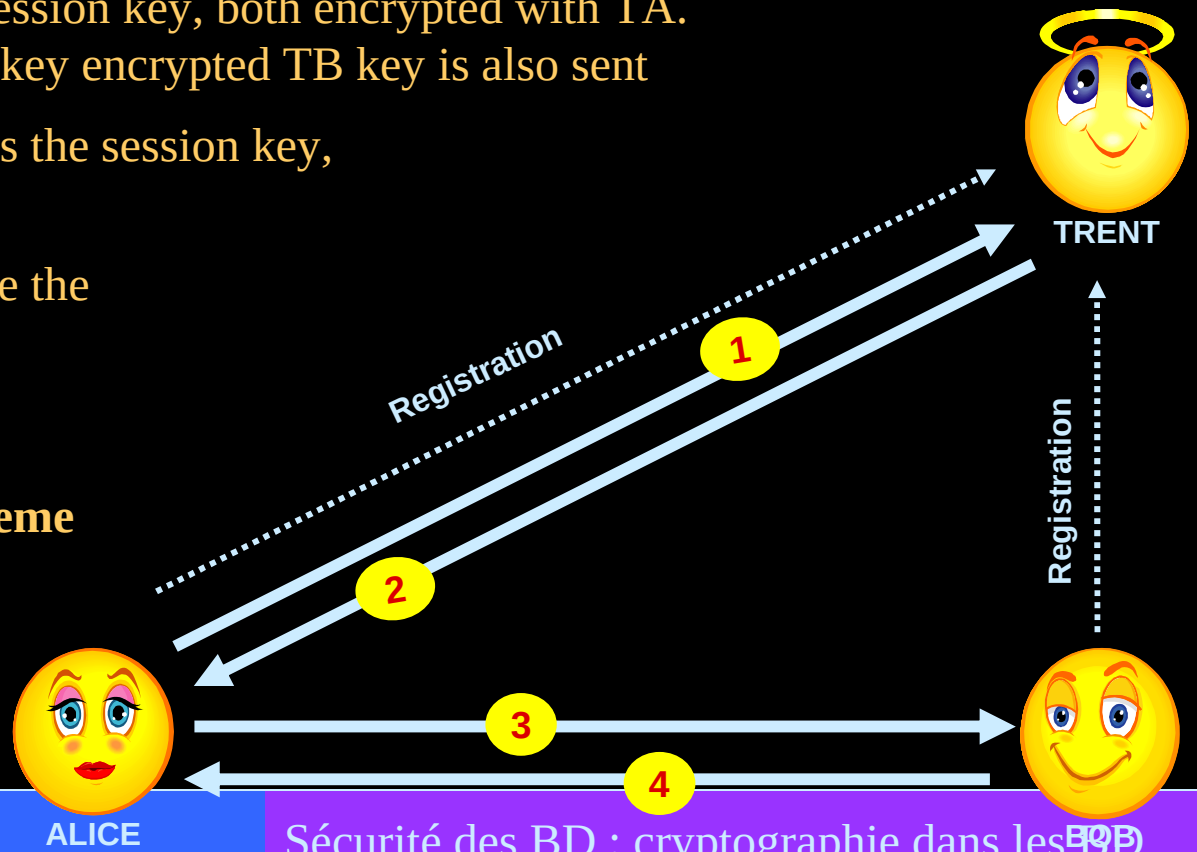
1 - Alice contacts Trent, sending a nonce for a communication with Bob

2 - Trent sends the nonce and the session key, both encrypted with TA.
A ticket, including the session key encrypted TB key is also sent

3 - Alice checks the nonce, decrypts the session key,
sends the Ticket to Bob

4 - Bob decrypts the Ticket, retrieve the
session key, and starts
communicating with Alice

KERBEROS is based on this scheme



(5) Authentication-5: Secret vs Public

- Both need trusted third parties.

- Secret (Kerberos)
  - Distributes symmetric keys
  - Operates on-line, when interaction takes place since it creates a new symmetric key for each session

- Public (SSL, Certification authority)
  - Distributes public keys (certificates)
  - Operates off-line, prior to interaction since public key is fixed
  - Once certificate created, intervention by CA no longer required

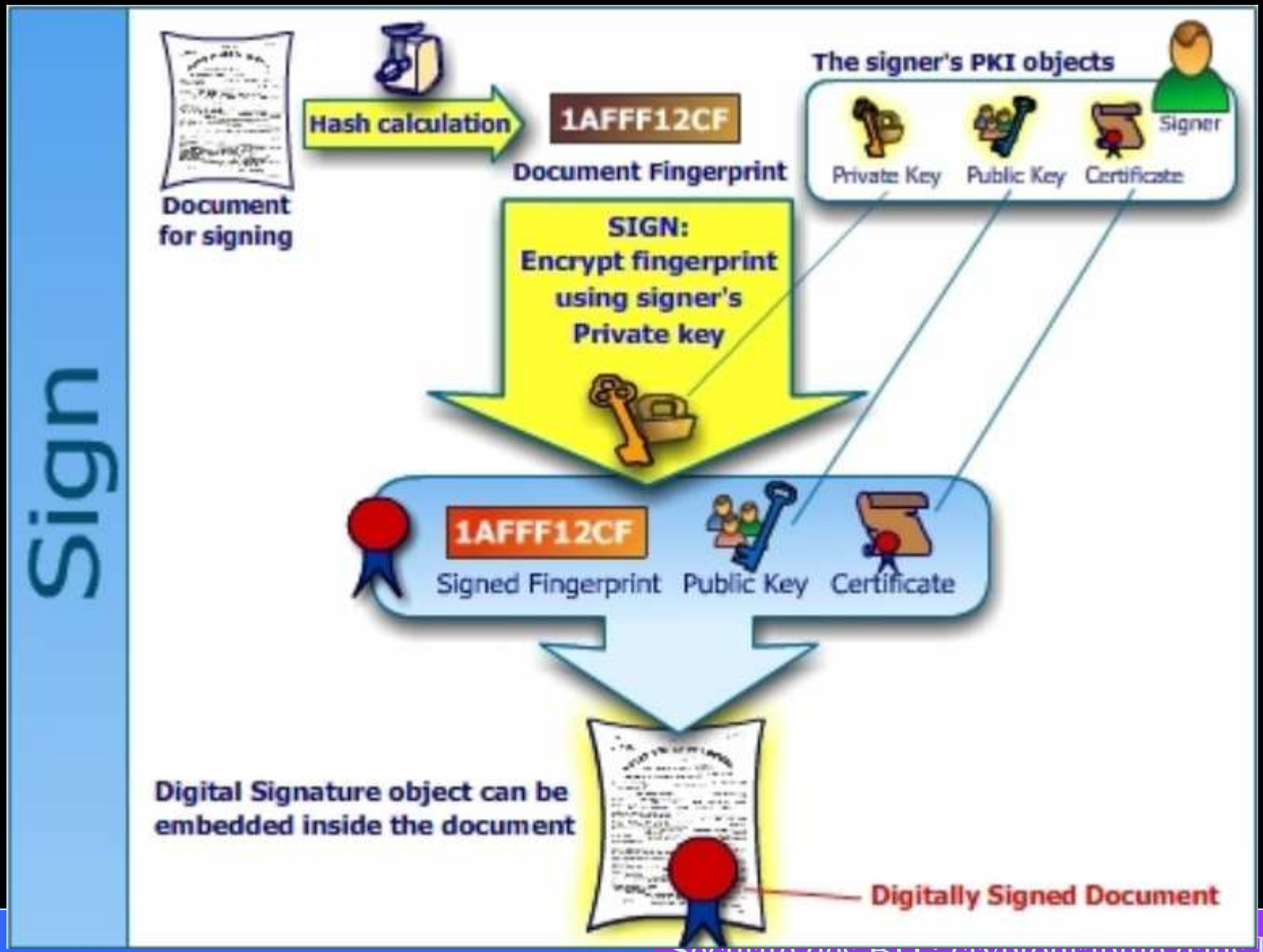
(6) repudiation: Digital Signature-1

- Digital Signatures can be used for
  - Proof of authorship
  - Non-repudiation by author
  - Guarantee of message integrity
  - Does not guarantee confidentiality (orthogonal)
    - If confidentiality is needed, encryption of the message must be used
- Important for many Internet applications
- Based on public key cryptography
  - Current systems use RSA algorithm
- Basic idea:

  (1)Hash the message

  (2)Encrypt the hashing with the sender private key

(6) Digital signature-2: Signing



(6) Digital Signature- 3: Verifying