Architecture du logiciel

Version	0.1
Date	7 Janvier 2013
Rédigé par	Yves Nouafo
Relu par	
Approuvé par	

MISES À JOUR

Version	Date	Modifications réalisées
0.1	16/11/2012	Création
0.2	23/11/2012	Avancement des schémas et complétion du document
0.3	1/12/2012	Modificatoin des schémas UML
0.4	12/12/2012	Ajout UC.19 – UC.18 – UC.17
0.5	14/12/2012	Modification après réunion du 12/12/12

Table des matières

1 Objet

Le présent document montre l'architecture utilisée pour réaliser un système de bavarde (ou chat en anglais) sécurisé. Plusieurs modules seront développés de manière indépendante, chacune ayant un rôle bien définie. Les différentes entités pourront communiquer entre elles et devront respecter les critères suivant :

- Gestion de la création et de la suppression d'un compte sécurisé ;
- Création par un utilisateur d'une salle de discussion pour un groupe de personnes ;
- Ajout/suppression d'un utilisateur autorisé dans une salle privé ;
- Confidentialité, intégrité et authentification sur les messages échangés ;
- Non-répudiation possible des messages ;
- Création d'une autorité de certification ;
- Demande de certificat pour accès au salon privé et communication sécurisé ;

2 Documents applicables et de référence

- IRC (RFC 2810 à 2813 de avril 2000)
- STB (Spécification Technique des Besoins)

3 Terminologie et sigles utilisés

- **Public Key Infrastructure (PKI)** : Infrastructure de gestion des clés publiques qui permet de gérer clés publiques et d'en assurer la fiabilité pour des entités dans un réseaux. Elle permet un échange sécurisé des informations en garantissant les principaux points de la cryptographie : l'intégrité, l'authentification, la confidentialité et la non-répudiation lors d'échanges d'informations.
- **Autorité d'enregistrement (AR)** : Organisme qui génère les certificats et effectue les vérifications d'usage sur les utilisateurs.
- **Confidentialité** : Les informations échangées deviennent illisibles, cette confidentialité est assurée par le chiffrement.
- **Non-répudiation** : L'émetteur des données ne peut pas nier être à l'origine du message.
- **Intégration** : Fonctionnement permettant d'assurer que l'information n'a pas subi de modification.
- **Signature** : Code électronique unique qui permet de signer un message codé. Elle permet d'identifier l'origine du message.
- **Certificat** : Document électronique qui fait correspondre une clé avec une entité. Cette correspondance est validée par une autorité de certification.
- **Authentification (Authent)** : connexion sur le chat via un identifiant.
- **Authentification sécurisé (Authent S)** : Identification de l'origine de l'information.
- **Liste de dépots (LD)** : A pour mission de stocker les certificats numériques ainsi que les listes de révocation.
- **Client sécurisé (Client S)** : A pour mission de chiffrer et déchiffrer les messages.

- **Base de données (BDD)** : Conteneur informatique permettant de stocker dans un même endroit l'intégralité des informations.

4 Configuration requise

4.1 Performances du calculateur

- 1Go RAM
- Intel Celeron
- Virtual Machine (VM)

4.2 Système d'exploitation

- MacOS 10.8.2 Mountain Lion
- Linux 3.X – unix (Ubuntu 12.X ou LTS)
- (test a effectuer sur debian)

4.3 Produits logiciels nécessaires

- Tinyca (interface OpenSSL pour la creation de certificats)
- OpenSSL
- GTK

5 Architecture statique

5.1 Structure

Les principales parties à développer :

- L'application client : le système de bavardage
- Les serveurs :
  - Le serveur non sécurisé
  - Le serveur sécurisé
  - La PKI avec : CA, RA, politique de certification
- Les données :
  - Liste dépots
  - Bases de données gérant les utilisateurs non sécurisés et sécurisés

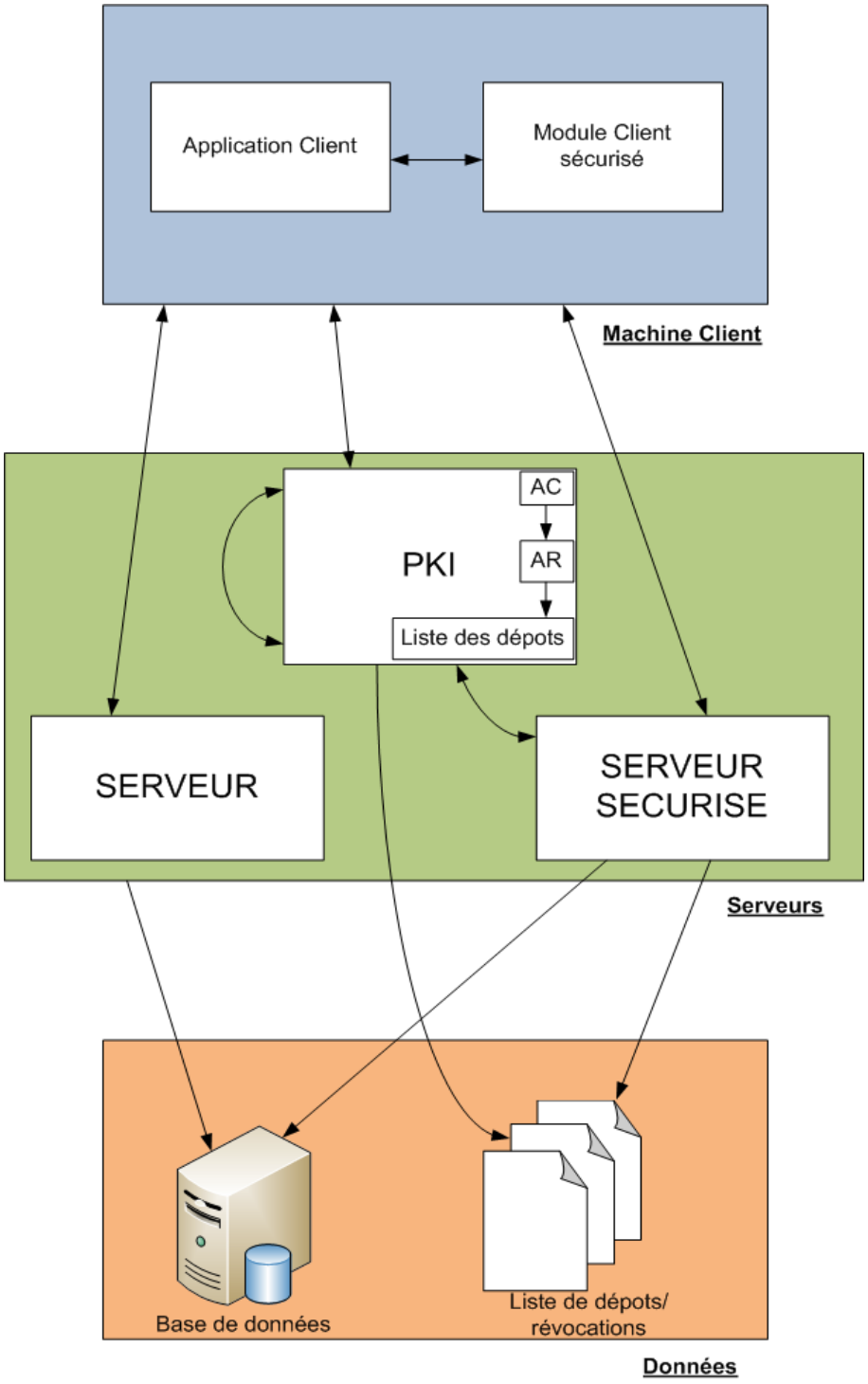


FIGURE 1 – Fonctionnement de l'application

5.2 Description des constituants

	Application client
Rôle	– Fournir à l'utilisateur une interface pour bavarder
Propriétés et attributs de caractérisation	– Permet la communication entre utilisateurs grâce à l'envoi et la réception des messages – Création de salon
Dépendances avec d'autres constituants	– Server : envoi et/ou réception de messages – Server s : demande d'authentification et/ou réception des clés + enregistrement – PKI : demande et/ou reception de certificats ; Demande de certification
Langages de programmation	– C, Vala
Procédé de développement	– Établissement des fonctionnalités présente sur l'interface – Schématisation de l'interface – Développement de l'interface
Taille complexité	– 4% du projet – Complexité du à la programmation réseaux – Interfaçage du C par le Vala

	Application client sécurisé
Rôle	– Chiffrement et/ou déchiffrement des messages – Authentification sur le Server S – Création de salon privés
Propriétés et attributs de caractérisation	– Module en complément à l'application client
Dépendances avec d'autres constituants	– Server s : Demande d'authentification - réception des clés – Client : Transmission et/ou reception de messages chiffrés
Langages de programmation	– C, Vala
Procédé de développement	– fonctions de chiffrement et de déchiffrement – Intégration à l'application client – OpenSSL – Gestion des salons privés
Taille complexité	– 4% du projet – Complexité du à la programmation réseaux – Interfaçage du C avec le Vala – Couche sécurisée

	Serveur non sécurisé
Rôle	– Transmettre les messages entre utilisateurs – Gérer les salons
Propriétés et attributs de caractérisation	
Dépendances avec d'autres constituants	– Client : Réception et/ou transmission de messages aux utilisateurs concernés – BDD : Enregistrement des pseudonymes des utilisateurs connectés sur le logiciel de bavardage.
Langages de programmation	– C
Procédé de développement	– Réalisation de la partie réseau – Intégration la base de données – Gestion des utilisateurs et des salons
Taille complexité	– 25% du projet – Complexité du à la programmation réseaux

	Serveur sécurisé
Rôle	– Traiter la demande d'authentification du client – Gérer les clés et les salons privés
Propriétés et attributs de caractérisation	– Envoi des clés générées au client
Dépendances avec d'autres constituants	– Client : Réception et/ou envoi du traitement de la demande d'authentification – Client S : Échange de clés – BDD : Enregistrement des pseudonymes de tous les utilisateurs sécurisés du système de bavardage
Langages de programmation	– C
Procédé de développement	– Réalisation de la partie réseau – Intégration de la bibliothèque OpenSSL – authentification auprès de la PKI – Vérification des autentifiés auprès de la PKI
Taille complexité	– 25% du projet – Complexité du à la programmation réseaux et à l'ajout de la couche sécurisée

	PKI
Rôle	– S'assurer de la fiabilité des certificats des utilisateurs présents sur le système de bavardage – Entité de confiance
Propriétés et attributs de caractérisation	– Entité interne : CA qui crée le certificat, AR qui vérifie les conditions de demande de certification – Suit des règles pour délivrer des certificats : politique interne de certification
Dépendances avec d'autres constituants	– Client : attribution de certificats – PKI : Auto-certification – Server s : attribution d'un certificat – LD : Enregistrement des certificats délivrés/révoqués
Langages de programmation	– C
Procédé de développement	– TinyCA – OpenSSL
Taille complexité	– 40% du projet – Complexité du à la programmation réseaux et la couche sécurisée

	Liste de dépots
Rôle	– Sauvegarder les données fournies par les différentes entités qui composent le logiciel de bavardage (certificats)
Propriétés et attributs de caractérisation	Constitué de deux fichiers : – Liste d'enregistrement des certificats – Liste de revocation des certificats
Dépendances avec d'autres constituants	– Server s : Consultation des certificats – PKI : sauvegarde des certificats délivrés
Langages de programmation	
Procédé de développement	– Fichier restauré au cours du temps
Taille complexité	– 1% du projet

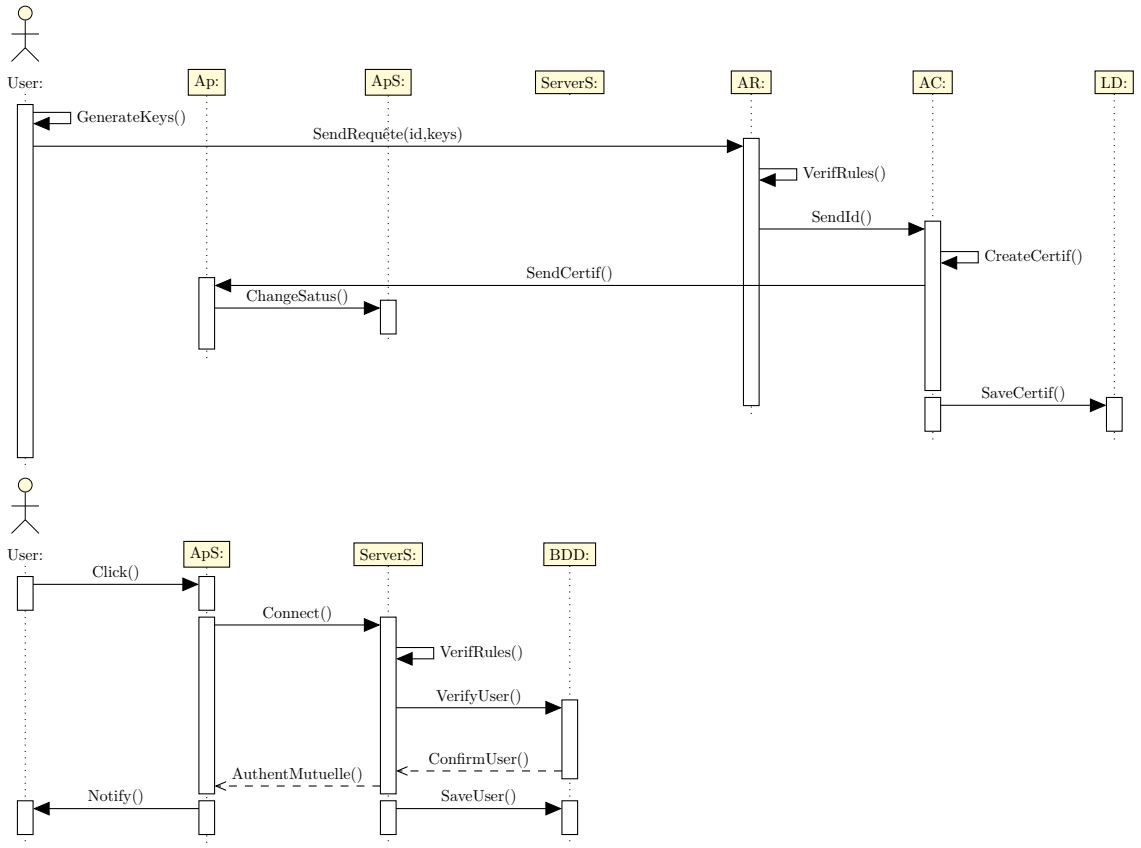
	Bases de données
Rôle	– Stocker les pseudonymes des utilisateurs (sécurisés ou non) présents sur le logiciel de bavardage
Propriétés et attributs de caractérisation	
Dépendances avec d'autres constituants	– Server/Server s : renseigne si un pseudonyme existe dans la base de données
Langages de programmation	– SqlLite
Procédé de développement	– Définir les différentes tables présentes dans les bases – Définir les relations entre les différentes tables – Créer les tables
Taille complexité	– 1% du projet

6 Fonctionnement dynamique

6.1 UC.1 : Création d'un compte utilisateur sécurisé

UC.1 : Création d'un compte utilisateur sécurisé	
Composants mis en jeu	User / AC / AR
Intervenants	Client
Processus de mise en œuvre	

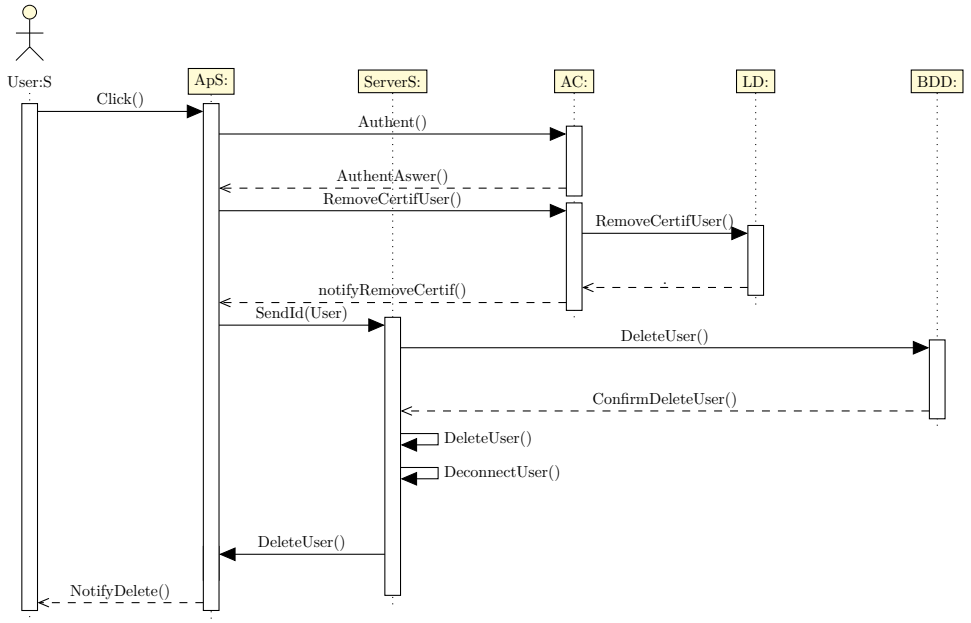
1. User envoi une demande de certificat à l'AR
2. Vérification de la demande par l'AR (selon politique de certification) et transfert de la demande à l'AC
3. Signature de la demande par l'AC et sauvegarde du certificat attribué
4. Envoi du certificat à l'User demandeur par l'AC
5. Sauvegarde du pseudonyme de l'utilisateur dans la base de données du Server s



6.2 UC.2 : Suppression d'un compte sécurisé par un utilisateur sécurisé

UC.2 : Suppression d'un compte sécurisé par un utilisateur sécurisé	
Composants mis en jeu	User S / Server S / AC
Intervenants	Client S
Processus de mise en œuvre	

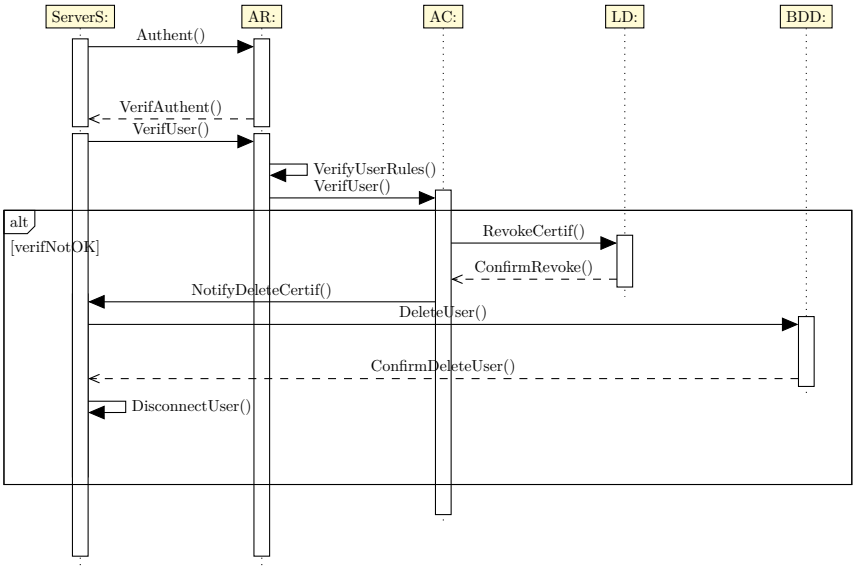
1. Demande de révocation de son compte par l'User s à l'AC
2. Suppression de l'User S dans la base de données du Server S
3. Vérification de l'existence du certificat par l'AC dans la LD
4. Suppression du certificat dans la LD
5. Déconnexion du Server S
6. Suppression de l'User dans la base de données du Server
7. Déconnexion du Server



6.3 UC.3 : Suppression d'un compte sécurisé par AC

UC.3 : Suppression d'un compte sécurisé par AC	
Composants mis en jeu	AC / Server S
Intervenants	User s, AR, LD, BDD
Processus de mise en œuvre	

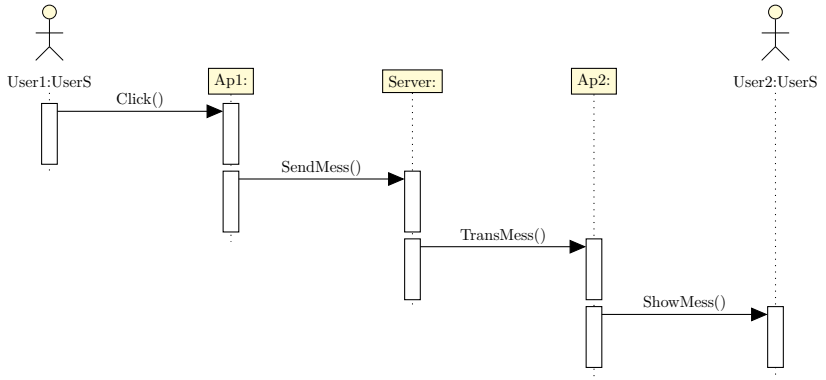
1. Identification de l'User S qui ne respecte pas les règles
2. Récupération de son certificat
3. Suppression de l'User S dans la base données
4. Suppression dans la LD du certificat de l'User S



6.4 UC.4 : Envoi de message sur salon général

UC.4 : Envoi de message sur salon général	
Composants mis en jeu	User / User S / Client / Server
Intervenants	
Processus de mise en œuvre	

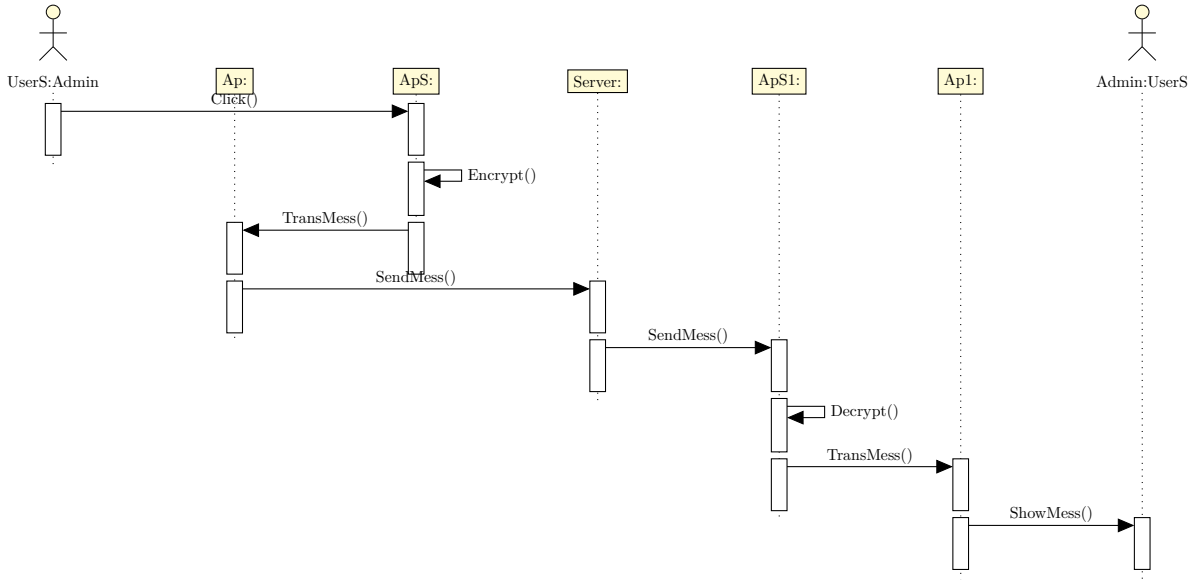
1. User, User S : utilisation du Client pour envoyer les messages sur le salon général
2. Les messages sont affichés par le Client après avoir été transférés par le Server



6.5 UC.5 : Envoi de message sur salon privé

UC.5 : Envoi de message sur salon privé	
Composants mis en jeu	User S / Admin / Server / Client S
Intervenants	
Processus de mise en œuvre	

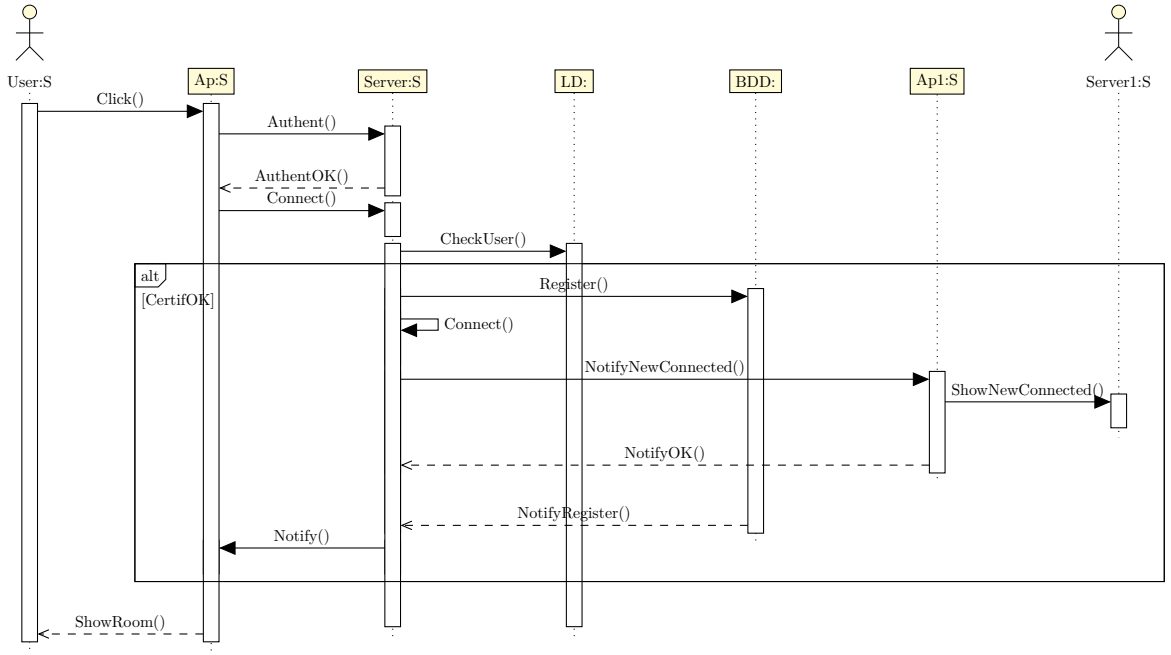
1. L'User S tape son message dans le Client
2. Le Client S chiffre son message grace à sa clé privée
3. Le Client S transmet le message au client
4. Le Client envoi son message au server avec l'identité du/des destinataire(s)
5. Le Server transmet le message
6. Le Client destinataire transmet le message à son Client S
7. Le Client S déchiffre le message grace à la clé publique de l'User S
8. Le Client affiche le message reçu par le(s) destinataire(s)



6.6 UC.6 : Rejoindre un serveur de chat sécurisé avec authentification

UC.6 : Rejoindre un serveur de chat sécurisé avec autentification	
Composants mis en jeu	User S / Server S / Client / Server
Intervenants	Client S, LD, BDD
Processus de mise en œuvre	

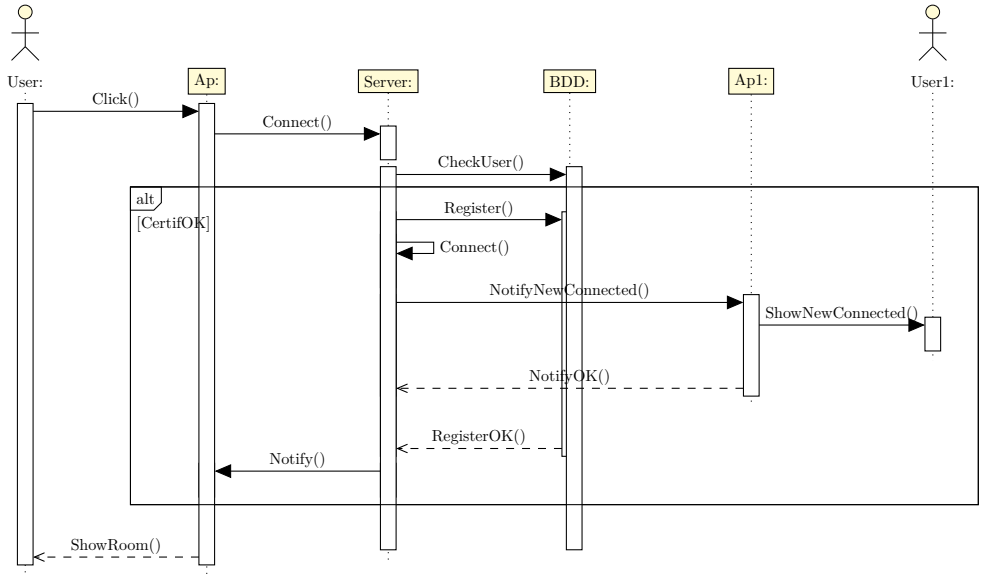
1. Demande de connection via le Client S
2. Envoi de la demande au Server S
3. Vérification du certificat par le Server S
4. Envoi de challenge pour vérifier l'identité du demandeur
5. Enregistrement de l'utilisateur dans la base de données
6. Connection de l'utilisateur
7. Création du compte sécurisé



6.7 UC.7 : Rejoindre un serveur de chat sans authentification

UC.7 : Rejoindre un serveur de chat sans authentification	
Composants mis en jeu	User / Server / Client
Intervenants	LD
Processus de mise en œuvre	

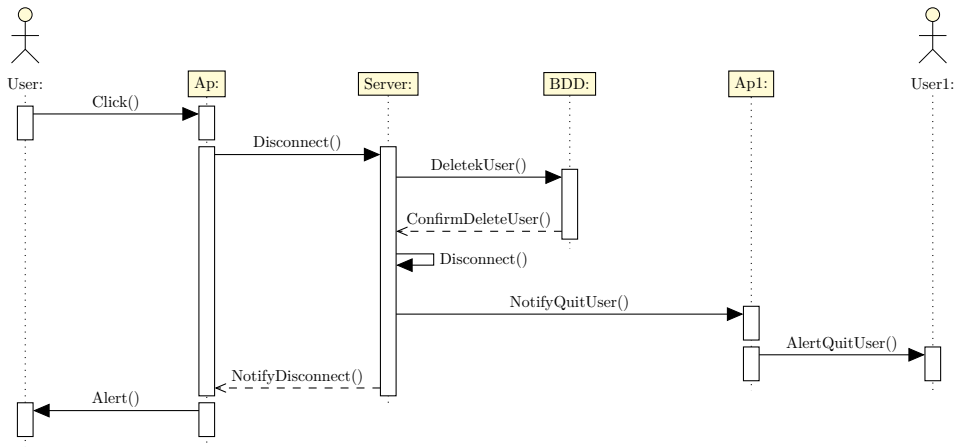
1. L'utilisateur fait une demande de connexion sur le Client
2. Le Client envoi la demande de connexion au Server
3. Le Server vérifie si le pseudonyme est disponible et l'enregistre dans la base de données
4. Le Server connecte l'User dans le salon général



6.8 UC.8 : Quitter un serveur de chat

UC.8 : Quitter un serveur de chat	
Composants mis en jeu	User / Server / Client
Intervenants	BDD
Processus de mise en œuvre	

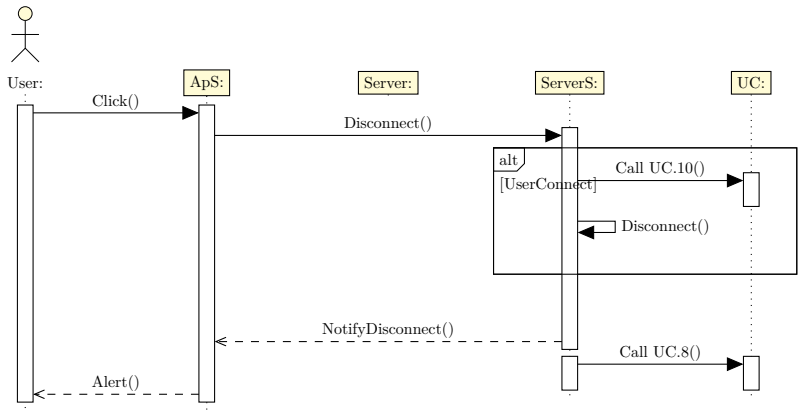
1. Demande de déconnexion de l'User dans le Client
2. Demande de déconnexion du Client au Server
3. Suppression de l'User dans la BDD par le Server S
4. Déconnexion de l'utilisateur
5. Notification à l'User



6.9 UC.9 : Quitter un serveur de chat sécurisé

UC.9 : Quitter un serveur de chat sécurisé	
Composants mis en jeu	User S / Server S / Client
Intervenants	Ap
Processus de mise en œuvre	

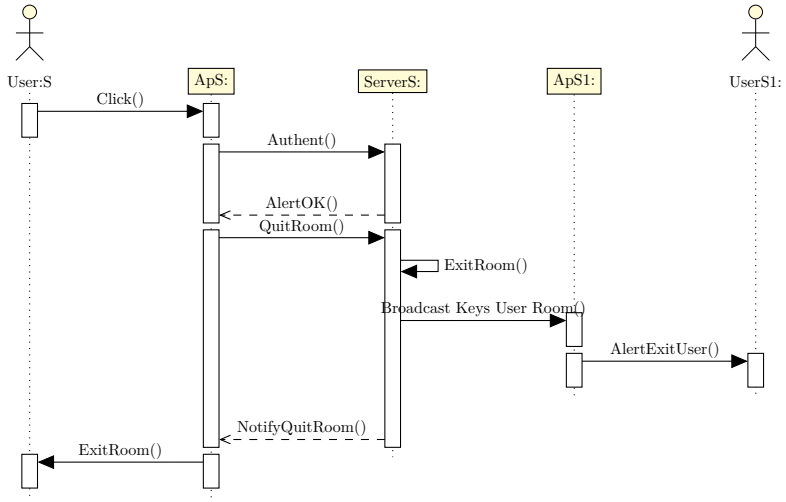
1. Demande de déconnexion de l'User dans le Client
2. Demande de déconnection du Client au Server
3. Suppression de l'User dans la BDD par le Server
4. Déconnexion de l'utilisateur
5. Notification à l'User



6.10 UC.10 : Quitter un salon privé

UC.10 : Quitter un salon privé	
Composants mis en jeu	User S / Server S / Client S
Intervenants	
Processus de mise en œuvre	

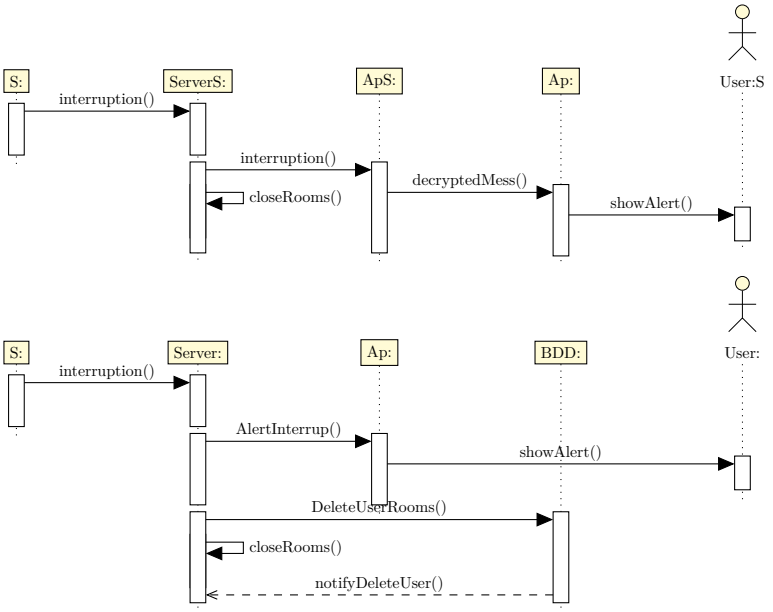
1. Demande de déconnexion par l'User sur le Client
2. Transmission par le Client de la demande au Server
3. Déconnexion du salon privé
4. Notification de déconnexion à l'User



6.11 UC.11 : Deconnexion servers

UC.11 : Deconnexion servers	
Composants mis en jeu	Users / User S / Server / Server S / Admin
Intervenants	client s
Processus de mise en œuvre	

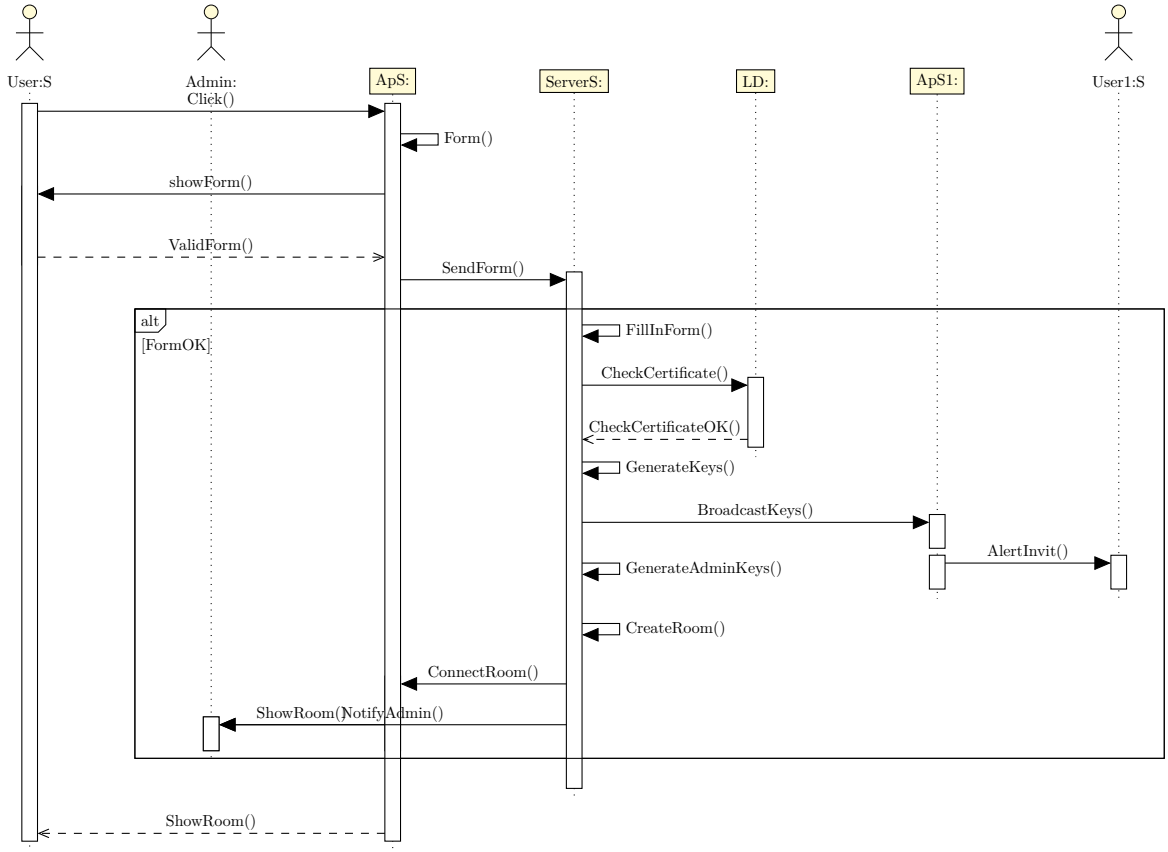
1. Le Server S ou le Server reçoivent un signal de coupure
2. Le Server S et le Server déconnectent tous les utilisateurs de tous les salons créés
3. Alerte de déconnexion auprès des utilisateurs
4. Suppression des salons présent sur les serveurs



6.12 UC.12 : Créer un salon privé

UC.12 : Créer un salon privé	
Composants mis en jeu	User S / Server S / Client S / Server
Intervenants	LD
Processus de mise en œuvre	

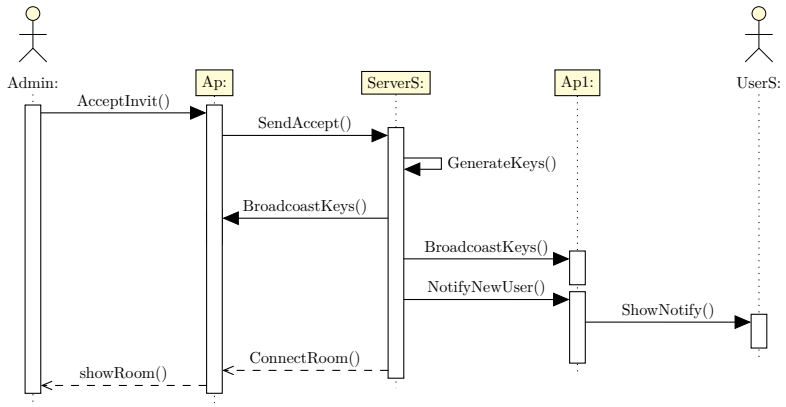
1. Demande de création par l'User sur le Client
2. Envoi de la demande au Server S
3. Vérification verification du formulaire par l'AR
4. Envoi de demande de clé à l'AC
5. génération des clés par l'AC
6. Envoi des clés générés au Client S
7. Demande de création par le Client S au Server S
8. Création du salon et connexion de l'User S



6.13 UC.13 : Rejoindre un salon privé

UC.13 : Rejoindre un salon privé	
Composants mis en jeu	User S / Server S
Intervenants	
Processus de mise en œuvre	

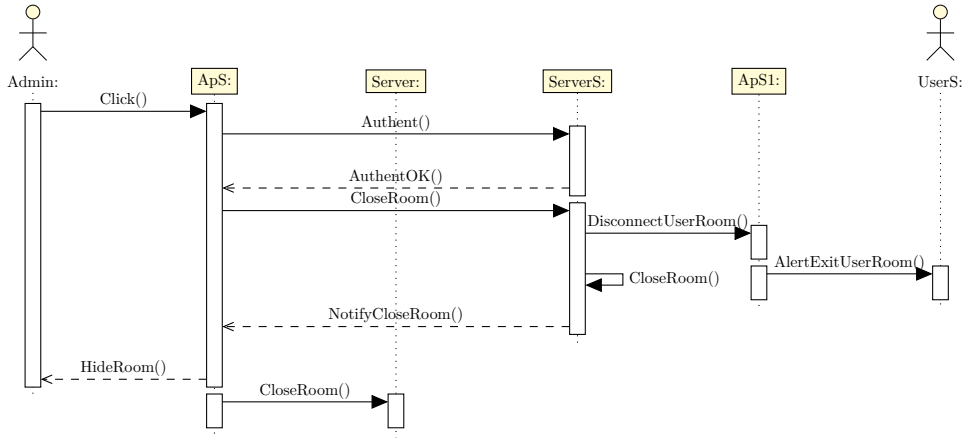
1. Admin accepte la demande de rejoindre le salon
2. La demande est prise en compte et le Server S génère de nouvelles clés pour le salon
3. Les nouvelles clés sont diffusées à tous les utilisateurs du salon
4. Un message de notification d'un nouvel utilisateur est envoyé aux User S déjà présents
5. Les messages du salon sont visibles par le nouvel User S



6.14 UC.14 : Fermeture d'un salon privé

UC.14 : Fermeture d'un salon privé	
Composants mis en jeu	Admin / Server S / Client S
Intervenants	Server
Processus de mise en œuvre	

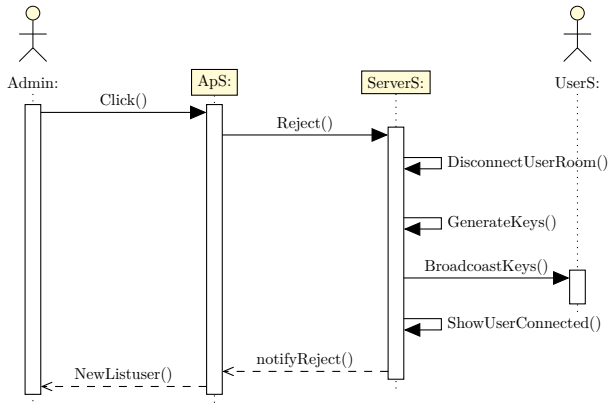
1. Demande de cloture d'un salon privé
2. Envoi de la demande de cloture au Server S
3. Déconnexion des User s présent sur le salon
4. Fermeture du salon
5. Notification à l'User



6.15 UC.15 : Exclure un utilisateur securise d'un salon prive

UC.15 : Exclure un utilisateur securise d'un salon prive	
Composants mis en jeu	Admin / Server S / Client S
Intervenants	
Processus de mise en œuvre	

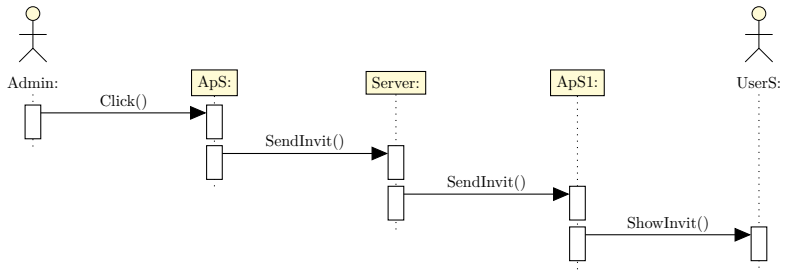
1. Admin demande d'exclusion de l'User sur le Client S
2. Transmission de la demande par le Client S au Server S
3. Déconnexion de l'User S présent dans le salon
4. Notificatron au Client S
5. Demande de clé par le Client S
6. Génération et envoie des clés par l'AC a Client S
7. Connexion du nouvel utilisateur



6.16 UC.16 : Invitation user s dans un salon

UC.16 : Invitation user s dans un salon	
Composants mis en jeu	Admin / Server / User S / Client S
Intervenants	
Processus de mise en œuvre	

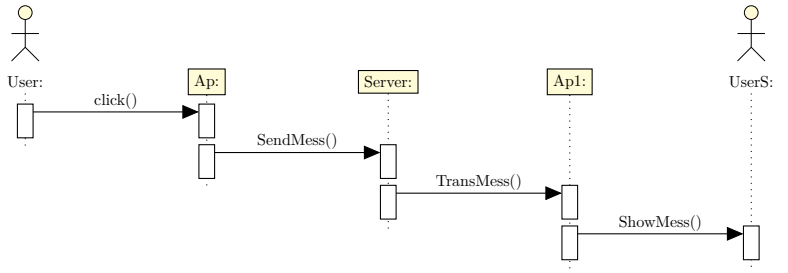
1. Demande d'invitation de l'User S sur le Client S
2. Envoi de demande par le Client S au Server
3. Transmission du message du Server au Client S de l'User destinataire
4. Acceptation de demande par l'User S destinataire



6.17 UC.17 : Envoi d'un message prive non sécurisé

UC.17 : Envoi d'un message prive non sécurisé	
Composants mis en jeu	User / User S / Client / Server
Intervenants	
Processus de mise en œuvre	

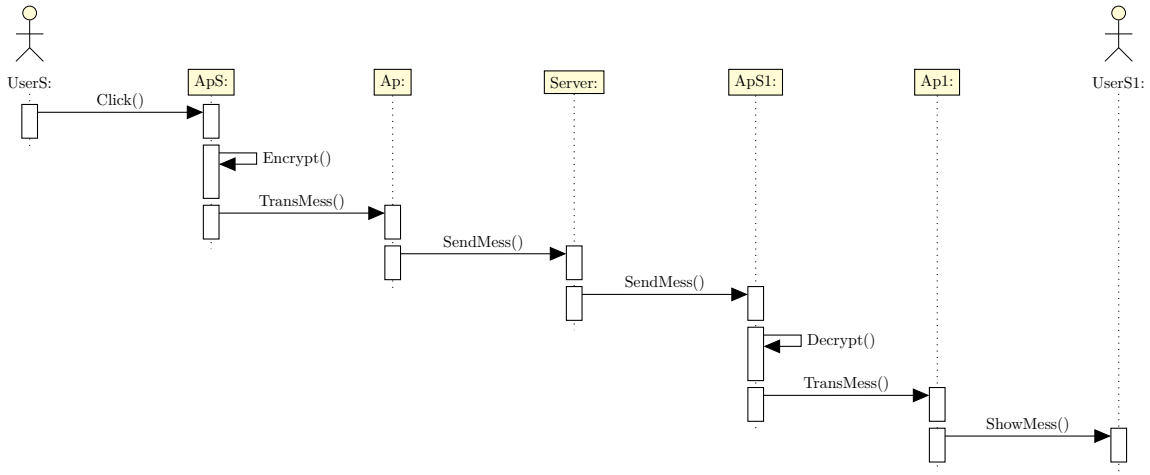
1. L'User envoie son message via le Client
2. Le client envoi le message au Server
3. Le Server se charge de transmettre le message au Client de l'User S destinataire
4. Le Client montre le message à l'utilisateur



6.18 UC.18 : Envoi d'un message prive sécurisé

UC.18 : Envoi d'un message prive sécurisé	
Composants mis en jeu	User S / Client S / Server
Intervenants	Client
Processus de mise en œuvre	

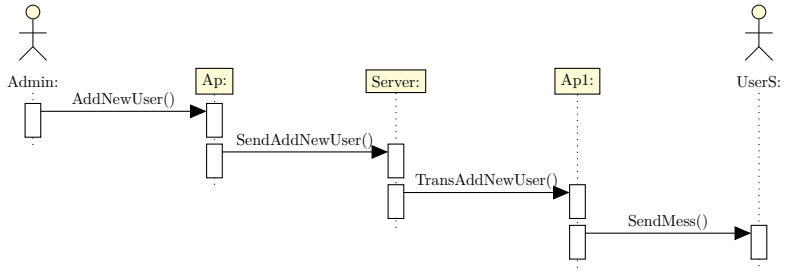
1. Chiffrement du message par Client S
2. Envoi du message via le Client au Server
3. Le Server transmet le message au Client du destinataire et déchiffrement de celui-ci



6.19 UC.19 : Demande d'ajout dans un salon privé

UC.19 : Demande d'ajout dans un salon privé	
Composants mis en jeu	User S / Admin / Server / Client
Intervenants	
Processus de mise en œuvre	

1. L'Admin effectue la demande d'ajout d'un User S
2. La demande est transmise par le Client au Server
3. Le Server transmet la demande au Client de l'User S destinataire
4. L'User S destinataire recoit une notification de la demande



7 Sur le plan cryptographique

La PKI doit s'assurer que les échanges soient sécurisés et garantir l'intégrité, l'authentification, la confidentialité et la non-répudiation lors d'échanges d'informations. Ceci est garanti grace à la signature numérique du message.

7.1 Mécanisme de vérification d'une signature

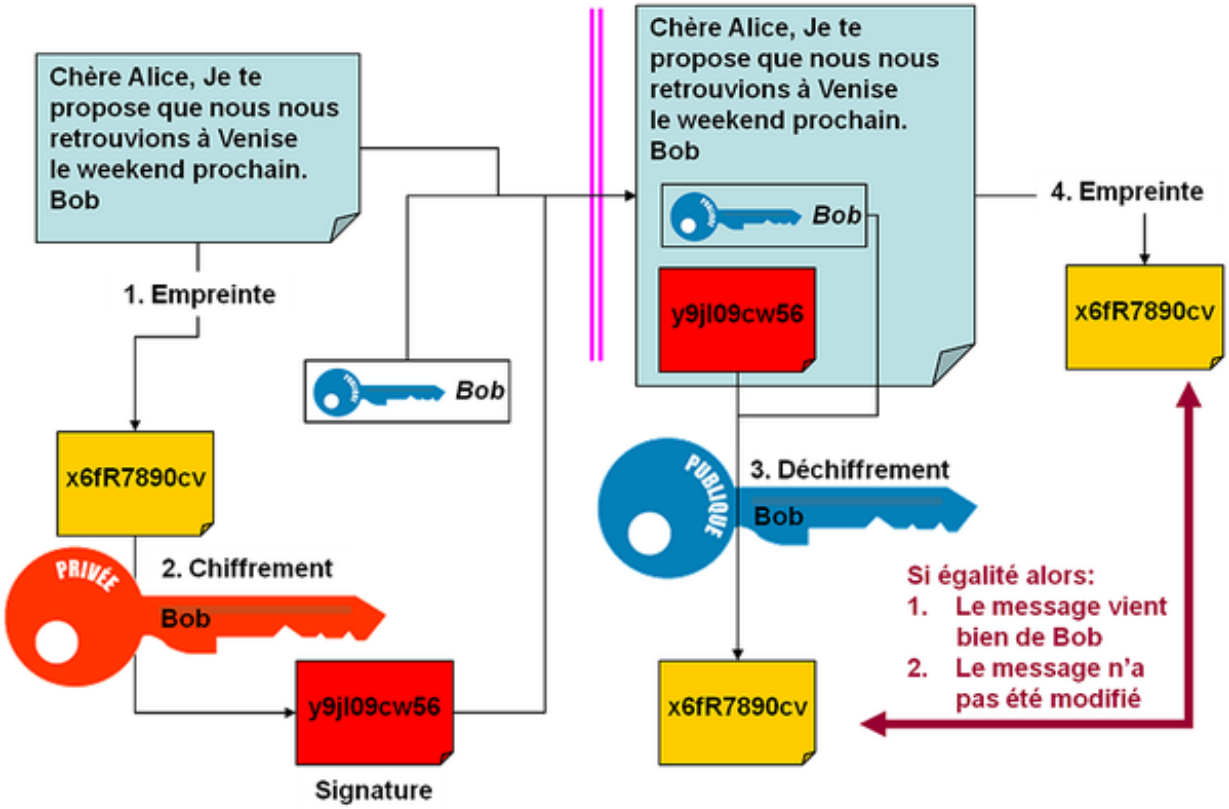


FIGURE 2 – Signature numérique