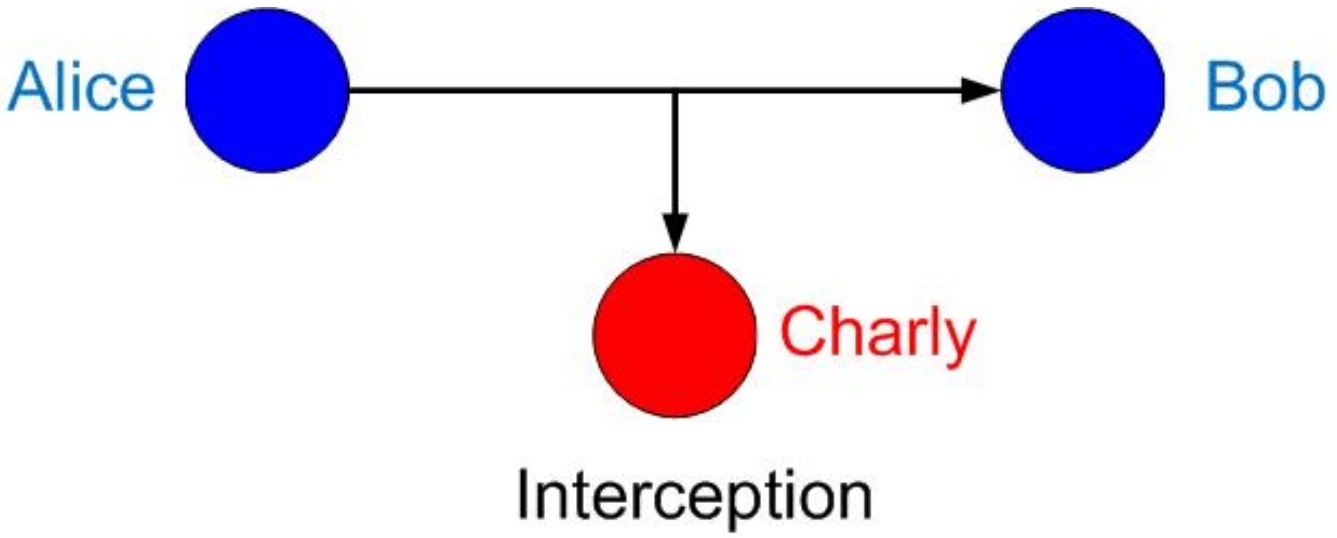
Introduction à la cryptographie

Démo

- Confidentialité, intégrité, authentification, non-répudiation
- Chiffrement à clé symétrique, *brute force attack*
- Chiffrement à clés asymétriques, algorithme RSA
- *Message-Digest*
- Signature numérique
- Projet SwissQuantum
- Travail personnel

Principe de sécurité : confidentialité



- **3 acteurs :** Alice – Bob – Charly (intrus, *hacker*)
- **Risque** = écoute (interception) de données confidentielle par Charly
- Conserver le caractère privé → chiffrement du message
- Seul le destinataire (Bob) peut déchiffrer le message
- Charly devra passer un certain temps pour le décrypter

Chiffrement & cryptographie

- Ecriture secrète : … Mésopotamie, …

- Algorithme de César avec clé = 3 démo Cryptool

  Texte en clair : `attaquer`

  Texte chiffré : `dwwdtxhu`

- Algorithme de Blaise de Vigenère (diplomate)

  utilisé entre 16ème et 19ème siècle Voir slide suivant

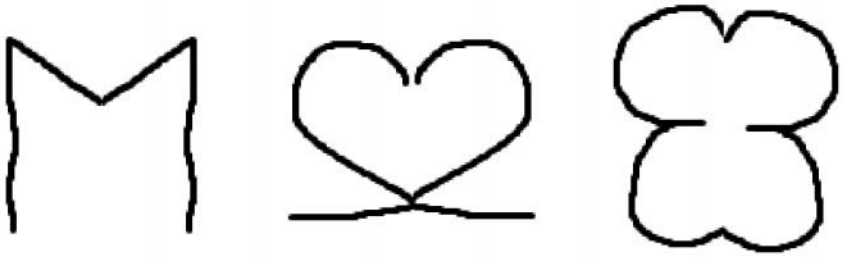
- Enigma développée dès 1919 et utilisée lors de la seconde guerre mondiale par les Allemands démo Cryptool

Algorithme de Blaise de Vigenère

	A	B	C	D	E	...	X	Y	Z
A	a	b	c	d	e		x	y	z
B	b	c	d	e	f		y	z	a
C	c	d	e	f	g		z	a	b
D	d	e	f	g	h		a	b	c
E	e	f	g	h	i		b	c	d

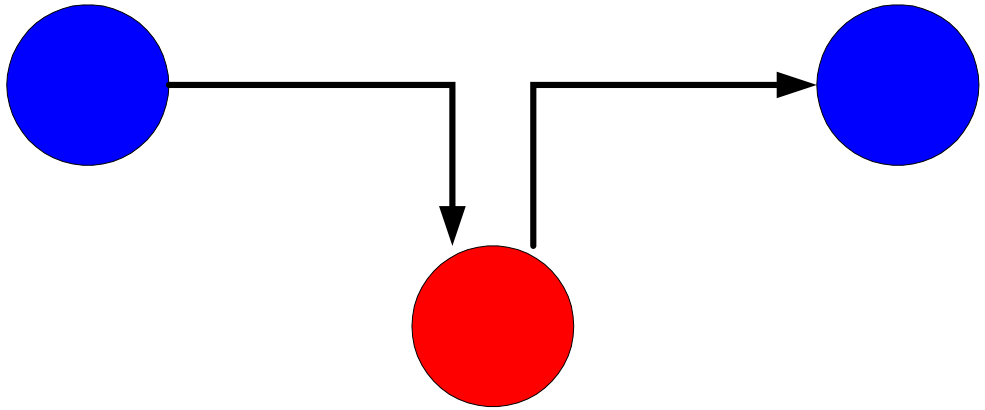
- Clé : **CADRECADRE**
- Texte en clair *(Cleartext)* : **BEAUCOUP**
- Texte chiffré *(Ciphertext)* : **dedlg...**

Classification selon Simon Singh



- Communication secrète
  - Stéganographie
  - Cryptographie
    - Transposition
    - Substitution
      - Chiffrement
        - Mono-alphabétique
        - Poly-alphabétique
        - One Time Pad
      - Codage

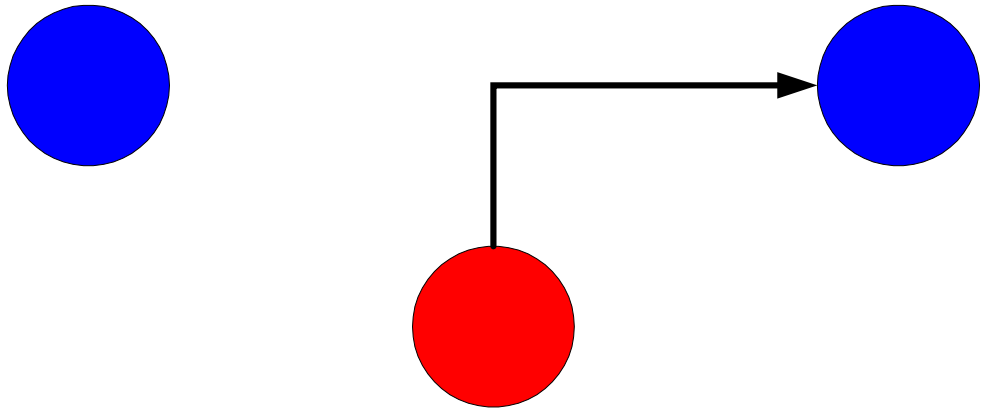
Principe de sécurité : intégrité



Modification

- **Risque** = attaque du type *man in the middle*
- Alice croit communiquer avec Bob mais Charly intercepte et modifie les messages
- Contrôle d'intégrité d'extrémité à extrémité

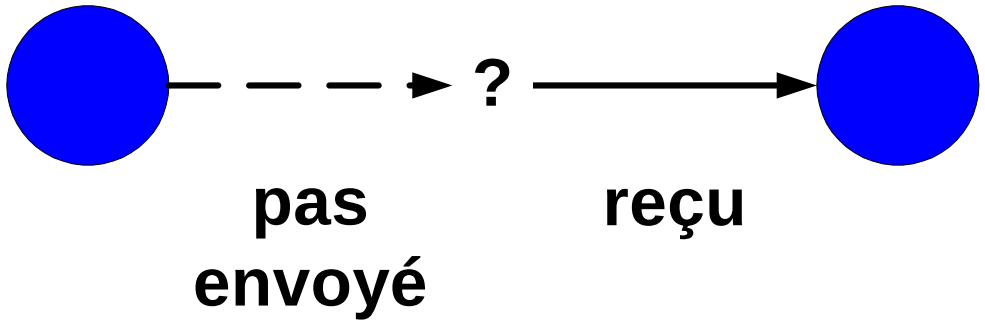
Principe de sécurité : authentification



Imposture

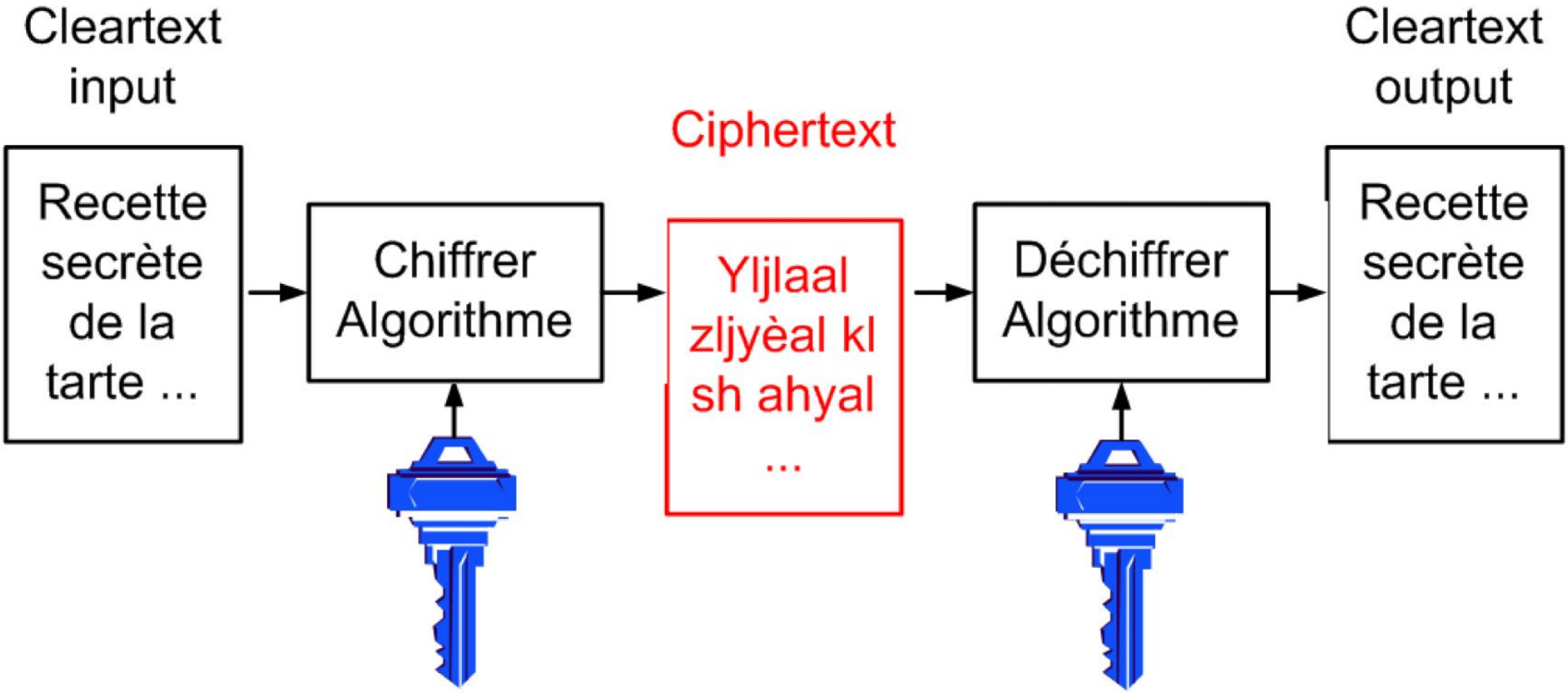
- Identité d’un individu, d’un serveur ou d’une application
- **Risque** (usurpation d’identité) : Charly se fait passer pour Alice

Principe de sécurité : non-répudiation



- Message reçu par Bob alors qu'Alice nie l'avoir envoyé
- Non-répudiation de l'envoi
- Non-répudiation de la réception

Chiffrement à clé symétrique (1)



La même clé est utilisée pour chiffrer & déchiffrer

Chiffrement à clé symétrique (2)

- Principaux algorithmes démo

  DES : *Data Encryption Standard* → clé de 56 bit

  3DES : *TripleDES* → clé de 168 bit (3 chiffrements DES)

  IDEA : *International Data Encryption Algorithm* – clé de 128 bit

  RC4 : *Rivest Cipher 4* → clé de 40 - 128 bit

  AES : *Advanced Encryption Standard* → clé de 128 / 256 bit

  Requête du *National Institute of Standard and Technology (NIST)* pour définir un remplaçant à DES

  Voir http://www.rijndael.com/ (2 math. belges)

- Histoire

  Il fut un temps (récent) où certains pays limitaient la longueur des clés → Export RC4 40 bit

  On parlait alors de chiffrement fort pour des longueurs de clé > 128

Chiffrement à clé symétrique (3)

- Algorithmes de chiffrement en continu *(Stream Cipher)*

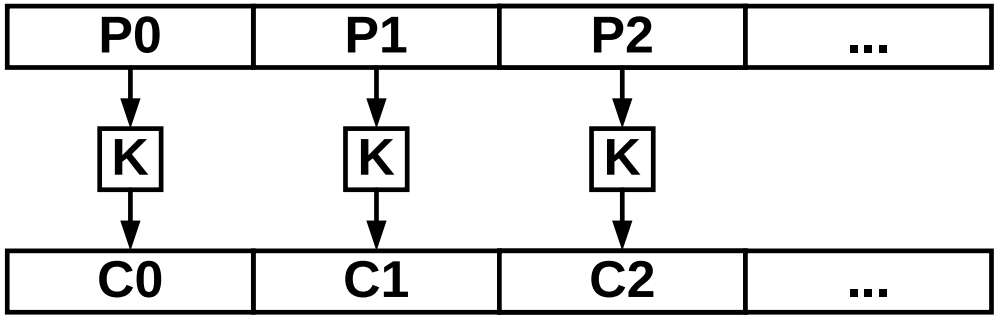
  Chiffrement bit à bit

  Exemple = RC4

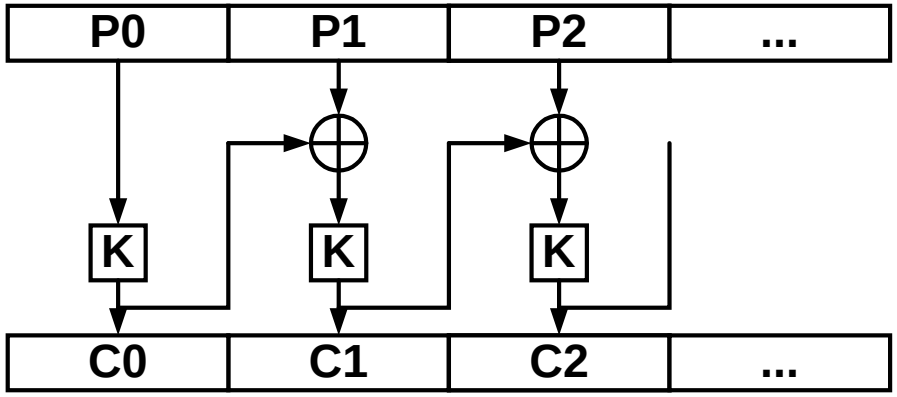
- Algorithmes de chiffrement par blocs *(Block Cipher)*

  Chiffrement par bloc (64 bit → DES, 128 bit → AES, …) du *Cleartext*

ECB=*Electronic Codebook Mode*



CBC=Cipher Block Chaining mode



Autres variantes : CFB, OFB

Chiffrement à clé symétrique : avantages

- Implémentation simple matérielle ou logicielle

- Capable de chiffrer des messages de grandes tailles

- Algorithmes connus (publiés) et sûrs (cryptoanalyse)

- Seul type d'attaque : *brute force attack*

  → Tester toutes les combinaisons

  Un concours, organisé par RSA en 1999, a montré qu'il suffisait d'environ 24 h à un supercalculateur pour casser une phrase secrète chiffrée avec une clé DES de 56 bits

  Seules 25 % des combinaisons ont été testées !

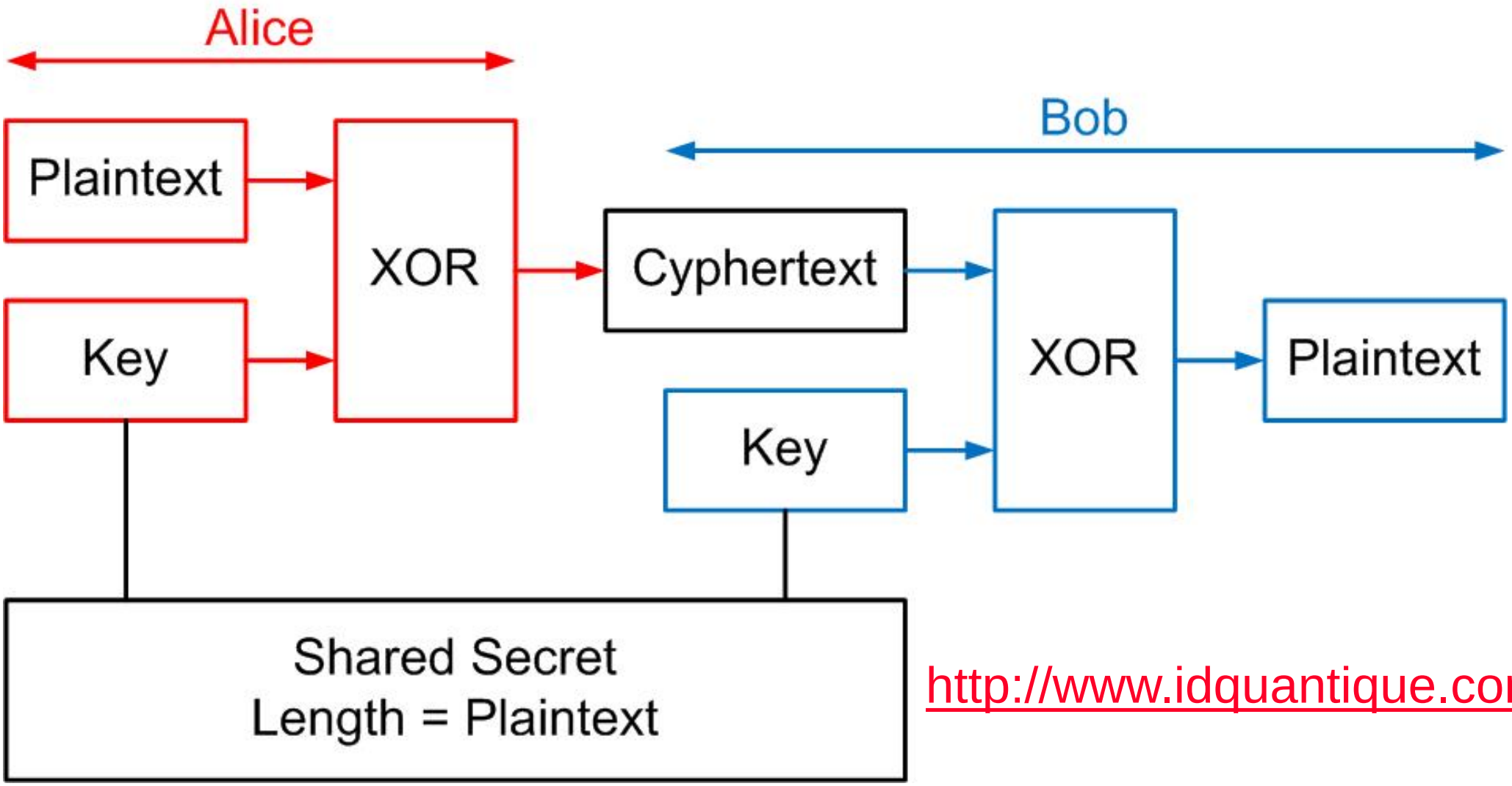
  Voir http://www.tdeig.ch/01_Cours_Supelec.pdf slides 95-96

Brute force attack

Longueur de la clé	Nombre de combinaisons
40 bits	2^{40} = 1’099’511’627’776
56 bits	2^{56} = 7.205759403793 x 10^{16}
128 bits	2^{128} = 3.402823669209 x 10^{38}

- Si l’ennemi pouvait générer 2^{56} combinaisons en 1 seconde; il aurait besoin de 149 mille milliard d’années pour 2^{128} (AES)

Confidentialité parfaite



http://www.idquantique.com/

- Masque jetable *(One Time Pad)*

Chiffrement à clé symétrique : inconvénients

- Difficulté de partager un secret
- Nombre de clés secrètes différentes que chaque particulier doit gérer; généralement une par partenaire
- Pas de non-répudiation
- La clé peut être interceptée

Types d'attaque

- Charly ne dispose que des documents chiffrés

  → Trouver la clé ou le *cleartext* à partir du *ciphertext*

  → *Ciphertext only*

- Charly possède la boîte noire (Enigma, …) et peut ainsi

  → Trouver la clé à partir de couples disponibles

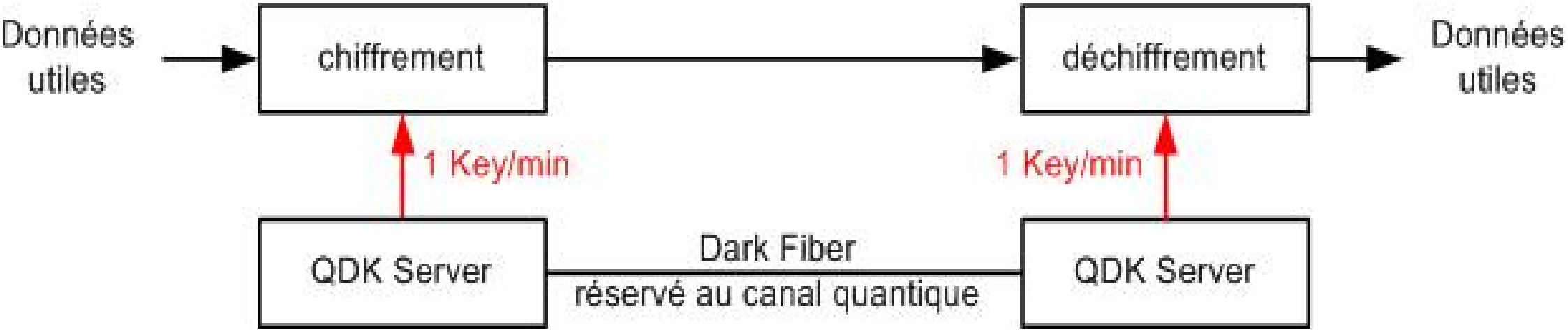
  → *Known cleartext – ciphertext*

  → Trouver la clé à partir de *cleartext* spécifique

  → *Chosen cleartext*

Projet SwissQuantum : principes

- La technologie développée par l'équipe du prof. Gisin (UniGE) et commercialisée par la société id Quantique dirigée par le Dr Grégoire Ribordy, garantit la confidentialité des données échangées en exploitant les lois de la physique quantique



- L'écoute du canal quantique (Charly) rend le canal de données inutilisable
- La durée de vie des clés symétriques peut être courte et leur longueur importante *(One Time Pad)*

Projet SwissQuantum : Technologie

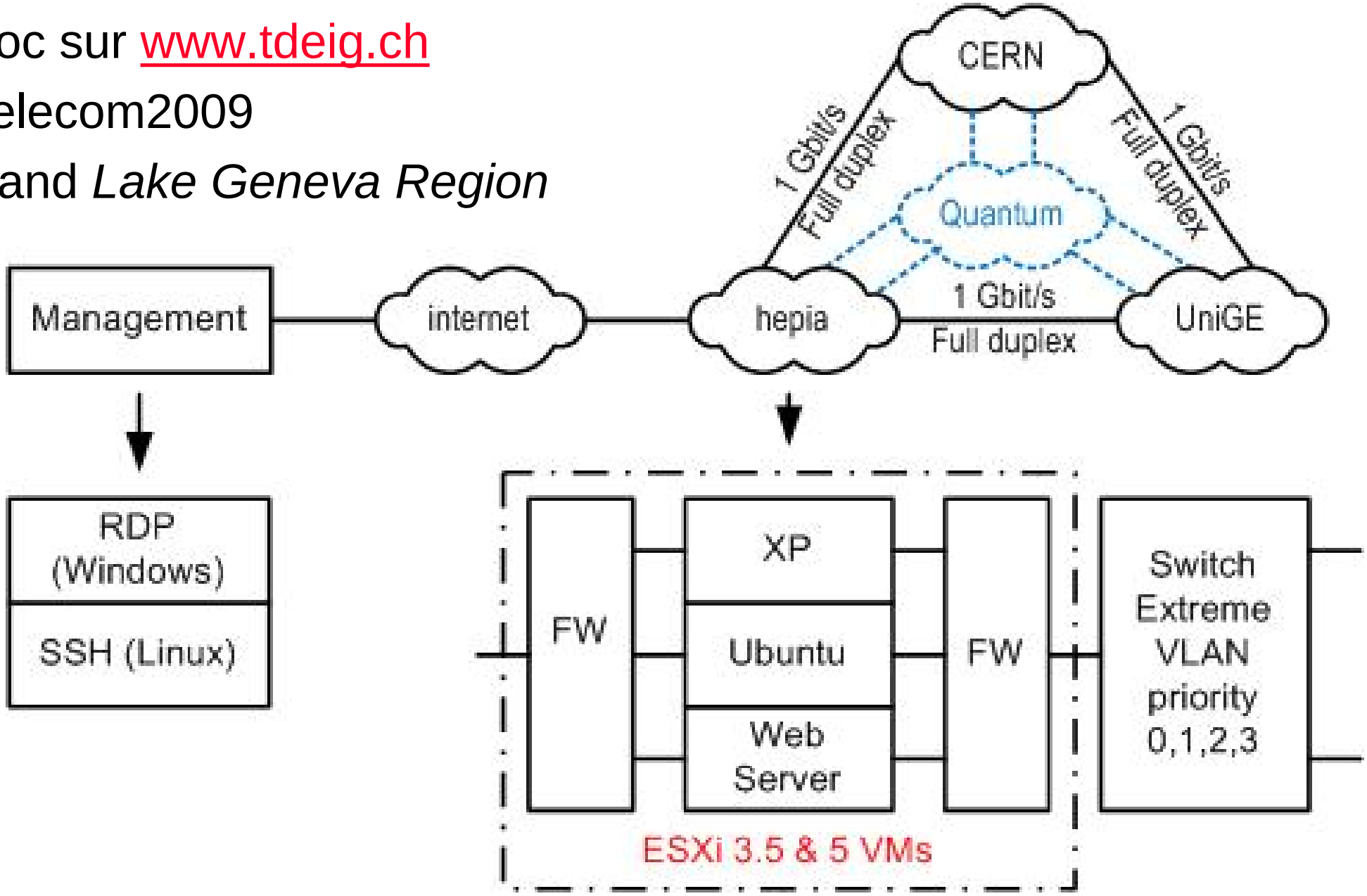
- Modules de chiffrement

- QKD *(Quantum Distribution Key)* server

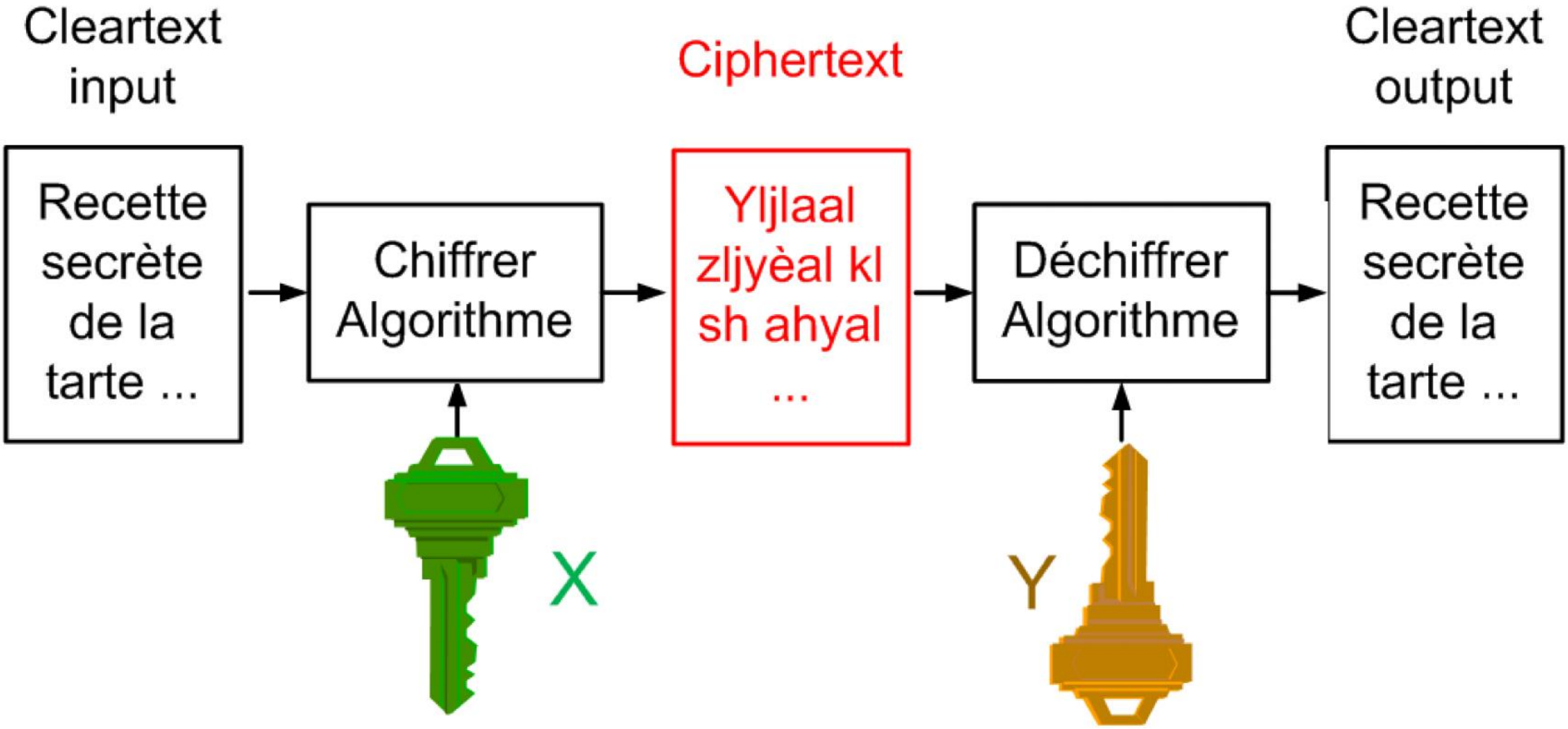
Projet SwissQuantum (cahier des charges)

- Doc sur www.tdeig.ch
- Telecom2009

  stand *Lake Geneva Region*



Chiffrement à clés asymétriques (1)

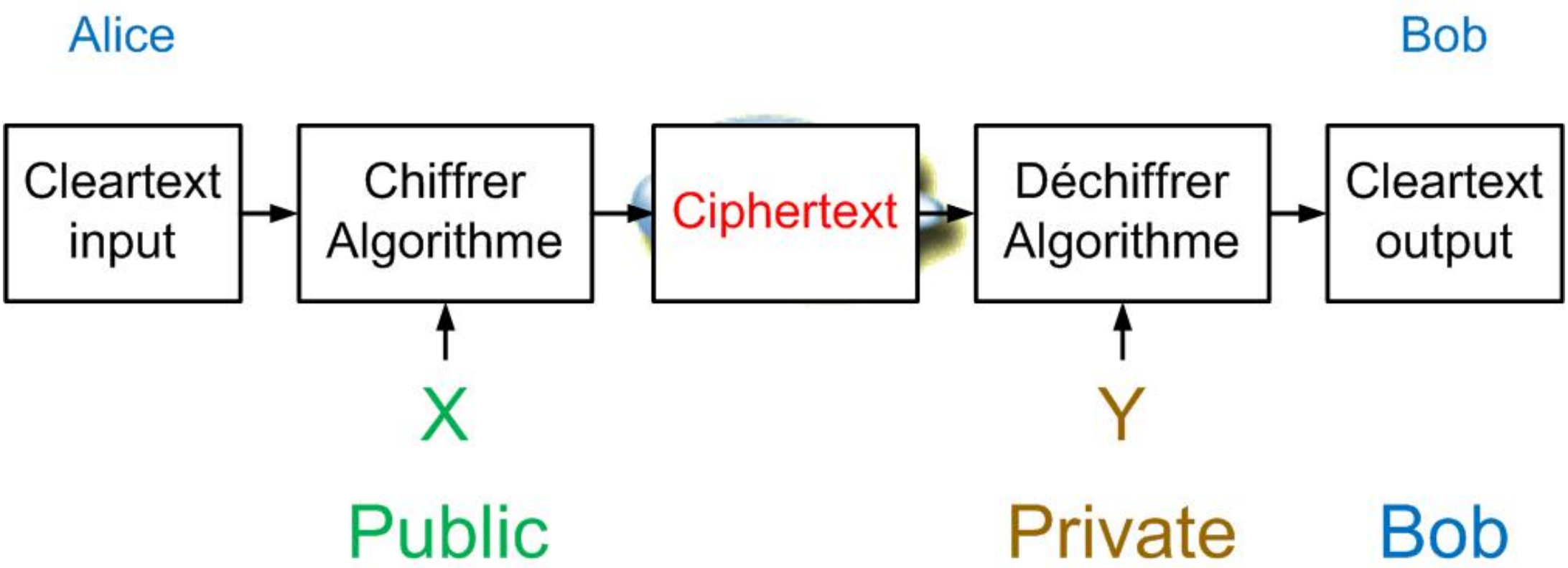


Ce qui est chiffré avec la clé X

ne peut être déchiffré qu'avec la clé Y

Chiffrement à clés asymétriques (2)

- Un couple de clés **X** et **Y** appairées est généré
- **X** ne peut être déduit de **Y** et réciproquement
- Alice possède la clé **X (publique)** et Bob utilise sa clé **Y (privée)**



Distribution des clés

- L'exemple précédent illustre le début d'un échange sécurisé https, où le serveur (**Bob**) transmet sa clé publique à **Alice**.

- **Bob**, ayant généré un couple de clés **(clé privée + clé publique)**, **doit en transmettre une à Alice**

- Comment Alice peut-elle être certaine que la clé publique reçue est bien celle de Bob ?

  → certificat numérique

- Dans l'exemple, Alice ne possède pas de clé privée

Chiffrement à clés asymétriques : avantages

- Pas de secret partagé
- Une paire de clé par partenaire
- Pas d’accord préalable
- Facile à administrer ?
- Non-répudiation supportée

Chiffrement à clés asymétriques : inconvénients

- Lent pour chiffrer des messages importants
- Les clés doivent être plus longues que celles utilisées avec les systèmes symétriques
- Impossible de chiffrer un message dont la taille dépasse la longueur de la clé

→ **Intérêt pour les systèmes hybrides utilisés par SSL, …**

Clé publique : RSA

- RSA → Rivest, Shamir & Adleman (inventeurs en 1977)
  Brevet a expiré le 20 sept 2000 → domaine public

- Longueur max de la clé de 4096
  Plus de limitation pour l'exportation

- Longueur du message < longueur de la clé

- Fondé sur la décomposition en nombres premiers de nombres très grands

Fonctionnement de RSA

1 Choisir aléatoirement 2 "grands" nb **premiers** p et q — p=7 q=13

2 Calculer n = p*q — n = 7*13 = 91

(p-1)(q-1) — 6*12 = 72

3 Choisir e tel que $1 < e < (p-1)(q-1)$ — e = 5

pgcd (72,e) = 1

→ clé publique <e,n> — <5,91>

4 Calculer $d = e^{-1} \bmod (p-1)(q-1)$ ou $d*e=1 \bmod (p-1)(q-1)$

→ d=(X*(p-1)(q-1) +1)/e essayez X =[1…n] afin que d soit un entier

d= (2* (p-1)(q-1) +1)/e — d= 29

→ clé privée <d,n> — <29,91>

5 Supprimer p, q, (p-1)(q-1)

Sécurité de RSA (1)

- Repose sur la difficulté de factoriser un nombre grand
- Nombre de 1024 bit comprend 308 chiffres → RSA-308
- Factorisation de RSA-200 annoncée le 9 mai 2005

2799783391 1221327870 8294676387 2260162107 0446786955
4285375600 0992932612 8400107609 3456710529 5536085606
1822351910 9513657886 3710595448 2006576775 0985805576
1357909873 4950144178 8631789462 9518723786 9221823983

=

3532461934 4027701212 7260497819 8464368671 1974001976
2502364930 3468776121 2536794232 0005854795 6528088349

x

7925869954 4783330333 4708584148 0059687737 9758573642
1996073433 0341455767 8728181521 3538140930 4740185467

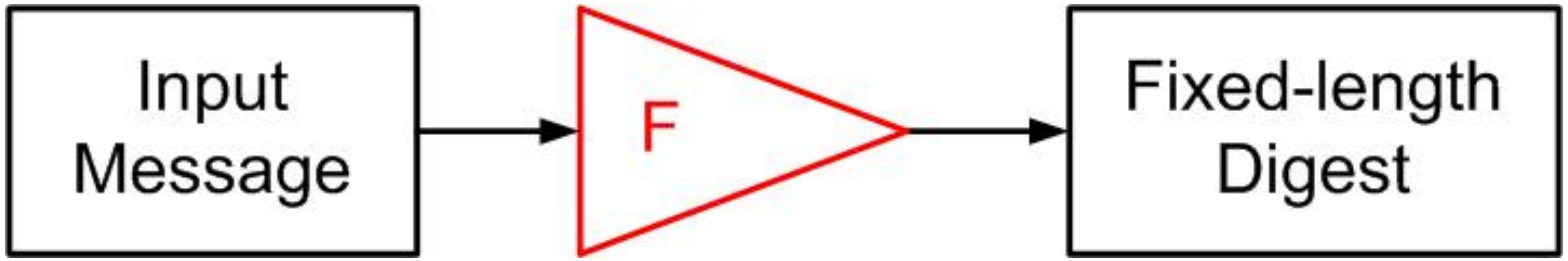
Sécurité de RSA (2)

- Les groupes informatiques de trois institutions – l'EPFL, l'Université de Bonn et la compagnie NTT au Japon – ont extrait, après onze mois de calculs, les facteurs premiers d'un nombre qui totalise 307 chiffres

- "C'est le plus grand nombre de cette forme que l'on factorise à ce jour," explique le professeur de cryptologie à l'EPFL Arjen Lenstra. Il a en effet une forme mathématique spéciale – il est proche d'une puissance de deux. Le nouveau nombre à 307 chiffres est tout prêt des 308 du fameux cryptage standard RSA de 1024 bits, …

- http://actualites.epfl.ch/presseinfo-com?id=439 mai 2007

Message-Digest (1)



- Message de longueur variable
- Fonction de hachage unidirectionnelle *(Hash Function)* qui produit un condensé de longueur constante (empreinte)
- *Fixed-length Digest = message digest = digest = hash*
- MD5 (128 bit), SHA-1 (160 bit), SHA-256, SHA-384, … Démo

Message-Digest (2)

Pour être sûre, la fonction de hachage doit :

- Empêcher de déterminer le message x à partir du condensé y

  Impossible, pour un y choisi, de trouver x tel que y = f(x)

  Fonction unidirectionnelle x → y

- Empêcher de trouver un message arbitraire qui donne un condensé particulier

  Soit x et y = f(x), impossible de trouver x' tel que f(x') = f(x)

  Collision = 2 messages (x et x') qui donnent le même condensé

  http://www.mscs.dal.ca/~selinger/md5collision/

- Etre très sensible aux changements du message

  démo : Indiv Proc – Hash – Hash demo

Illustration avec MD5

Message	Condensé (16 bytes)
Hello world…	**`8da0e4b9d4cddad19bfbaf3160ad6596`**
hello world…	**`c95f105ab2434587045d9cf1e79ee9ef`**

Un condensé MD5 (128 bit) peut prendre 2^{128} valeurs différentes
= 3.402823669209 x 10^{38} → collisions possibles

Message-Digest : applications

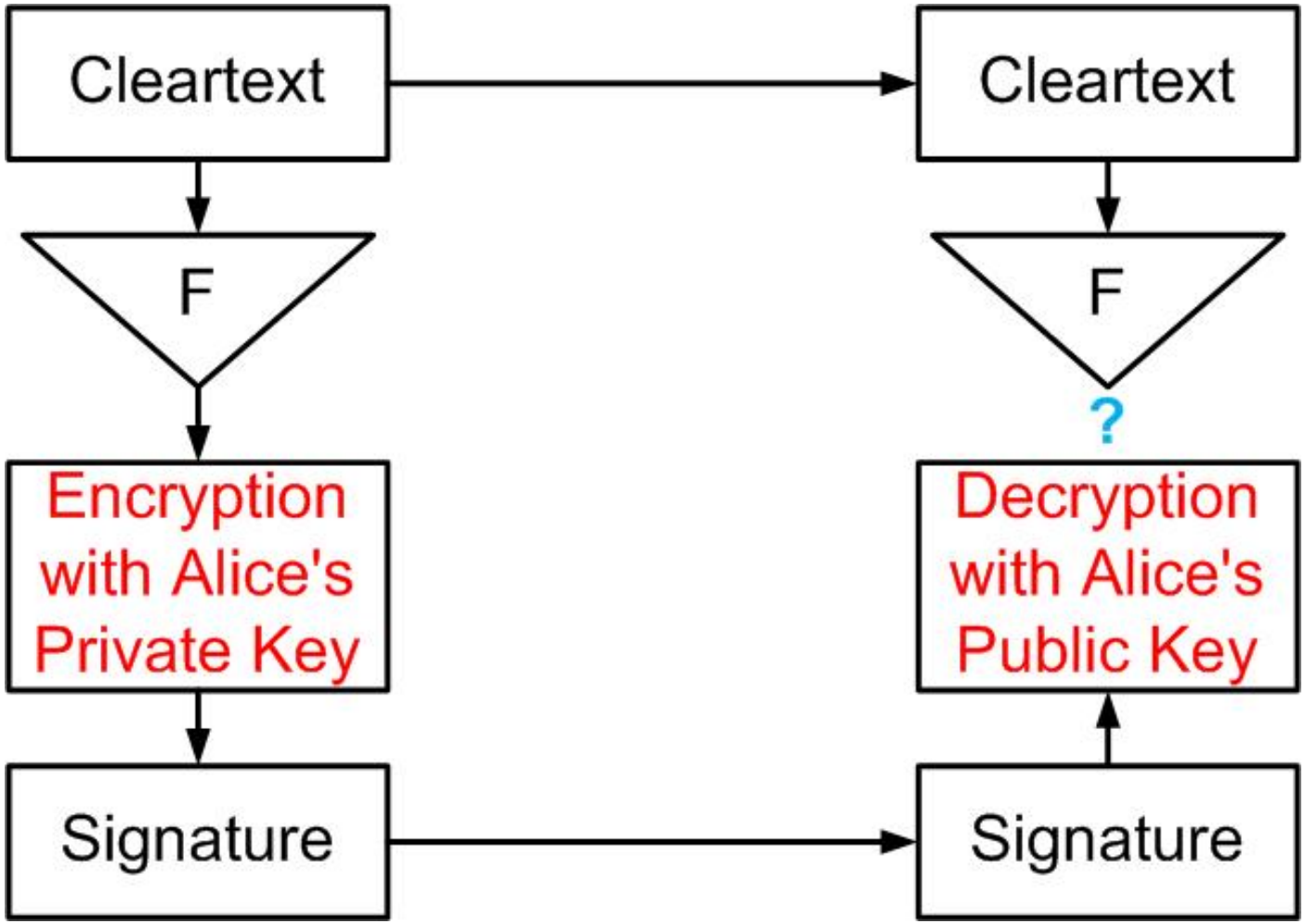
- Condensé du mot de passe est stocké
  → chap. Authentification

- Produire une clé secrète à partir d'un mot de passe *(passphrase)*

- Contrôle d'intégrité

- Signature numérique

Signature numérique (1)



Signature numérique (2)

- L'exemple précédent illustre l'envoi d'un *email* signé
- Alice calcule la signature numérique du message = un condensé du message *(cleartext)* à envoyer qu'elle chiffre avec sa clé privée
- Bob déchiffre la signature reçue avec la clé publique d'Alice

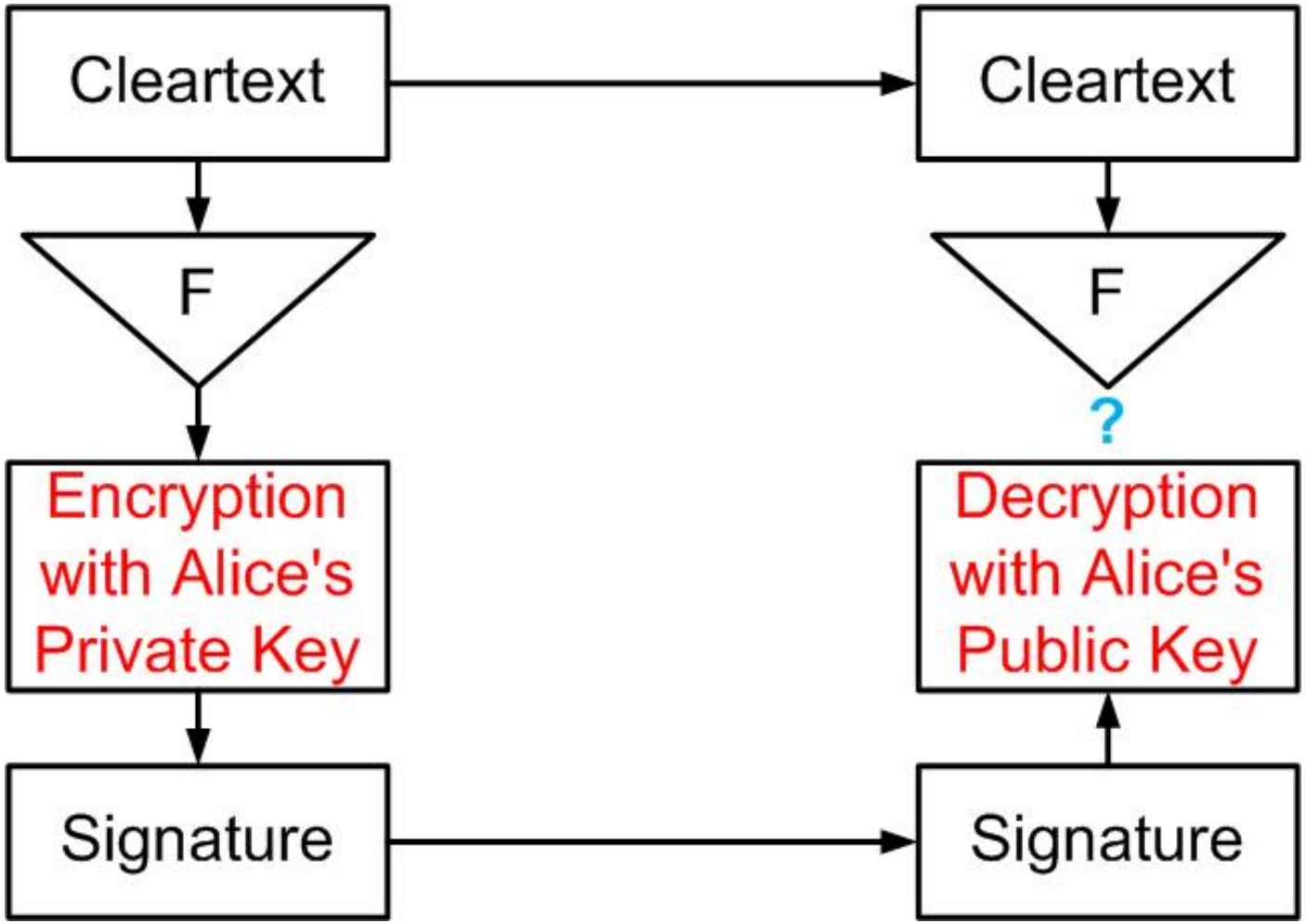
  pour comparer ce résultat avec le calcul du condensé effectué à partir du message
- **En cas d'égalité, que peut affirmer Bob ?**

Signature numérique est sûre

- **Authenticité de l'émetteur du message**
- **Intégrité du message envoyé**
- **Non-répudiation de l'émission du message**
- **Le message est transmis en clair !**

Exercice – Solution → slide 43

- Modifier la figure de la signature pour ajouter la confidentialité au message



Message signé & chiffré

- confidentialité : seul Bob est capable de lire le message car il possède la bonne clé
- intégrité : le message reçu par Bob correspond exactement à l'original (comparaison des empreintes)
- authenticité : Bob peut contrôler que le message vient d'Alice; et non de Charly
- non-répudiation : Alice ne peut pas nier avoir émis ce message à Bob

En résumé

- **Clé privée pour signer & déchiffrer**
- **Clé publique pour vérifier & chiffrer**
- **Ne jamais transmettre sa clé privée**
- **La mémoriser sur une carte à puce (USB, …)**
- **Effectuer une sauvegarde (certificat + clé privée)**

Dual Key

2 paires de clés sont parfois utilisées pour :

- Chiffrer → *backup, keyrecovery*

  La paire dédiée au chiffrement est sauvegardée afin de permettre le déchiffrement des documents alors que l'employé a quitté l'entreprise par exemple

- Signer

  Pour pouvoir garantir la non répudiation, la clé privée nécessaire à la signature doit rester unique !

Travail personnel

- Etudier à partir des slides 1-39, 61-77, 143-145 de
  http://www.tdeig.ch/01_Cours_Supelec.pdf Local

  Stéganographie

  Confusion et Diffusion

  Transposition et Substitution

  Chiffrement parfait, *One-time-pad*

  Chiffrement par flot, par bloc, RC4, ECB, CBC

  Fonctions de hachage

- Utiliser l'excellent logiciel CrypTool pour étudier la cryptographie
  http://www.cryptool.org/

Outil, livres & lien

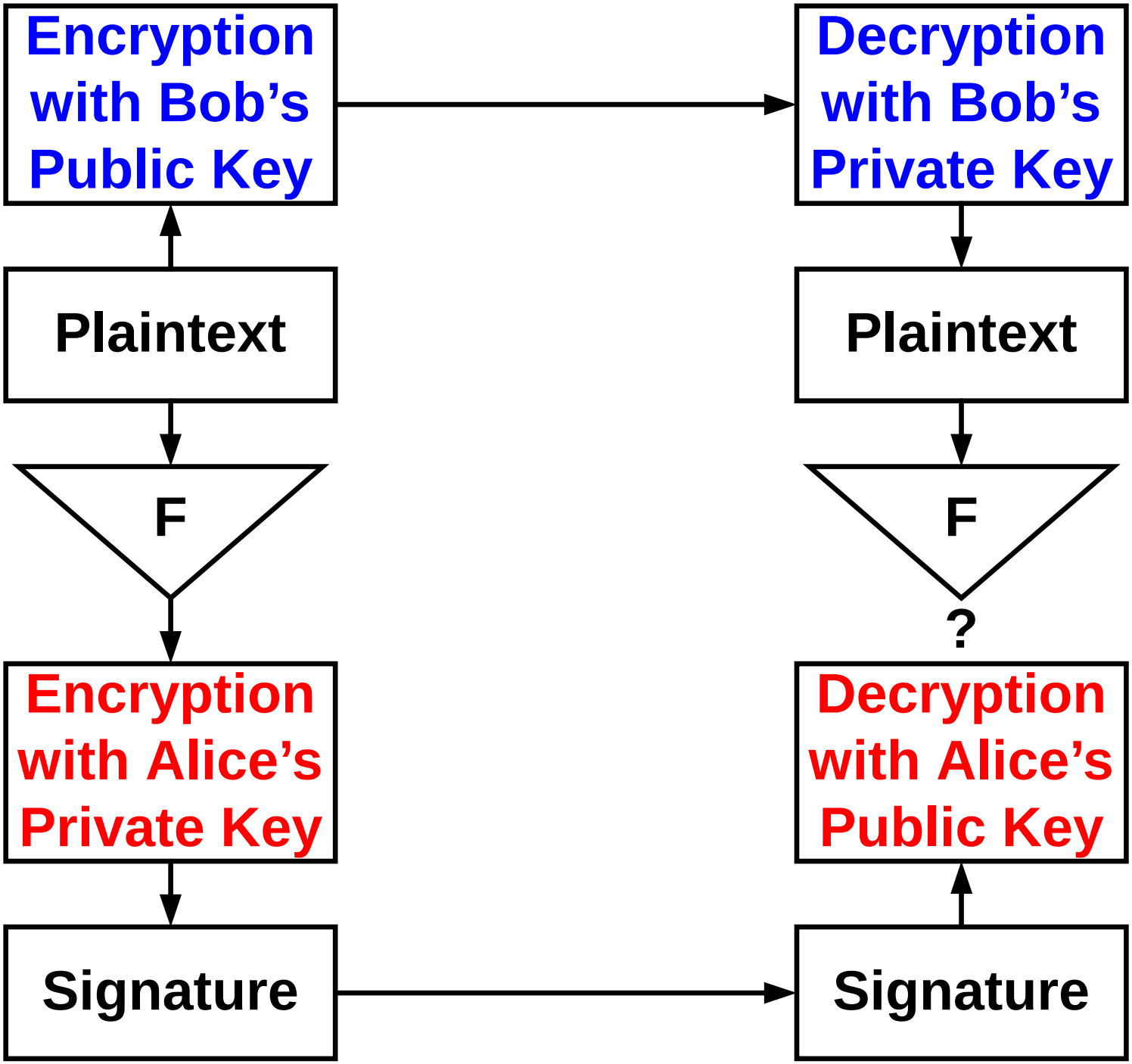
- Boîte à outils openssl (ligne de cmd, librairies, SSL, …)
  www.openssl.org

- Cryptographie : théorie et pratique — Stinson
  ISBN 2711786757 — traduit par Vaudenay (EPFL)

- Cryptographie appliquée — Bruce Schneier
  ISBN 2-84180-036-9 — traduction de Laurent Viennot

- *Handbook of Applied Cryptography* — *Menezes, van Oorschot, Vanstone*
  www.cacr.math.uwaterloo.ca/hac

Solution


Encryption with Bob's Public Key
Decryption with Bob's Private Key
Plaintext
Plaintext
F
F
?
Encryption with Alice's Private Key
Decryption with Alice's Public Key
Signature
Signature