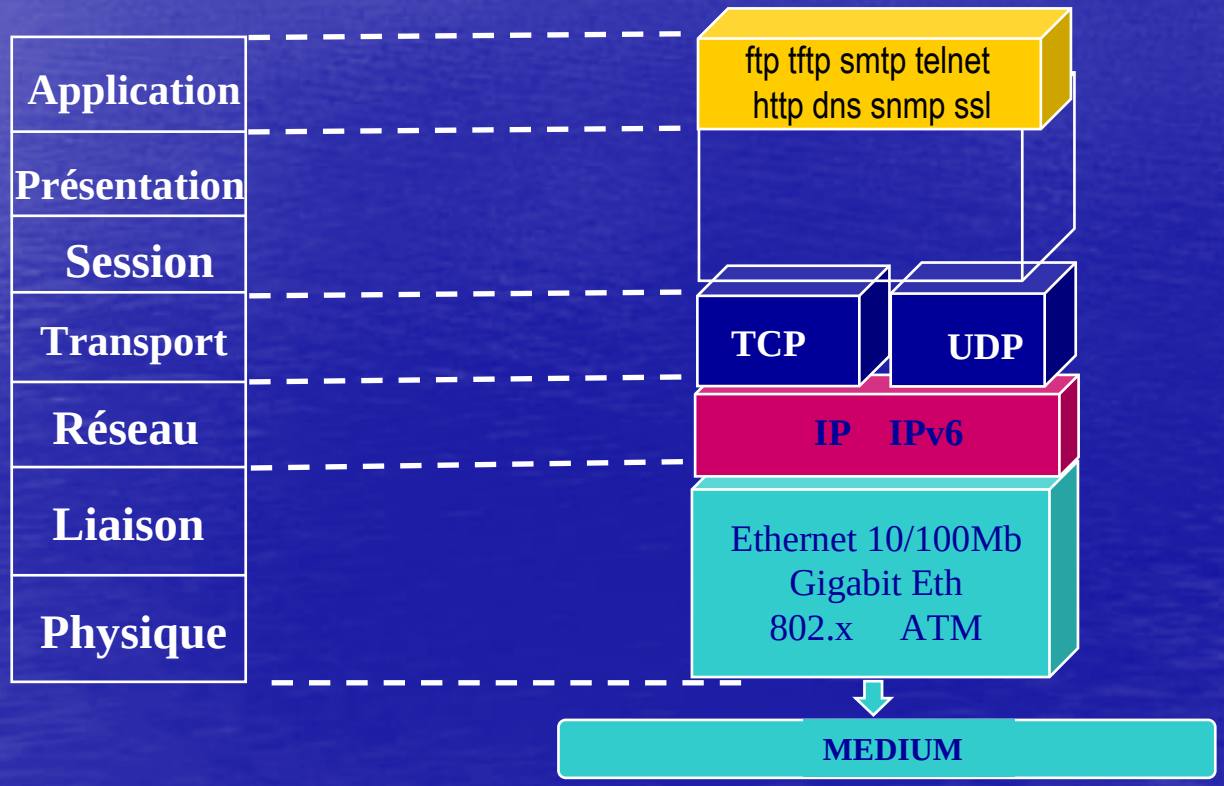
Sécurité des services

- Les services TCP/IP
- Le protocole SSL/TLS
- Modèle d'interaction SSL
- Ssh et redirection
- Le service HTTPS
- Kerberos et authentification

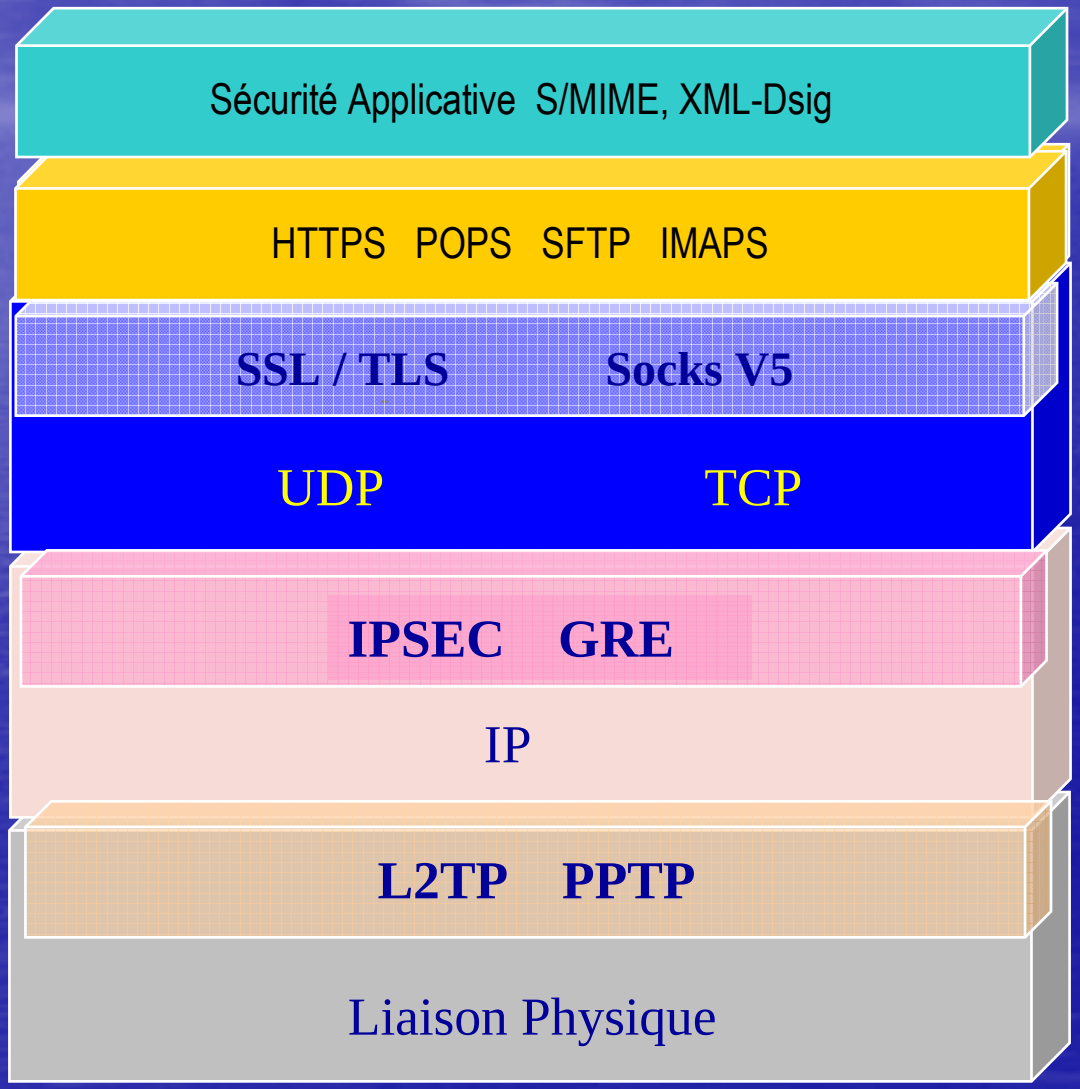
Services et ports

le problème

- Dans le modèle TCP/IP, les services réseaux sont rendus en s'appuyant sur les protocoles transport TCP ou UDP, et ne bénéficient d'aucune sécurité



Modèle OSI et Sécurité



Services et ports

la solution

Utiliser SSL : Secure Socket Layer

- Sécurité des communications au niveau Transport
- Offre un service sécurisé au niveau Transport de manière indépendante
- Chaque service rendu est identifié par un numéro de port (ex : http 80)
- Chaque service sera sécurisé en utilisant un autre port (ex : https 443)
- Les communications bénéficient d'authentification, d'intégrité et de confidentialité

SSL Historique

- Créé et développé par Netscape
  - SSL v1.0 sans implémentation
  - SSL v2.0 a été implémenté pour sécuriser le navigateur Netscape Communicator (1994)
  - SSL v3.0 a été proposée à l'IETF sous forme d'Internet Draft (1996)
- Standardisé par l'IETF sous TLS
  Transport Layer Security
  - TLS 1.0 (RFC2248 1999)
    - Analogies avec SSL v3.1
    - Mais non inter-opérable avec SSL

TLS et l'IETF

- RFC 3546 - **Transport Layer Security (TLS) Extensions**
- RFC 3268 - **Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)**
- RFC 2818 - **HTTP Over TLS**
- RFC 2817 - **Upgrading to TLS Within HTTP/1.1**
- RFC 2716 - **PPP EAP TLS Authentication Protocol**
- RFC 2712 - **Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)**
- RFC 2595 - **Using TLS with IMAP, POP3 and ACAP**
- RFC 2487 - **SMTP Service Extension for Secure SMTP over TLS**
- RFC 2246 - **The TLS Protocol Version 1.0 (ext RFC 3546)**
- **...**

Sécurité SSL

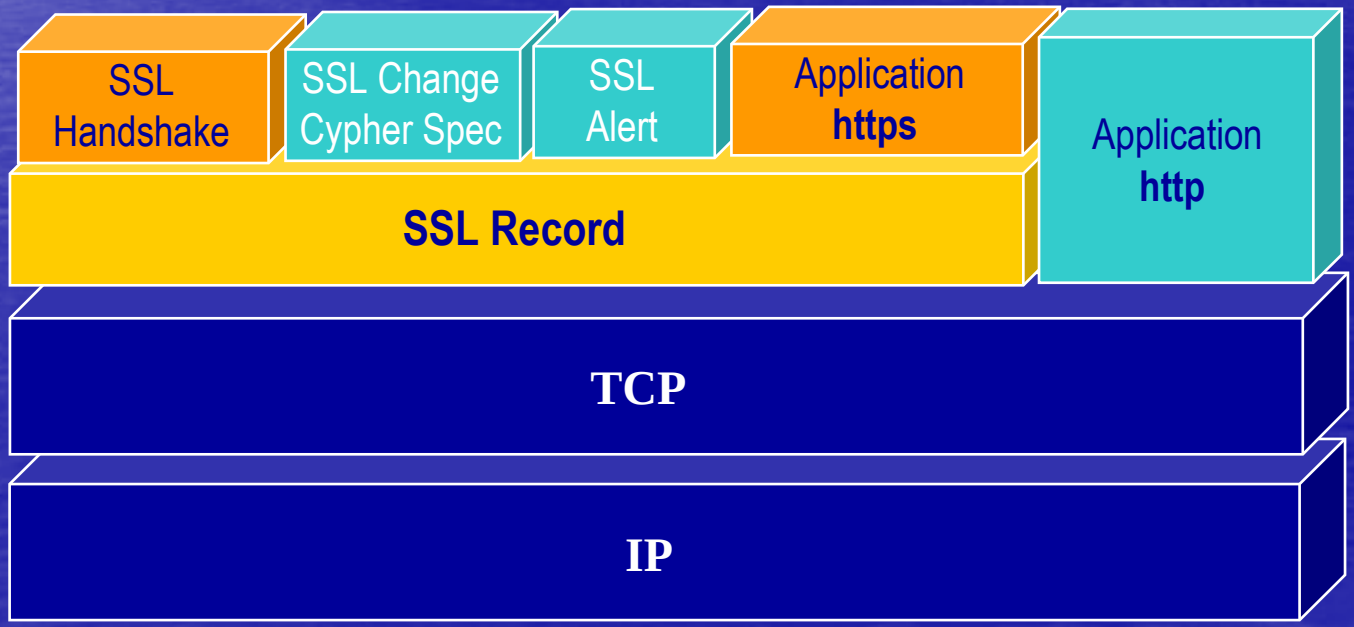
- Authentification avec certificats X.509
  - unidirectionnelle dans le cas de l'authentification du serveur
  - bidirectionnelle dans le cas de l'authentification optionnelle du client
- Intégrité avec condensé
  - MAC (Message Authentification Code)
- Confidentialité
  - Chiffrement des communications par une clé secrète de session
  - Échange de la clé secrète sécurisée par le certificat serveur

SSL : les protocoles

- SSL Handshake
  - Négociations des paramètres et algorithmes
  - Authentification du serveur (opt. du client)
  - Échange des clés
- SSL Change Cypher Spec
  - Modification des paramètres, négociation des algorithmes
- SSL Alert
  - Messages d'erreurs
- SSL Record
  - Traitement des flux : fragmentation, compression
  - Vérification intégrité
  - Chiffrement

SSL : les protocoles

- SSL Handshake
- SSL Record
- SSL Change Cypher Spec
- SSL Alert



SSL Handshake

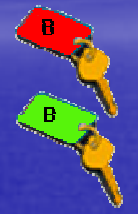
- Une session SSL est établie entre un client et un serveur
- Une session peut maintenir les informations d'état de plusieurs connexions SLL
- Informations d'état d'une session
  - Identificateur de session
  - Certificat X509 du correspondant (opt.)
  - Méthode de compression
  - Algorithmes
    - Chiffrement (DES, 3DES, …)
    - Condensé (MD5, SHA-1, …)
  - Secret ou clé de session
  - État des connexions

SSL : modèle d'interaction

- Les services en mode client/serveur peuvent être sécurisés par le protocole SSL
- Le modèle d'interaction de base permet une authentification serveur

1. Le serveur obtient un certificat et l'installe (clé privée sécurisée sur le serveur)

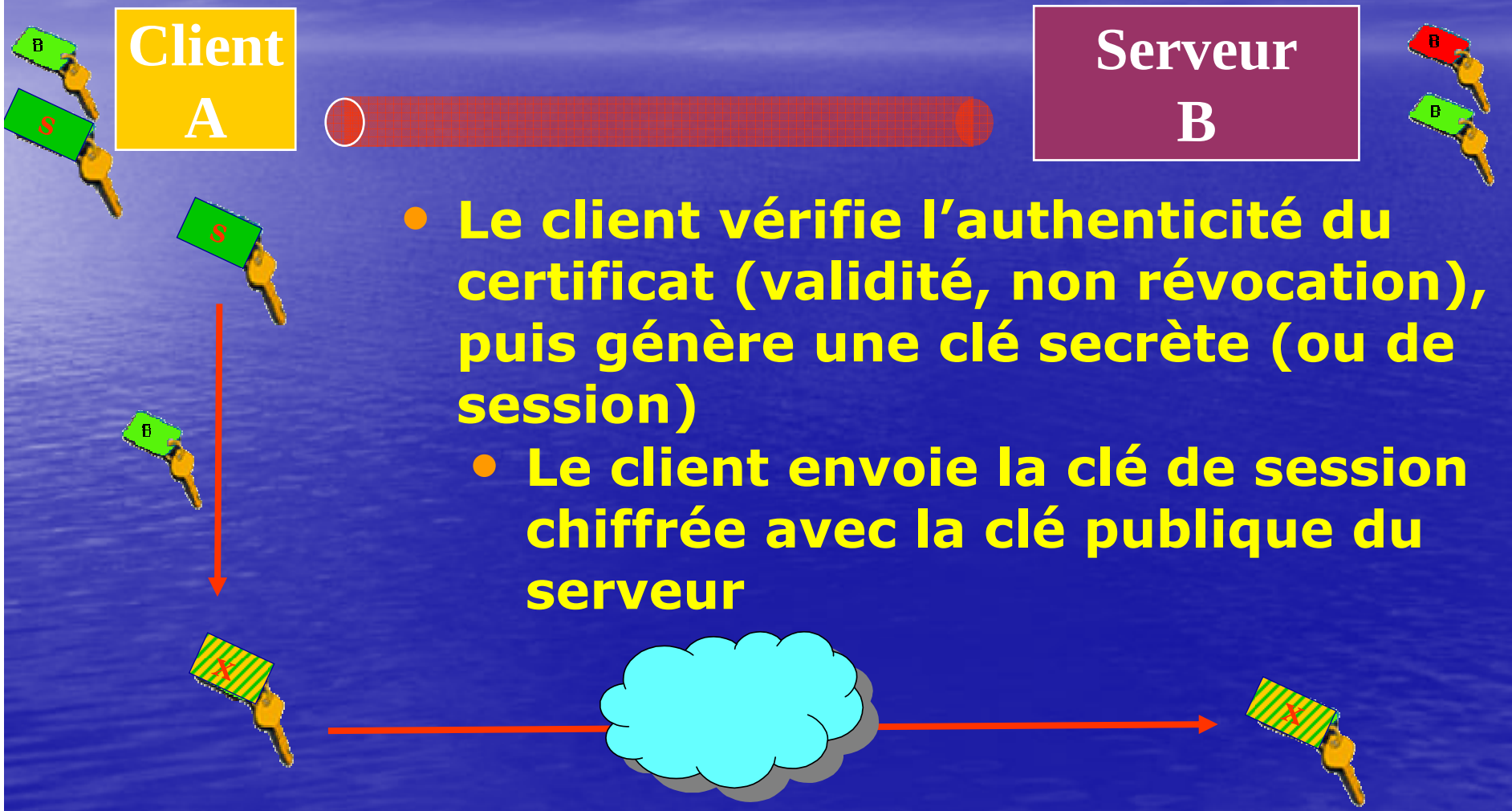
Serveur B



Authentification serveur

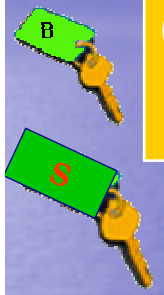
- 2. **Le client établit une connexion vers le serveur**
- 3. **Le serveur accepte la connexion non sécurisée**
- 4. **Le serveur envoie son certificat (intégrant sa clé publique) au client qui la mémorise**

Authentification serveur

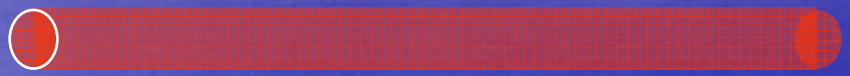


- Le client vérifie l'authenticité du certificat (validité, non révocation), puis génère une clé secrète (ou de session)
  - Le client envoie la clé de session chiffrée avec la clé publique du serveur

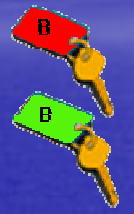
Authentification serveur



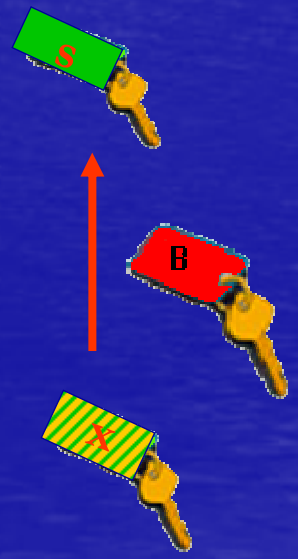
Client A

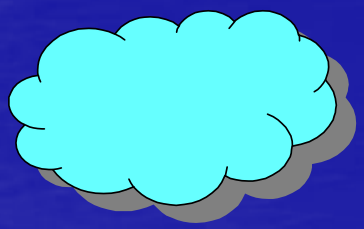


Serveur B



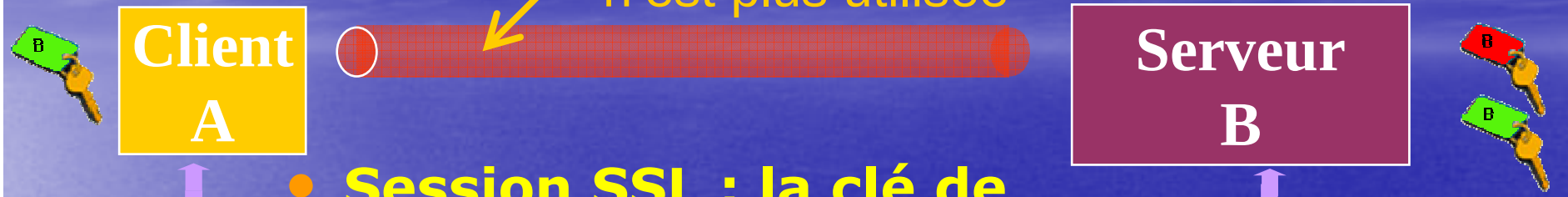
- Le serveur déchiffre la clé secrète avec sa clé privée





Authentification serveur

La connexion *en clair* n'est plus utilisée

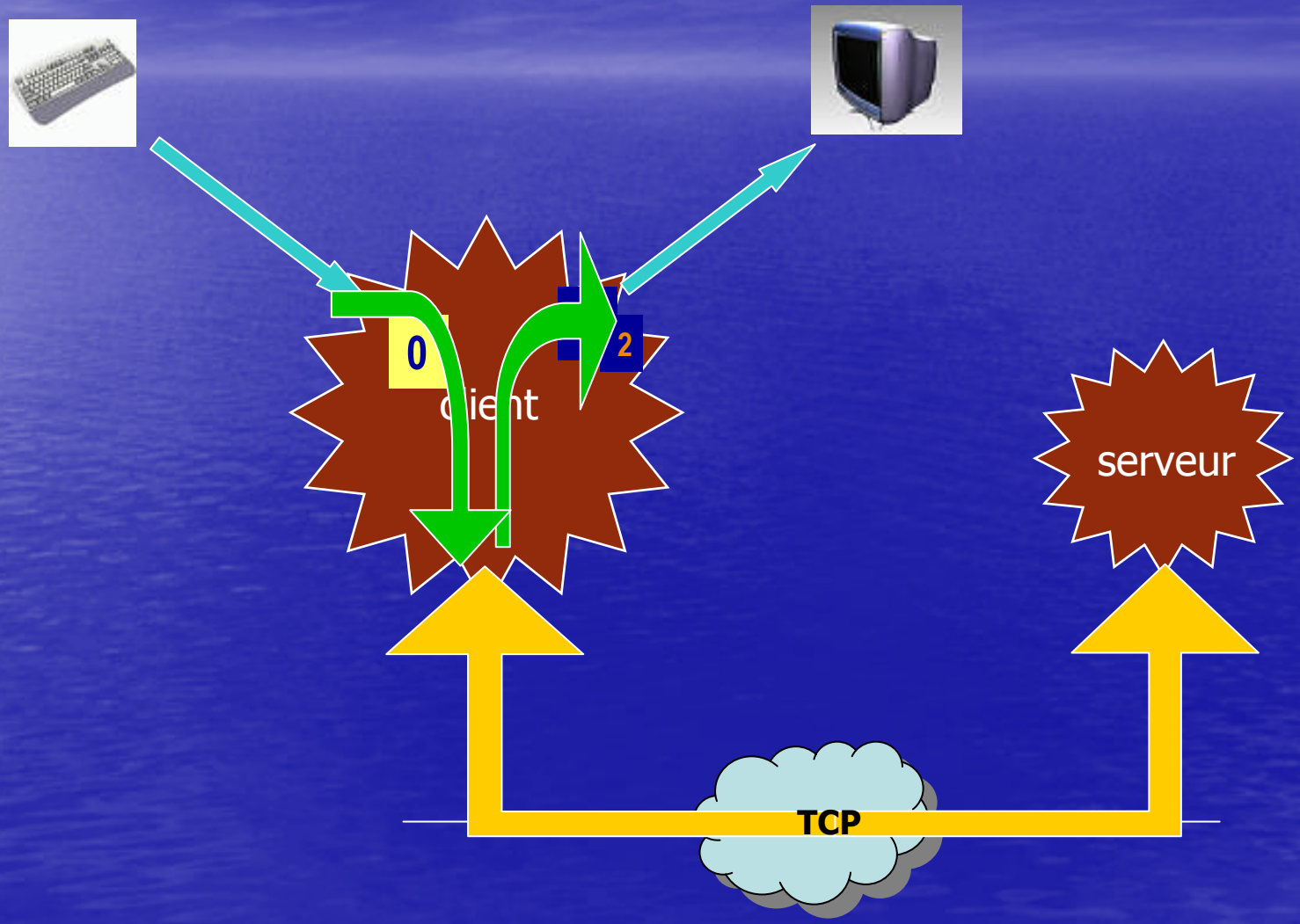


- **Session SSL : la clé de session chiffre et déchiffre les communications**

SSL et services

- Largement utilisé pour mettre en œuvre des communications sécurisées de niveau applicatif de type VPN (*Virtual Private Network*)
  - Ports des services utilisant SSL/TLS
    - FTPS : 990
    - HTTPS : 443
    - SSMTP : 465
    - SNNTP : 563
    - POP3S : 995
    - IMAPS : 993
    - …

telnet : shell distant

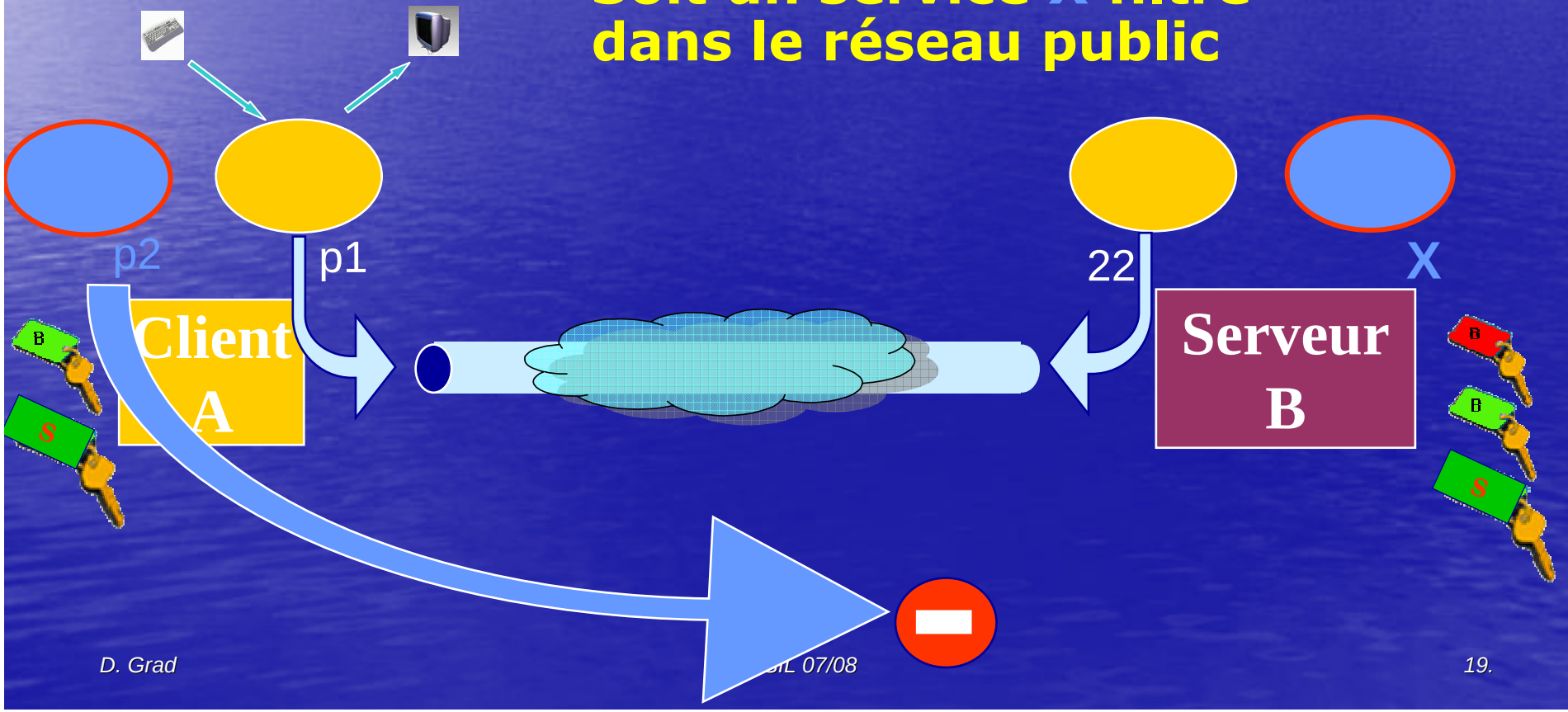


SSH (Secure Shell)

- SSH : shell sécurisé
  - Terminal distant sécurisé , équivalent à rlogin, rsh, telnet sécurisé
  - Modèle d'interaction standard ssl
  - Inclut le transfert de fichiers
  - Effectue une compression des données
  - Permet la redirection de ports
  - Combinaison de chiffrement symétrique ET asymétrique
  - Standardisé à l'IETF sous SSHv2

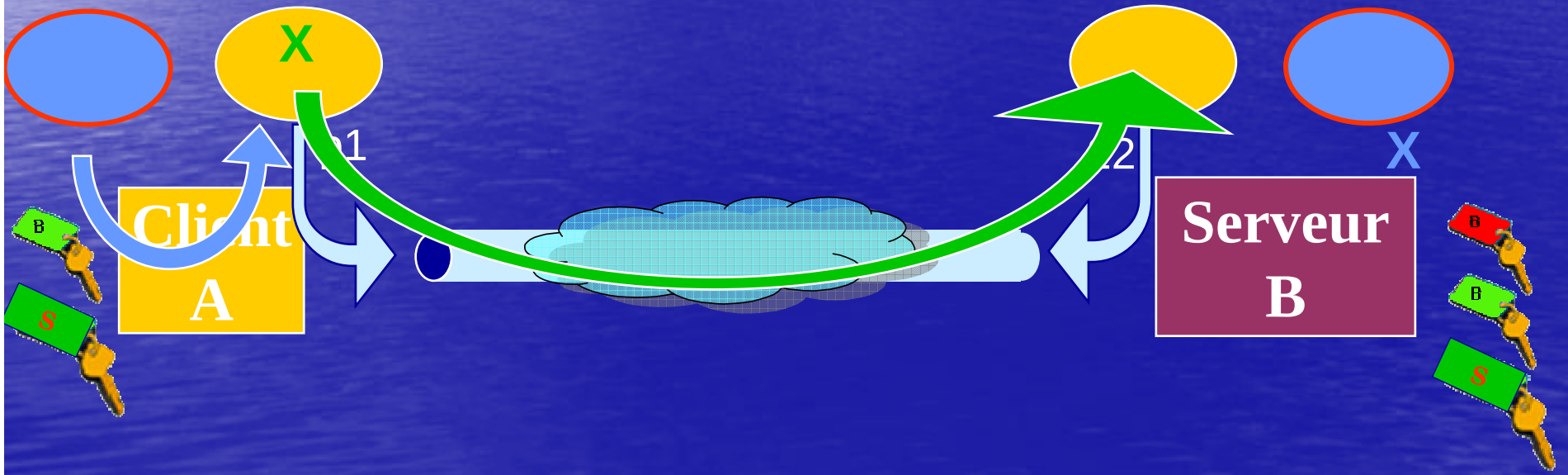
SSH et redirection de ports

- Session SSH est établie entre le client A et le serveur
  - Soit un service X filtré dans le réseau public



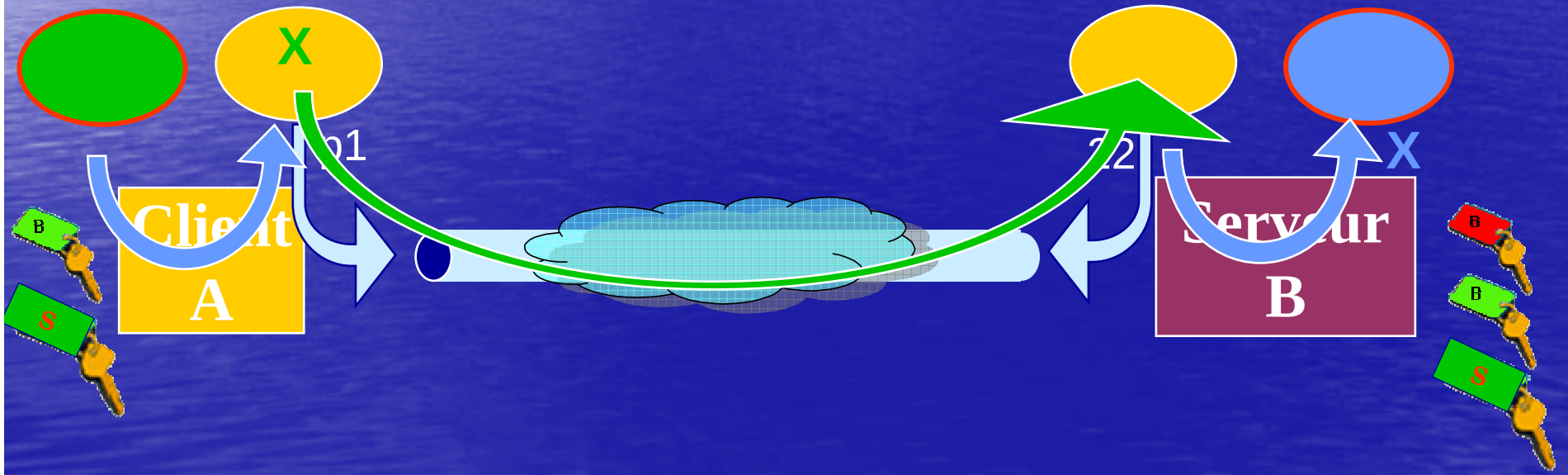
SSH et redirection de ports

- Par ssh, on configure sur le client la redirection du port X vers le serveur
- Les données du service X sont redirigées, dans la connexion ssh



SSH et redirection de ports

- Le serveur X reçoit la demande comme si elle avait été émise par un processus local



Le service HTTP sécurisé

- Le service www largement utilisé dans l'Internet utilise le protocole HTTP et TCP sur le port 80
- Le service standard ne fournit aucune sécurité
  → toutes les informations circulent en clair
- Le serveur HTTP le plus répandu est le serveur Apache (http://www.apache.org)
- Il existe deux solutions pour sécuriser un serveur HTTP et utiliser https
  - Apache-SSL (http://www.apache-ssl.org)
  - Apache modSSL (http://www.modssl.org)
- Le service https est rendu suivant le modèle d'interaction présenté précédemment avec TCP sur le port 443

Kerberos v5 (Rfc1510)

- Protocole d'authentification réseau destiné aux applications client/serveur (MIT)
  - Serveur Kerberos
  - Service Réseau en mode connecté (sur TCP)
  - Client du service Réseau
- 3 niveaux d'authentification
  - À l'établissement de la connexion
  - A chaque message
  - Avec chiffrement de chaque message

Architecture Kerberos

Rend le service S

ClientC souhaite accéder au ServeurS

KDC Key Distribution Center
TGS Ticket Granting Server

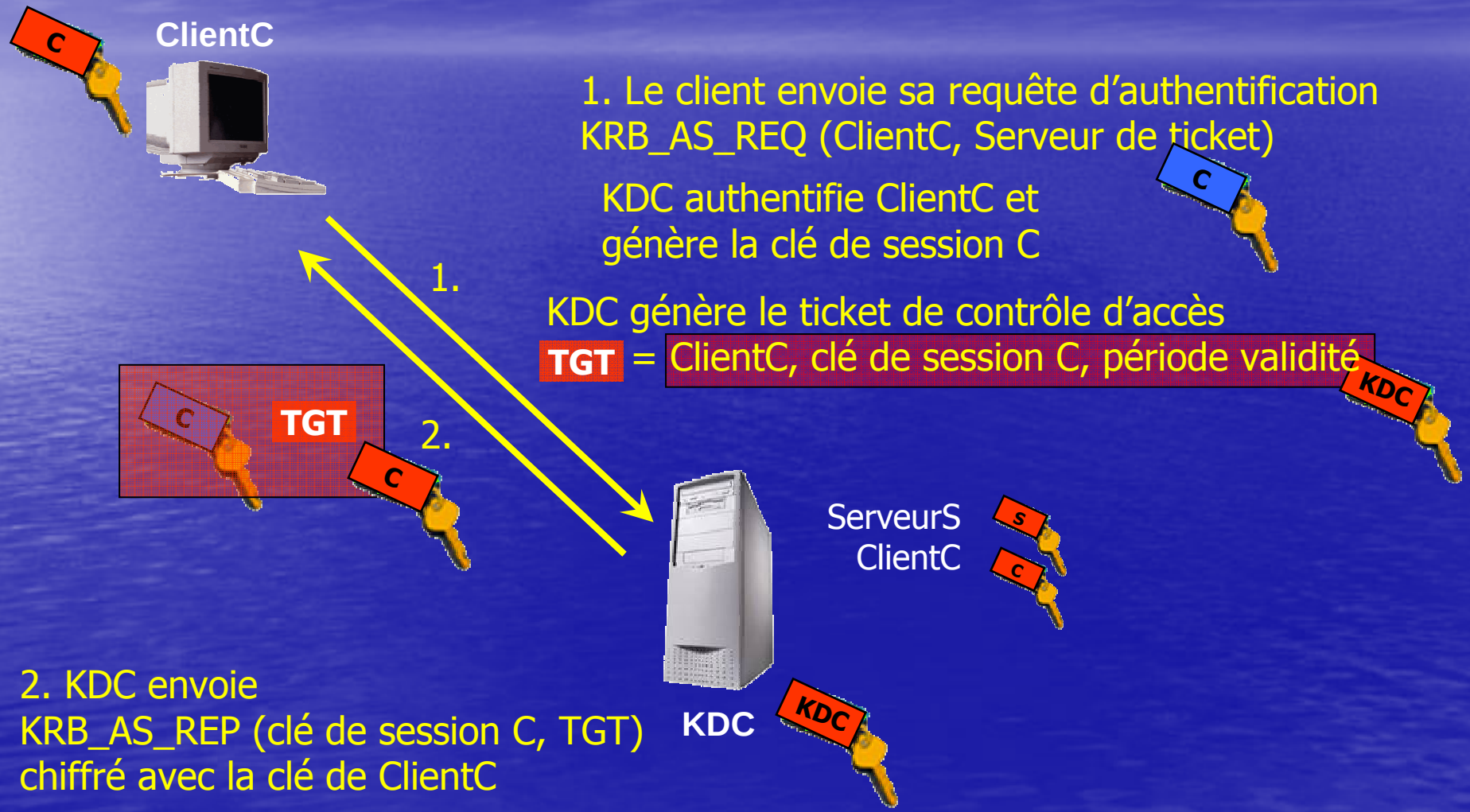
Chaque *principal*
- ***Utilisateur***
- ***Machine***
- ***Service***

possède une clé secrète

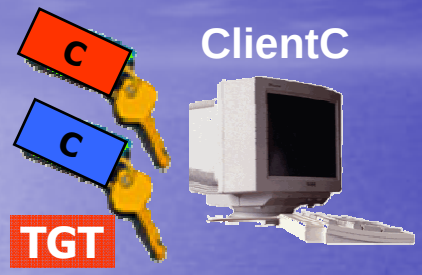
ServeurS
ClientC

KDC possède une base de données *principal / clé secrète*

Architecture Kerberos
Phase 1 : authentification



Architecture Kerberos
Phase 1 : authentification



- Seuls ClientC et KDC peuvent déchiffrer 

- Seul KDC peut déchiffrer TGT
- La vérification de l'identité de ClientC n'est réalisée qu'une fois lors de la connexion de l'utilisateur sur le ClientC
- KDC ne mémorise ni 

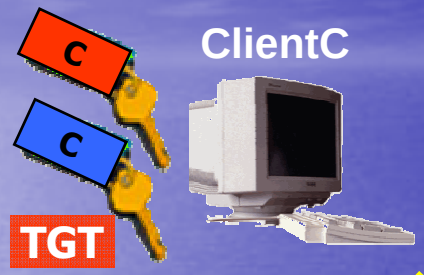
ni TGT

ServeurS
ClientC

KDC

Architecture Kerberos

Phase 2 : Obtenir un ticket d'accès au service



3. Le client envoie la demande d'accès à ServeurS
KRB_TGS_REQ (ServeurS, **TGT**, date courante)

3.

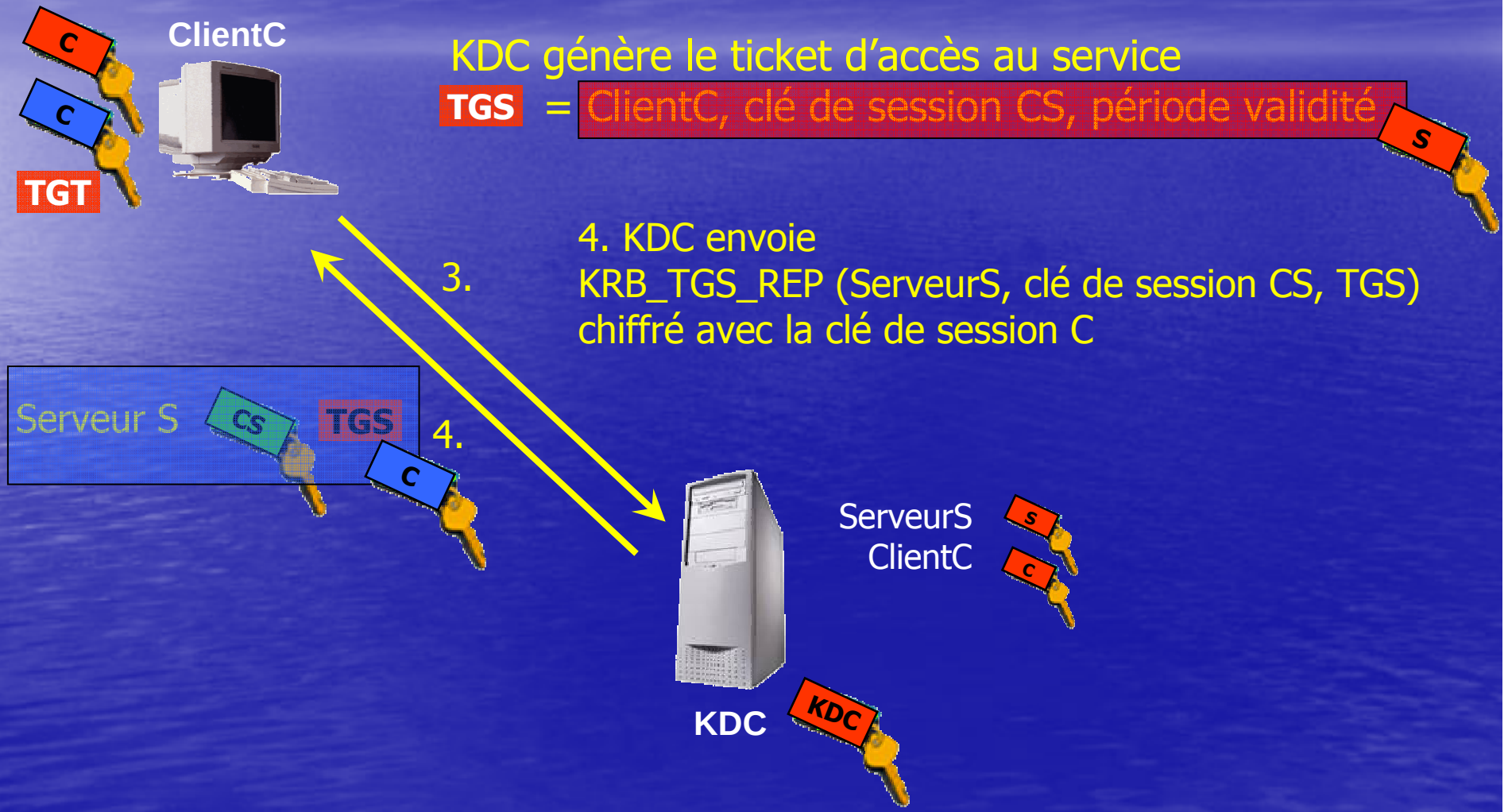
- KDC déchiffre **TGT** pour obtenir
- KDC déchiffre la date et vérifie la synchro
- KDC consulte les ACL pour accès de C à S
- KDC génère une clé de session

ServeurS
ClientC

KDC

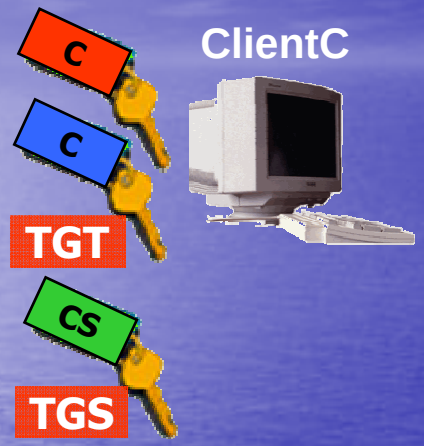
Architecture Kerberos

Phase 2 : Obtenir un ticket d'accès au service

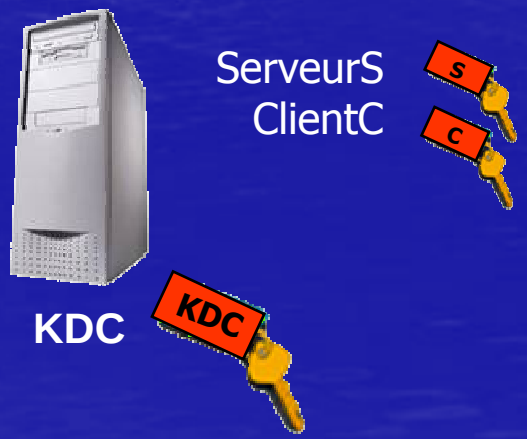


Architecture Kerberos

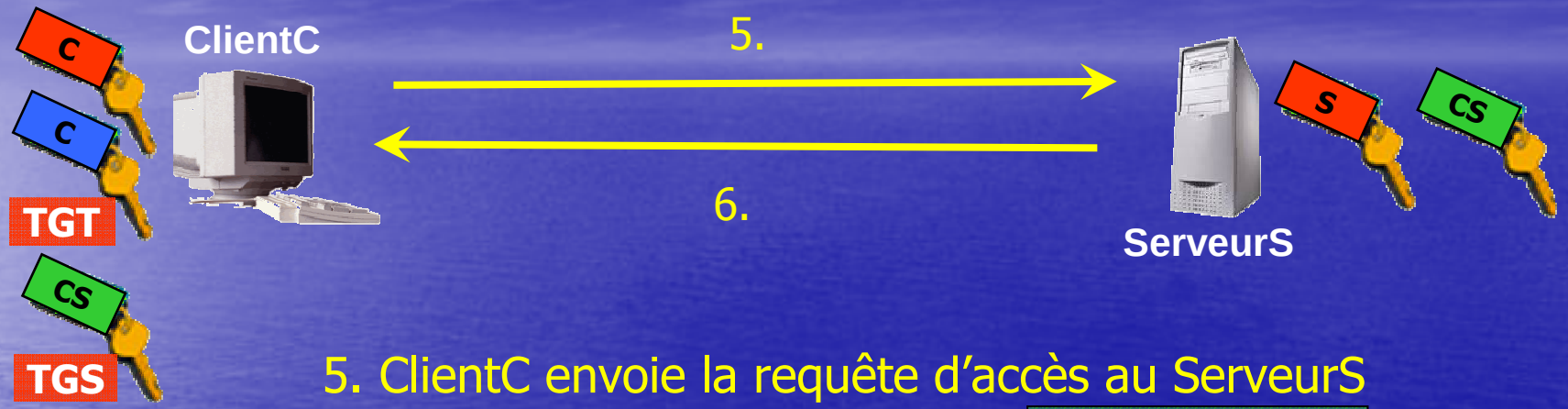
Phase 2 : Obtenir un ticket d'accès au service



- Seuls ClientC et KDC peuvent déchiffrer la réponse et obtenir CS
- Seuls ServeurS et KDC peuvent déchiffrer TGS
- KDC ne mémorise ni TGS ni CS
- Tant que TGS reste valable, ClientC ne contacte plus KDC



Architecture Kerberos
Phase 3 : Accès au service



5. ClientC envoie la requête d'accès au ServeurS
KRB_AP_REQ (Serveur S, **TGS** , date courante)$_{CS}$

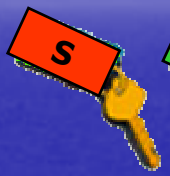
- ServeurS déchiffre **TGS** pour obtenir
- ServeurS déchiffre la date, vérifie la synchro et la validité du ticket

6. ServeurS envoie en réponse
KRB_AP_REP (date courante+1)$_{CS}$

ClientC déchiffre la réponse et vérifie la date+1 pour authentifier ServeurS

Architecture Kerberos
Phase 3 : Accès au service

ClientC



ServeurS

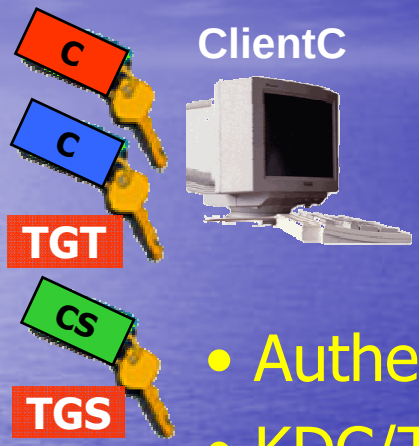
- ClientC et ServeurS se sont mutuellement authentifiés et partagent le secret

- Possibilité de chiffrer les échanges futurs entre ClientC et ServeurS

ServeurS
ClientC

KDC

Architecture Kerberos
Bilan



- Authentification forte entre clients et serveurs
- KDC/TGS considéré comme « tiers de confiance »
- Mécanisme « anti-rejeu » avec dates de validité
- Inconvénients
  - Nécessite la synchronisation des horloges
  - KDC mémorise les clés secrètes de tous les *personals* du *realm (à moins de la mise en place des certificats)*
-

Authentification des utilisateurs du DPt

- LDAP : Lightweight Directory Access Protocol : protocole d'accès aux annuaires X500 (RFC 3377)
- Active Directory : annuaire des ressources partagées (Win2K Server), inclus un annuaire de type LDAP
- Kerberos : protocole d'authentification, à base de tickets
- SSO : Single Sign On, mécanismes permettant de ne s'authentifier qu'une seule fois

Bilan TP1

- Rendus délai 14/12
- S51_TP1_<nom>.pdf |zip | mail signé
- Rendus 40
  - Hors délai 20
  - NON signé 5
  - NON zippé 31
  - Nom KO 5
  - Tout sauf 1 6 !
  - Tout OK 4 !
- RIEN 3 + 3