

# Cryptographie et sécurité

## Cours 1: sélectionner les algorithmes adéquats

Mickaël Bettinelli  
([mickael.bettinelli@univ-smb.fr](mailto:mickael.bettinelli@univ-smb.fr))



# Prérequis et objectifs

Connaissances nécessaire pour R4.C.08:

- Connaître et comprendre le fonctionnement du chiffrement symétrique et asymétrique (RSA)
- Comprendre le fonctionnement des fonctions de hachage

Connaissances et compétences à acquérir:

- Savoir sélectionner le bon algorithme en fonction de votre besoin
- Orchestrer des algorithmes de cryptographie pour sécuriser des communications
- Concevoir et implémenter une application sécurisée de partage de fichiers



# Evaluation

A réaliser: 1 projet sur 5 séances.

Un rapport à rendre qui sera noté sur 3 critères:

- Sécuriser des fichiers (chiffrement symétrique) / 6 pts
- Sécuriser les communications (chiffrement asymétrique) / 8 pts
- Sécuriser le transfert de fichier (fonctions de hachage) / 6 pts



# Evaluation

Contenu du rapport:

- Maximum 3 pages, pas de page de garde, pas de sommaire.
- Description des tâches réalisées
- Justification des technologies/algorithmes choisis
- Diagramme de classes de votre projet
- Instructions pour l'exécuter (comprenant les éventuels packages à installer)

Envoyer le rapport + le code dans une archive:

<https://forms.gle/jz784Zjv6Ef4tbeNA>



# Sommaire de ce cours

1. Chiffrements symétriques vs chiffrements asymétriques
2. Orchestration des méthodes de chiffrements pour une application communicante
3. Objectif et planning des prochaines séances



# Symétrique vs asymétrique - fonctionnement

## Symétrique

- La même clé est utilisée pour chiffrer et déchiffrer des données

## Asymétrique

- Utilisation d'une clé publique et privée
- Chiffrer avec une clé publique permet la confidentialité du message
- Chiffrer avec une clé privée permet d'authentifier le message



# Symétrique vs asymétrique - cas d'application

Symétrique

Asymétrique

Assurer la confidentialité de données



Authentifier le propriétaire de la donnée





# Orchestration des chiffrements

Quelles méthodes de sécurité ?

Comment les utiliser ensemble dans une application communicante ?

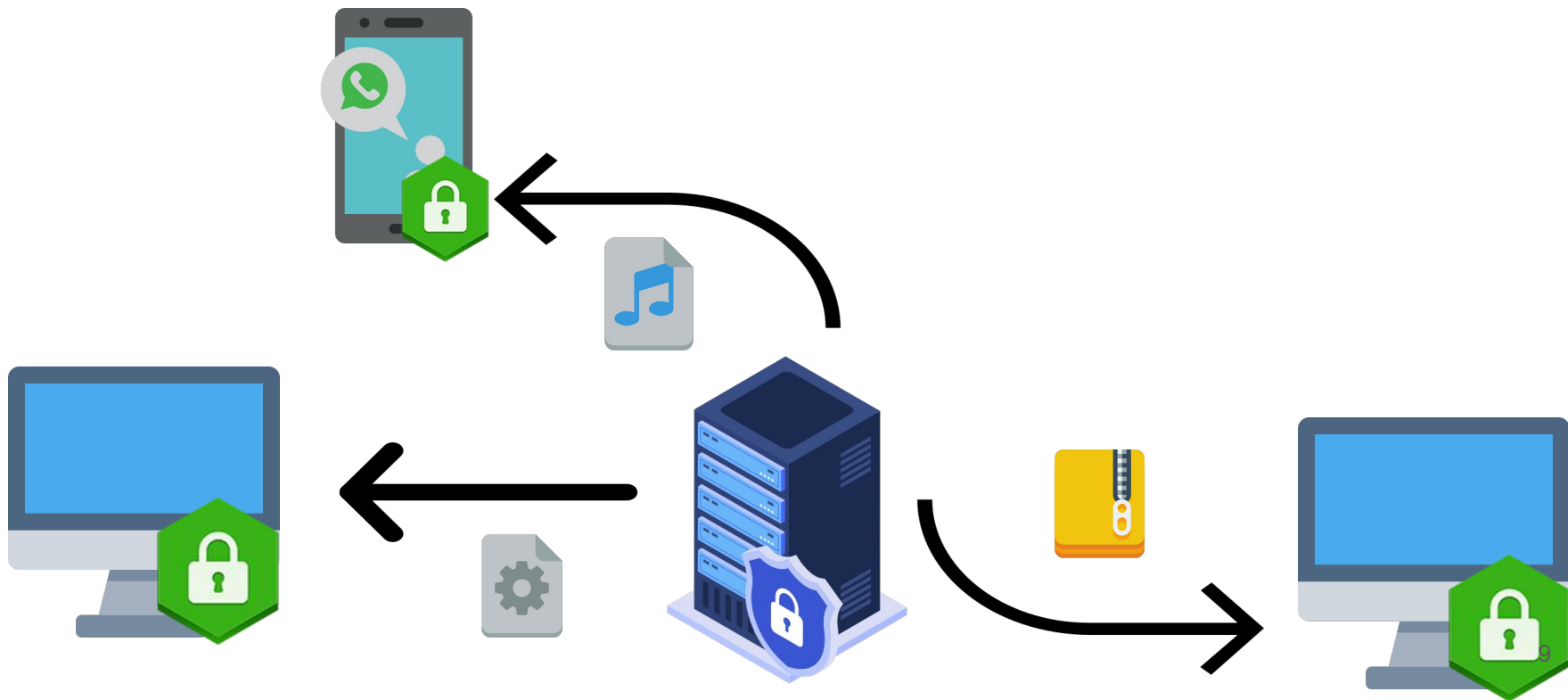
- Les méthodes asymétriques → partager des clés entre les interlocuteurs
- Les méthodes symétriques → partager des données contenant de la sémantique (et des motifs)
- Les fonctions de hachage → vérifier l'intégrité des données





# Objectif de la ressource

Implémenter une application sécurisée de partage de fichiers

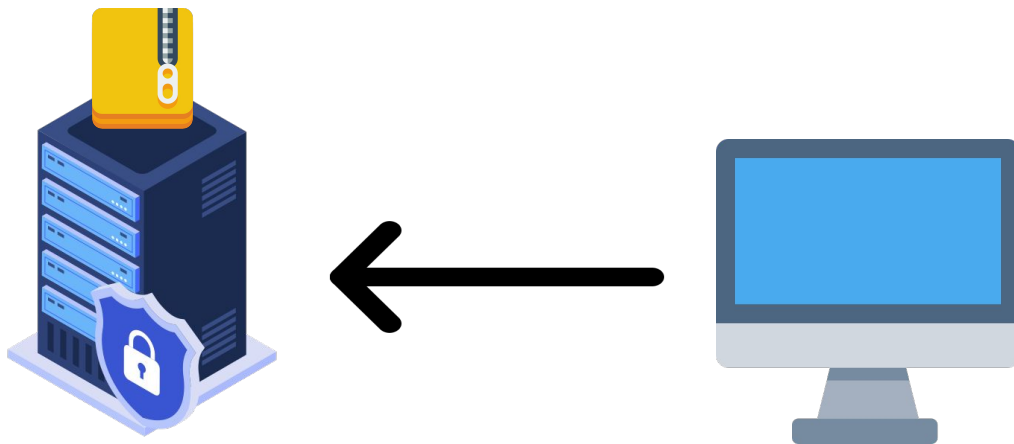




# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- **Un serveur** détient un fichier à envoyer à un client
- **Un client** se connecte au serveur pour télécharger le fichier





# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le client et le serveur **génèrent une clé publique et privée**



$K_{-, s}$



$K_{+, s}$



$K_{-, c}$



$K_{+, c}$





# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le serveur et le client **s'échangent leur clé publique**



$K_{-,s}$



$K_{+,s}$



$K_{+,c}$



$K_{+,c}$



$K_{+,s}$



$K_{-,c}$



$K_{+,c}$



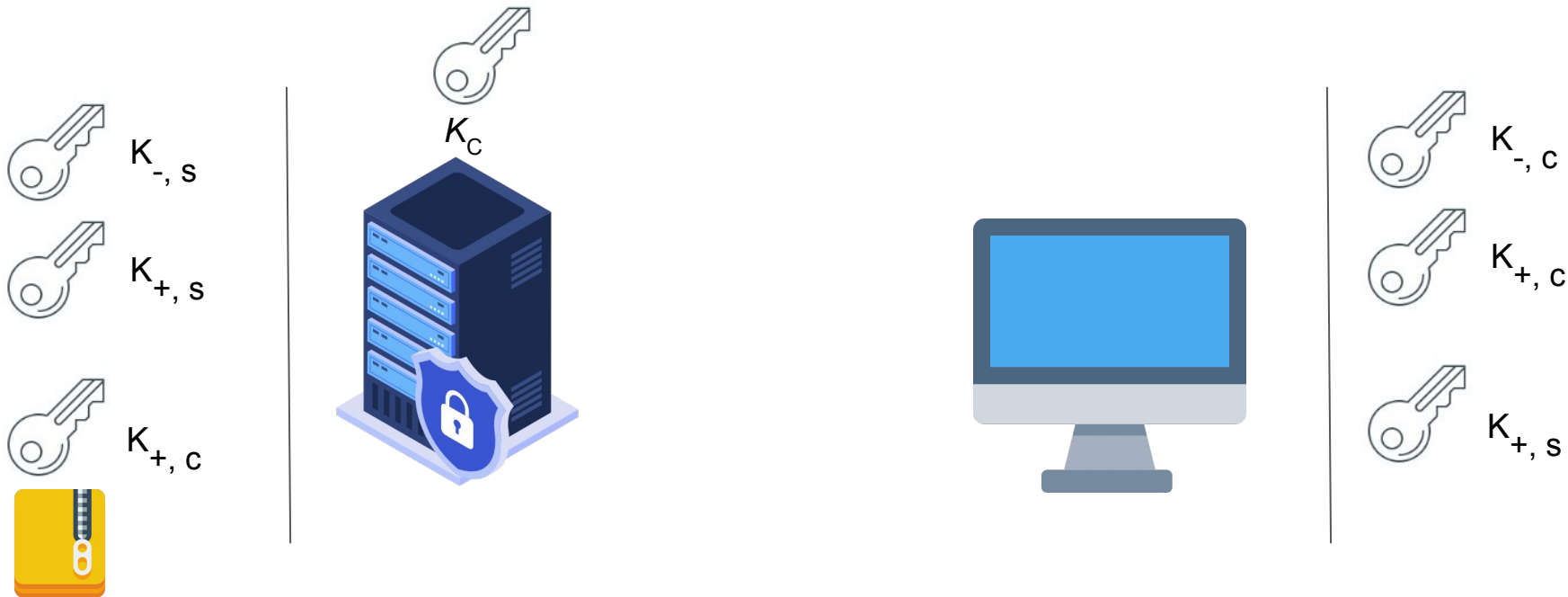
$K_{+,s}$



# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le serveur **génère** une clé  $K_c$  pour chiffrer son fichier





# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le serveur envoie  $K_c$  au client avec RSA



$K_{-,s}$



$K_{+,s}$



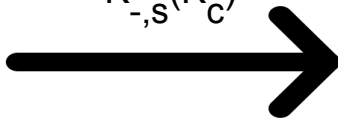
$K_{+,c}$



$K_c$



$K_{-,s}(K_c)$



$K_{-,c}$



$K_{+,c}$



$K_{+,s}$



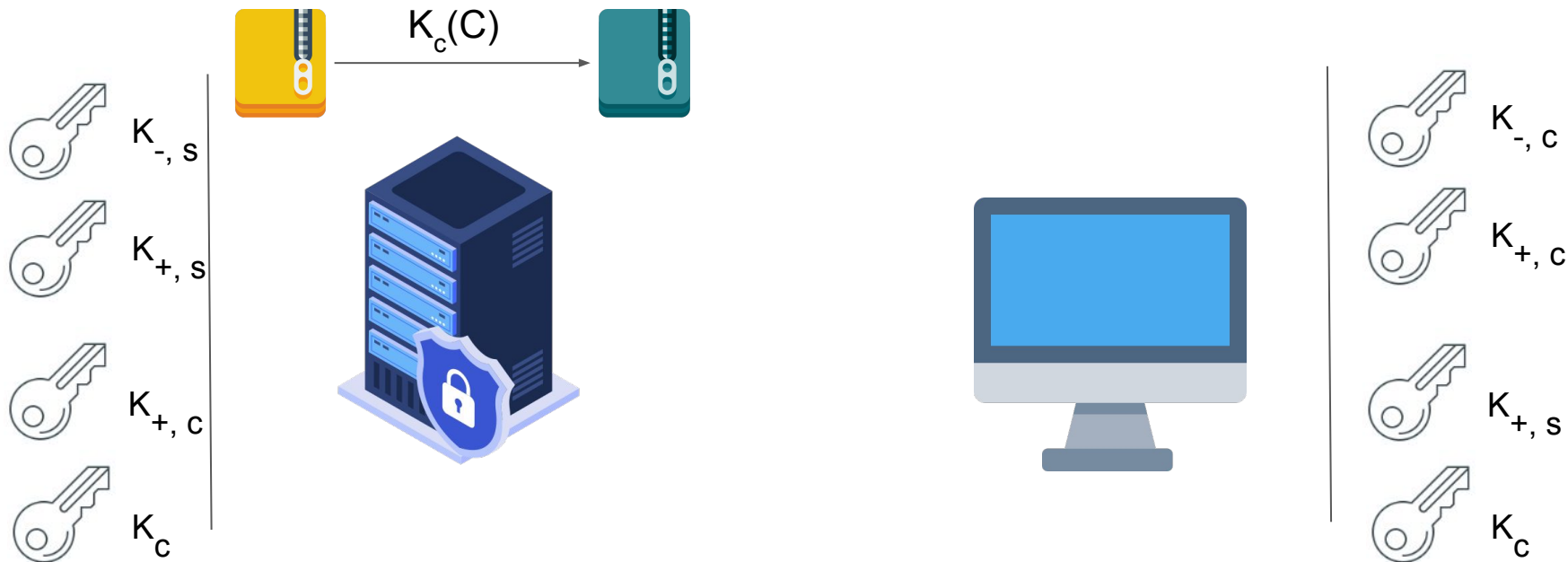
$K_c$



# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le serveur chiffre le fichier avec AES et la clé  $K_c$

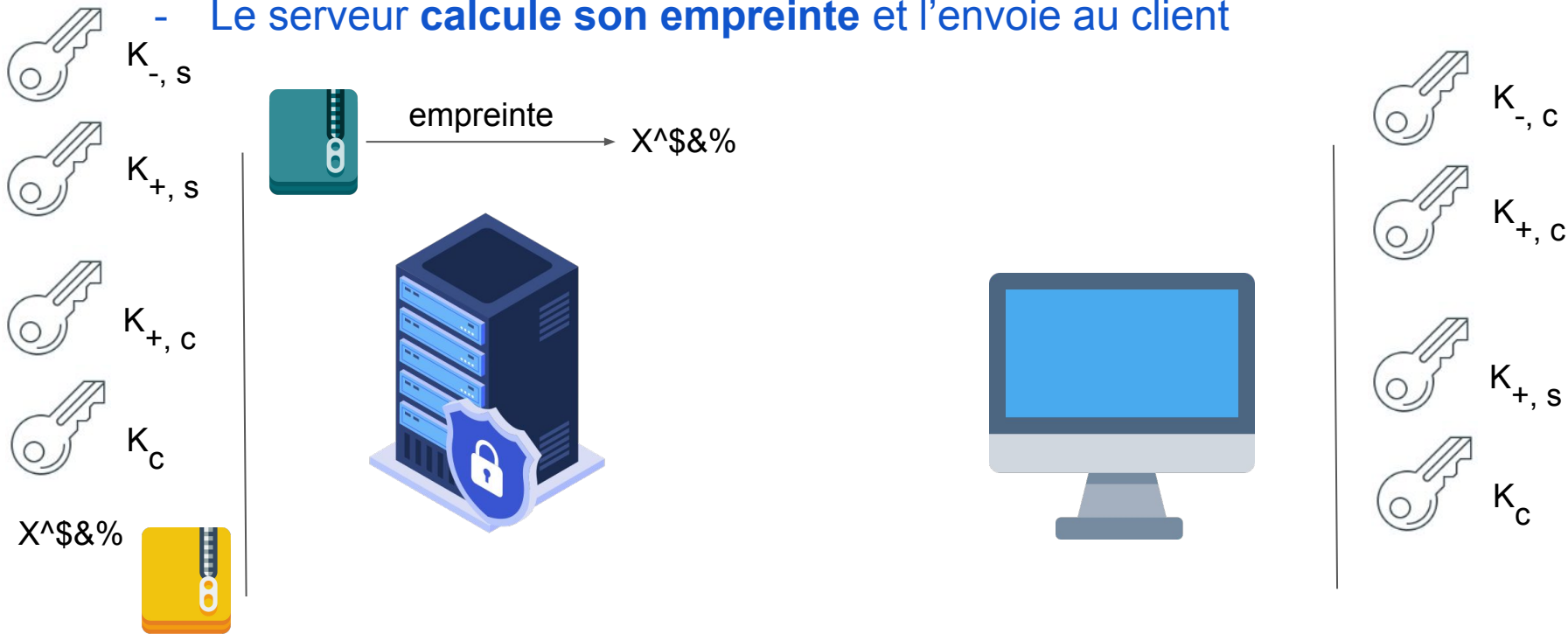




# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le serveur calcule son empreinte et l'envoie au client



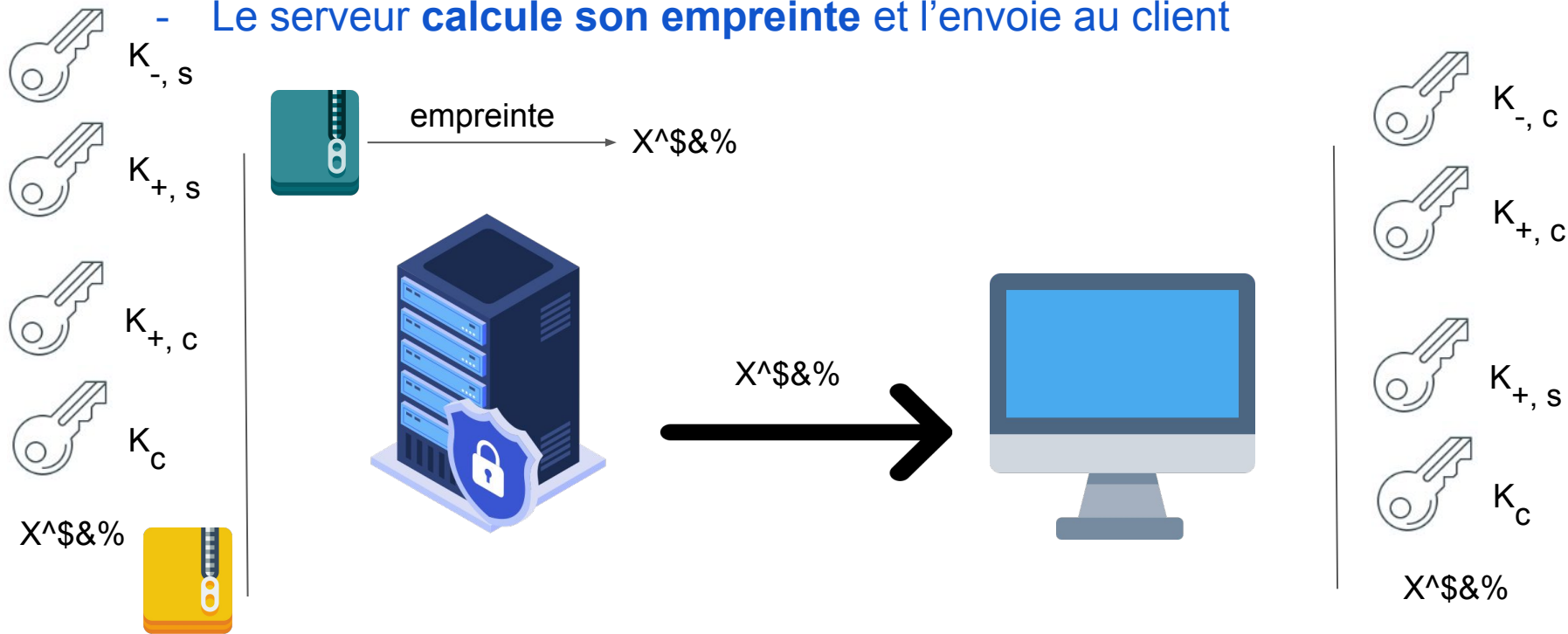




# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le serveur calcule son empreinte et l'envoie au client





# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le serveur transmet le fichier chiffré



$K_{-, s}$



$K_{+, s}$



$K_{+, c}$



$K_c$

$X^{\$ \& \%}$



$K_{-, c}$



$K_{+, c}$



$K_{+, s}$



$K_c$

$X^{\$ \& \%}$



# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le client reçoit le fichier et **compare son empreinte** avec celle reçue



$K_{-,s}$



$K_{+,s}$



$K_{+,c}$



$K_c$

$X^{\$ \& \%}$



empreinte

$X^{\$ \& \%}$



$K_{-,c}$



$K_{+,c}$



$K_{+,s}$



$K_c$

$X^{\$ \& \%}$



# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- Le client déchiffre le fichier avec  $K_c$



$K_{-, s}$



$K_{+, s}$



$K_{+, c}$



$K_c$

$X^{^{\$ \& \%}}$



$K_c(C')$



$K_{-, c}$



$K_{+, c}$



$K_{+, s}$



$K_c$

$X^{^{\$ \& \%}}$





# Vue d'ensemble des tâches

## Implémenter une application sécurisée de partage de fichiers

- **Un serveur** détient un fichier à envoyer à un client
- **Un client** se connecte au serveur pour télécharger le fichier
- Le client et le serveur **génèrent une clé publique et privée**
- Le serveur et le client **s'échangent leur clé publique**
- Le serveur **génère une clé  $K_c$**  pour chiffrer son fichier
- Le serveur **envoie  $K_c$  au client avec RSA**
- Le serveur **chiffre le fichier avec AES et la clé  $K_c$**
- Le serveur **calcule son empreinte** et l'envoie au client
- Le serveur transmet le fichier chiffré
- Le client reçoit le fichier et **compare son empreinte** avec celle reçue
- Le client **déchiffre le fichier avec  $K_c$**



# Planning des séances

Tâches\Séances	06/03	06/03	13/03	20/03	27/03	03/04
Prise en main du code (architecture client/serveur)	(CM1)					
Implémentation de RSA						
Échange des clés publiques						
Génération de la clé AES et échange avec le client			(CM2)			
(Dé)Chiffrement du fichier						
Calcul de l'empreinte du fichier chiffré				(CM3)		
Échange et comparaison des empreintes						



# Planning des séances

Tâches\Séances	06/03	06/03	13/03	21/03	27/03	03/04
Prise en main du code (architecture client/serveur)	(CM1)					
Implémentation de RSA						
Échange des clés publiques						
Génération de la clé AES et échange avec le client			(CM2)			
(Dé)Chiffrement du fichier						
Calcul de l'empreinte du fichier chiffré avec SHA-3				(CM3)		
Échange et comparaison des empreintes						

## Le 03/04, au choix:

- présentation des certificats
- introduction à la blockchain
- une séance de plus pour finir votre application



## Derniers détails

- Vous devez respecter le fonctionnement de l'application tel que présenté dans le slide 10.
- RSA et l'architecture client/serveur fournie sont imposées. Vous avez le choix sur tous les autres algorithmes de sécurité.
- Le code de l'architecture client/serveur peut être modifié
- Minimum attendu “*pour avoir 10*”: échange de clés publique/privée et transmission d'un fichier chiffré
- Chaque fonctionnalité supplémentaire liée à la sécurité peut donner droit à des points bonus lors de la correction (si mentionné dans le rapport)