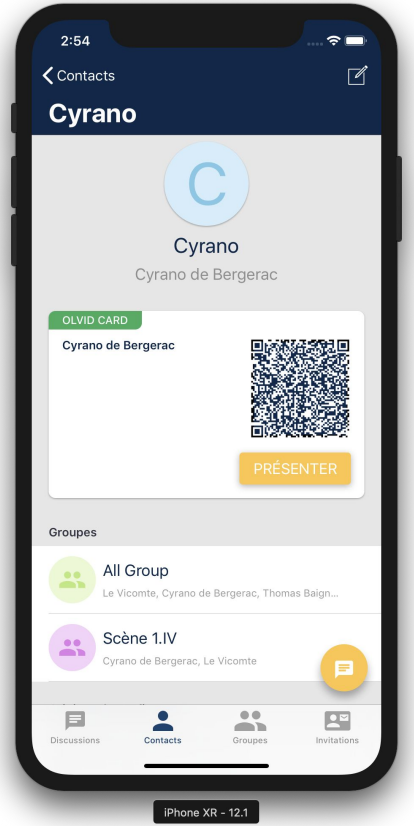
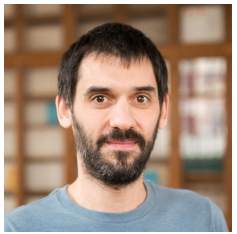
Olvid.

Construction d'une messagerie réellement sécurisée :
Quel modèle de sécurité ?
Quels moyens cryptographiques ?

OSSIR - 9 avril 2019



Qui suis-je ?



Matthieu Finiasz

CTO @ Olvid.

Docteur en cryptographie (ENS - INRIA)

Co-fondateur CybelAngel

Olvid.

La messagerie mobile réellement sécurisée

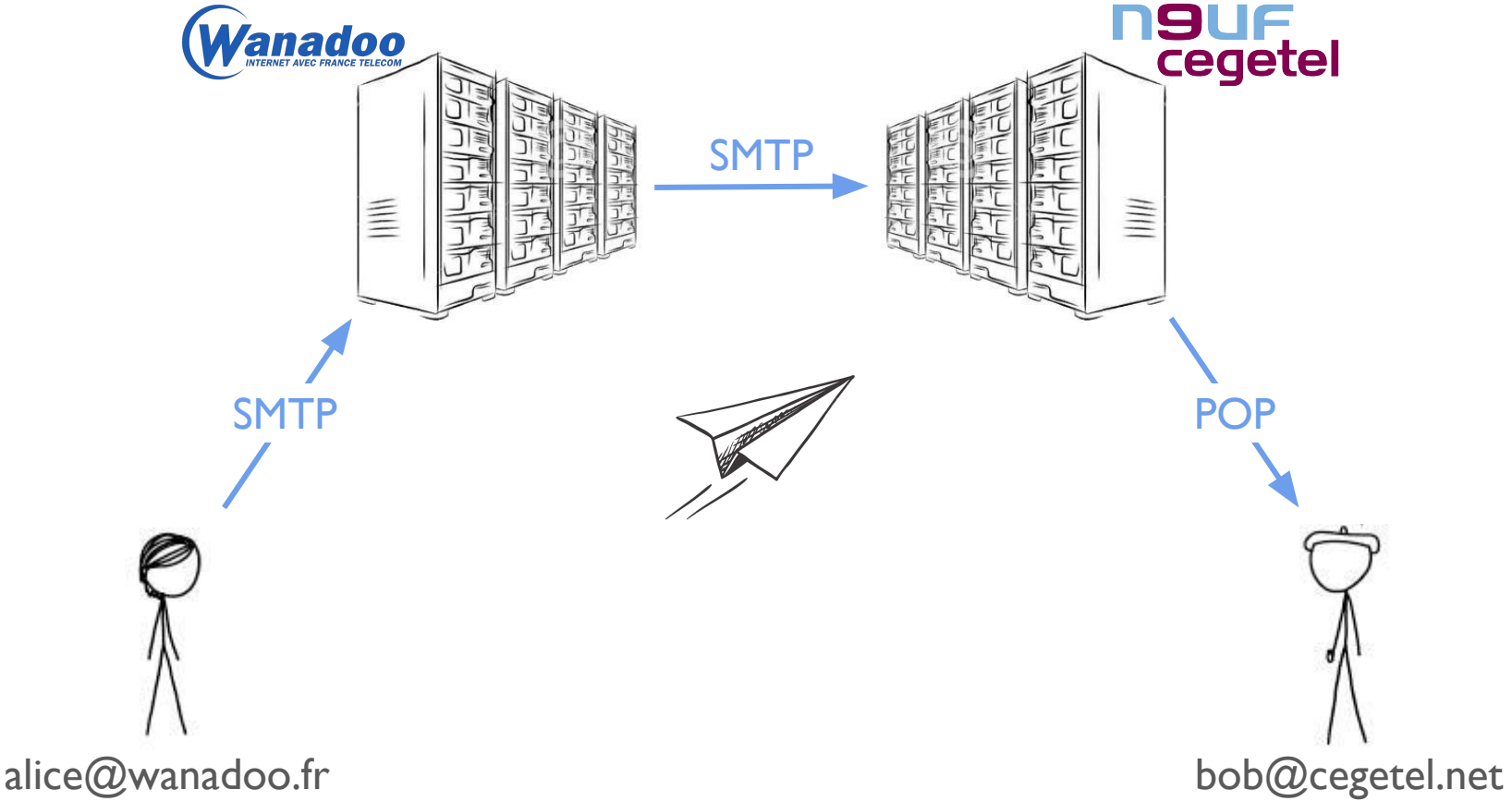
Seule la cryptographie peut garantir la sécurité totale de vos communications

Plan

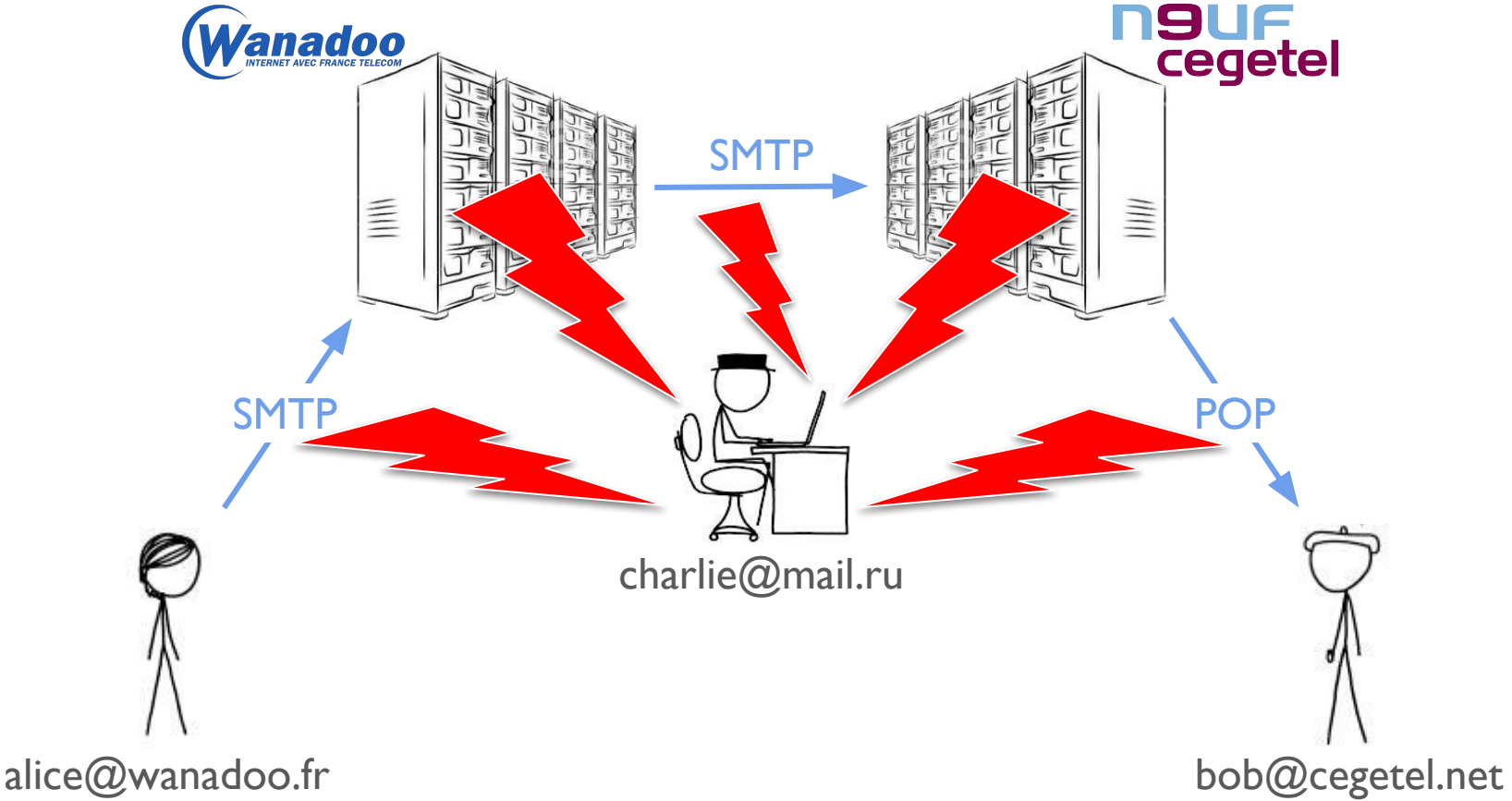
1. Historique
2. Quelles propriétés de sécurité attendre ?
3. Quel modèle de sécurité ?
4. Tour d'horizon des solutions existantes
5. Quels moyens cryptographiques pour sécuriser Olvid ?

Historique.

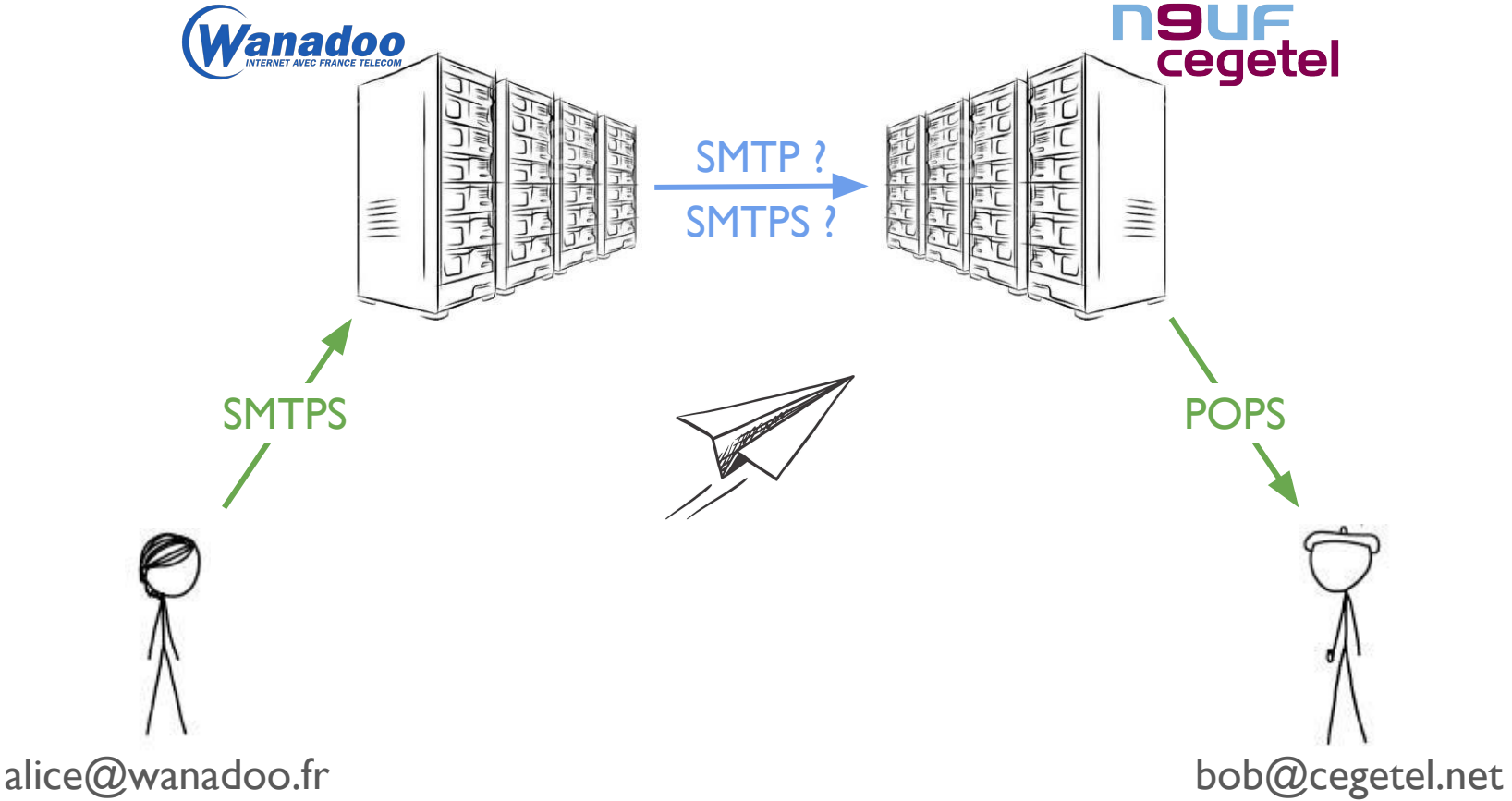
Le mail, la première messagerie.



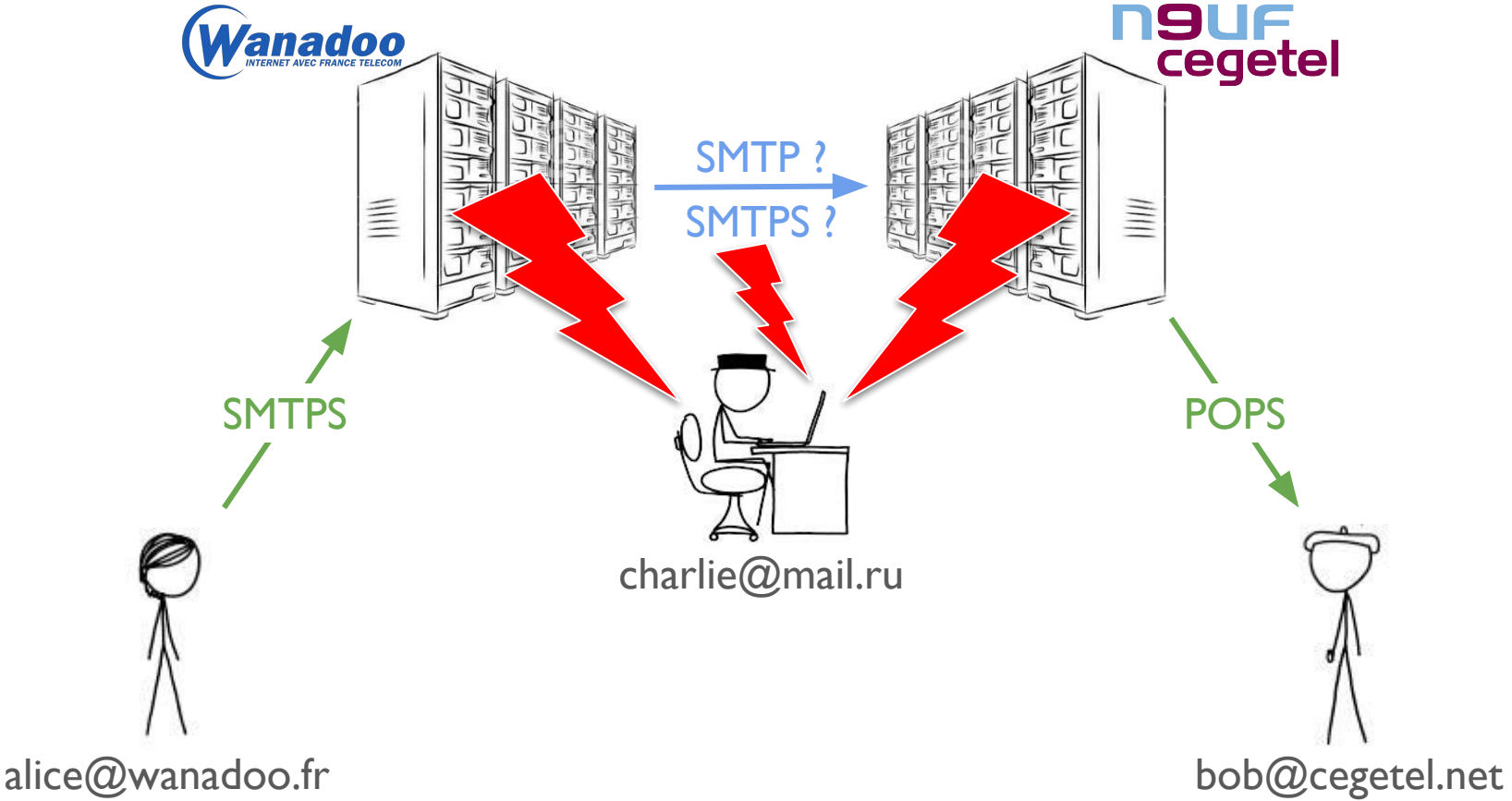
Le mail, la première messagerie.



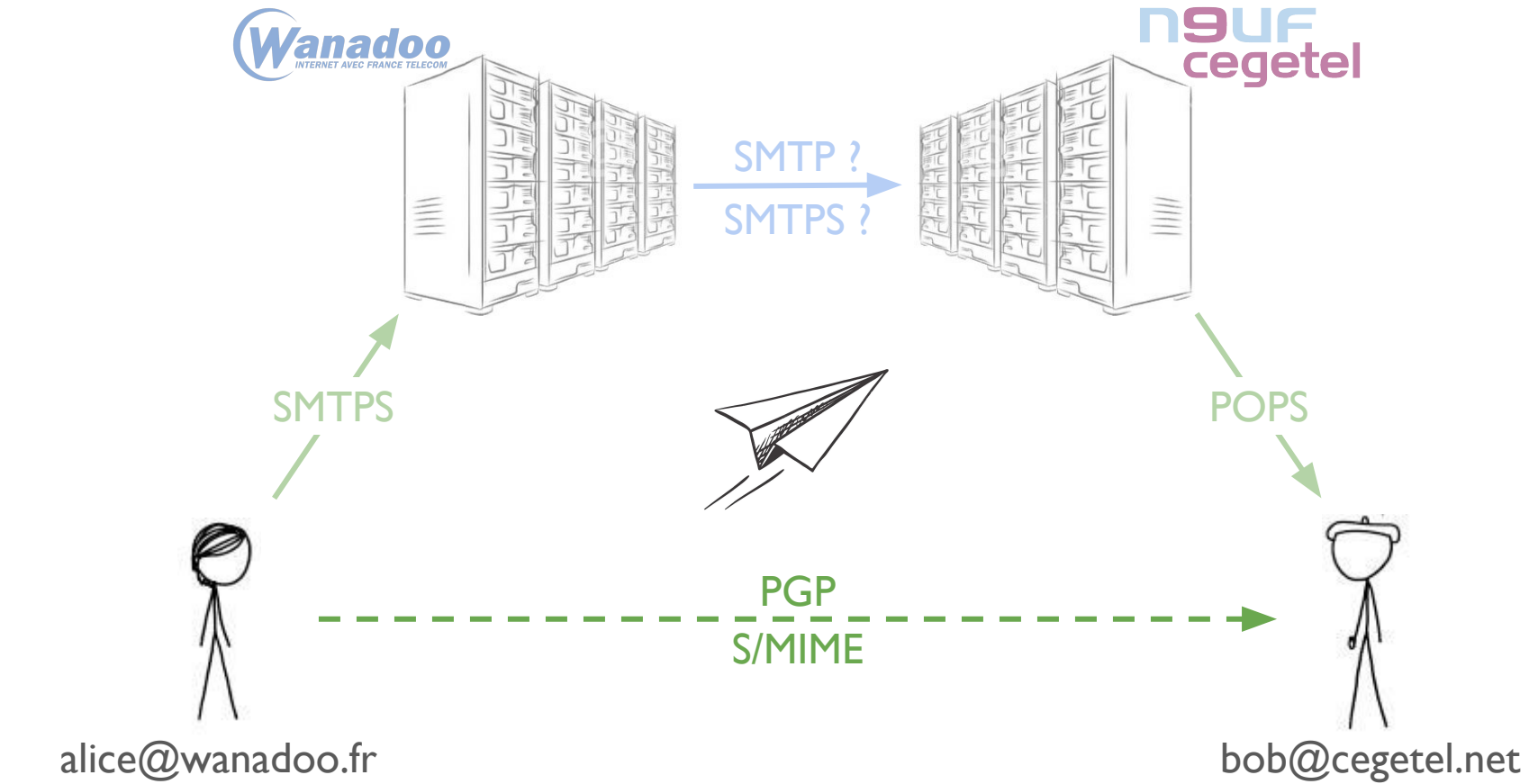
Le mail, en utilisant SSL.



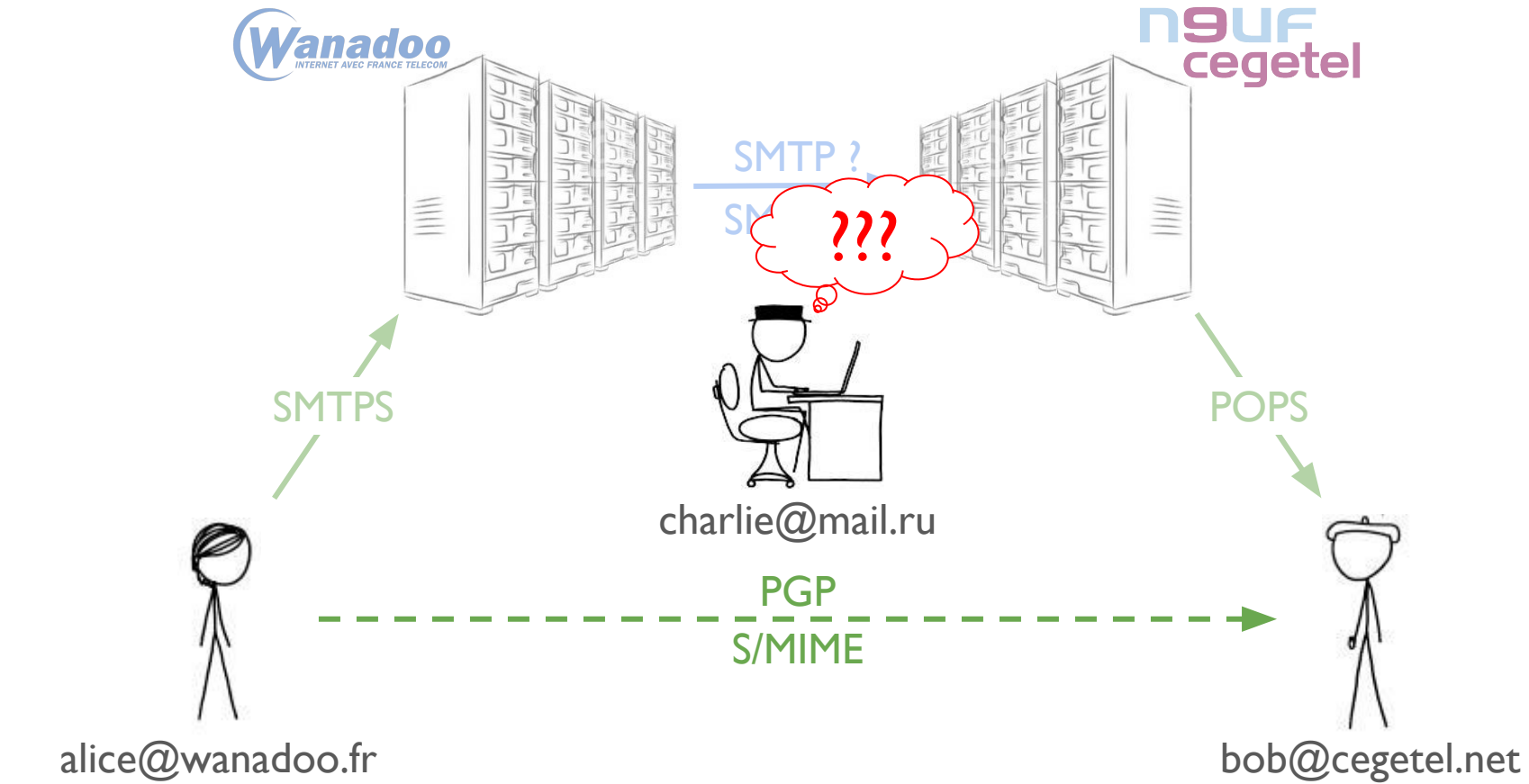
Le mail, en utilisant SSL.



Le mail, avec PGP ou S/MIME.



Le mail, avec PGP ou S/MIME.



Le mail, avec PGP ou S/MIME.

```
Return-Path: <alice@wanadoo.fr>
Received: from [10.0.101.17] (tui75-2-82-66-245-153.wanadoo.fr. [76.66.245.153])
        by smtp.cegetel.net with ESMTPSA id w125sm2216593wmw.18.2019.05.09.03.26.14
        for <bob@cegetel.net>
        (version=TLSv1/SSLv3 cipher=OTHER);
        Fri, 05 Apr 2019 03:26:15 -0700 (PDT)
Subject: Document confidentiel
References: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
To: Bob <bob@cegetel.net>
From: Alice <alice@wanadoo.fr>
X-Forwarded-Message-Id: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
Message-ID: <56F26F45.2080208@wanadoo.fr>
Date: Fri, 05 Apr 2019 11:26:13 +0200
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Thunderbird/60.6.1
MIME-Version: 1.0
In-Reply-To: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
Content-Type: multipart/mixed; boundary="------------030309080003040107080504"

This is a multi-part message in MIME format.
--------------030309080003040107080504
Content-Type: text/plain; charset=windows-1252
Content-Transfer-Encoding: 8bit

-----BEGIN PGP MESSAGE-----
Charset: windows-1252
Version: GnuPG v2

hQEMA/zpMwW7l2uOAQf/UBMBBMN0PDgs9bSEpXshUBKVXULpBsbg/M8LLnomdgTm
cs0+0HsINcY6+d5wLOTdPIVbK9iYoUzAhkfmjFya8/2Ntjldd5C7F9tsREcQjJXT
dWtCoGlQPBwp7gBRmcUlnYK0zWga9VMB782XsDJLPFclKMUNS3CmAKy0aZby7sCS
nKGb8P22wk6odCS5NTIxazvLbnLz24MCUgVbaTkksUYuhvlH0PNu+nVvg4nEdoWe
TwHNrHFnyeto2F9NjRwH6/mYj92xzJu9o/c8dyLJsjdjXZHrcvZInQcPtUDvpJbX
VGG9LX+RknNqSHjrI7bys73w8N/VWuxKBrSgbTmmYIyjoJwA420b5/O7gIujZiI0
WdHjLNWH77OHAp2dtF4ggoZCwBy4WTVcU+1SdwNqBTXI8j1whZk1nf+/SO8b7Sg2
HPgrsMTxnaUf
=isg2
-----END PGP MESSAGE-----

--------------030309080003040107080504
Content-Type: application/octet-stream;
 name="brevet end2end encryption.docx.pgp"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
 filename="brevet end2end encryption.docx.pgp"

hQEMA/zpMwW7l2uOAQf/V2zalW4esYvN2STnekSx7HSREWs8ZC752QLMIJ/6hSTEVcdaMycp
guP4bC8vBeFq5aelofgxjf+ki3XmlHY4dEPfiWMPpuaZuLcOw9cdZftsb4S6khe99z9laNS7
NyNZNPraqEy3pkzjaROvwsDXoiCm4ZtGaV5TSErCknd8X3IfcH1icMxdFoOBBOhLv/WckxC9
llcWGAXhRDEMC/hvIsknnH5RhEtYJDaEfK56CVmxl3BQT9c7/PRzda8EFeEn6z/i3JUquir3
TEGvXaiOPwt0W+l/wla7g8lPf6SdEM+DY8xWbEAlpvNfofG4VaPr5Py1I+QVmiOHo/FxZJnO
```

Le mail, avec PGP ou S/MIME.

```
Return-Path: <alice@wanadoo.fr>
Received: from [10.0.101.17] (tui75-2-82-66-245-153.wanadoo.fr. [76.66.245.153])
        by smtp.cegetel.net with ESMTPSA id w125sm2216593wmw.18.2019.05.09.03.26.14
        for <bob@cegetel.net>
        (version=TLSv1/SSLv3 cipher=OTHER);
        Fri, 05 Apr 2019 03:26:15 -0700 (PDT)
Subject: Document confidentiel
References: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
To: Bob <bob@cegetel.net>
From: Alice <alice@wanadoo.fr>
X-Forwarded-Message-Id: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
Message-ID: <56F26F45.2080208@wanadoo.fr>
Date: Fri, 05 Apr 2019 11:26:13 +0200
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Thunderbird/60.6.1
MIME-Version: 1.0
In-Reply-To: <3C0A69BF-D444-4C2F-9E61-D06D43503D6A@cegetel.net>
Content-Type: multipart/mixed; boundary="------------030309080003040107080504"

This is a multi-part message in MIME format.
--------------030309080003040107080504
Content-Type: text/plain; charset=windows-1252
Content-Transfer-Encoding: 8bit

-----BEGIN PGP MESSAGE-----
Charset: windows-1252
Version: GnuPG v2

hQEMA/zpMwW7l2uOAQf/UBMBBMN0PDgs9bSEpXshUBKVXULpBsbg/M8LLnomdgTm
cs0+0HsINcY6+d5wLOTdPIVbK9iYoUzAhkfmjFya8/2Ntjldd5C7F9tsREcQjJXT
dWtCoG1QPBwp7gBRmcU1nYK0zWga9VMB782XsDJLPFclKMUNS3CmAKy0aZby7sCS
nKGb8P22wk6odCS5NTIxazvLbnLz24MCUgVbaTkksUYuhvlH0PNu+nVvg4nEdoWe
TwHNrHFnyeto2F9NjRwH6/mYj92xzJu9o/c8dyLJsjdjXZHrcvZInQcPtUDvpJbX
VGG9LX+RknNqSHjrI7bys73w8N/VWuxKBrSgbTmmYIyjoJwA420b5/O7gIujZiI0
WdHjLNWH77OHAp2dtF4ggoZCwBy4WTVcU+1SdwNqBTXI8j1whZk1nf+/SO8b7Sg2
HPgrsMTxnaUf
=isg2
-----END PGP MESSAGE-----

--------------030309080003040107080504
Content-Type: application/octet-stream;
 name="brevet end2end encryption.docx.pgp"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
 filename="brevet end2end encryption.docx.pgp"

hQEMA/zpMwW7l2uOAQf/vZ2aIW4esYVN2STnekSx7HSREWs8ZC752QLMIJ/6hSTEVcdaMycp
guP4bC8vBeFq5aelofgxjf+ki3XmlHY4dEPfiWMPpuaZuLcOw9cdZftsb4S6khe99z91aNS7
NyNZNPraqEy3pkzjaROvwsDXoiCm4ZtGaV5TSErCknd8X3IfcH1icMxdFoOBBOhLv/WckxC9
11cWGAXhRDEMC/hvIsknnH5RhEtYJDaEfK56CVmxl3BQT9c7/PRzda8EfeEn6z/i3JUquir3
TEGvXaiOPwt0W+l/wla7g8lPf6SdEM+DY8xWbEAlpvNfofG4VaPr5Py1I+QVmiOHo/FxZJnO
```

```
Subject: Document confidentiel
To: Bob <bob@cegetel.net>
From: Alice <alice@wanadoo.fr>
```

```
Date: Fri, 05 Apr 2019 11:26:13 +0200
```

```
filename="brevet end2end encryption.docx.pgp"
```

Quelles propriétés de sécurité attendre ?

Quelles propriétés de sécurité ?

Reproduire la sécurité d'une **conversation à huis clos**

Dans un **monde numérique**

Quelles propriétés de sécurité ?

Reproduire la sécurité d'une **conversation à huis clos**

- Aucun tiers n'entend ce qui se dit
- Chaque interlocuteur sait précisément à qui il s'adresse
- La discussion ne laisse aucune trace

Dans un **monde numérique**

Quelles propriétés de sécurité ?

Reproduire la sécurité d'une **conversation à huis clos**

- Aucun tiers n'entend ce qui se dit
- Chaque interlocuteur sait précisément à qui il s'adresse
- La discussion ne laisse aucune trace

Dans un **monde numérique**

- Communications asynchrones
- Pièces jointes de toute nature
- Instantanéité, malgré la distance

3 Piliers

Authentification

Chiffrement des données

Chiffrement des métadonnées

3 Piliers

Authentification

Chiffrement des données

Chiffrement des métadonnées

1 Principe

« Minimal Disclosure »

Toujours divulguer le minimum d'information au minimum de personnes.

3 Piliers

Authentification

Chiffrement des données

Chiffrement des métadonnées

1 Principe

« Minimal Disclosure »

Toujours divulguer le minimum d'information au minimum de personnes.

2 Contraintes

Efficacité

Côté serveur, *côté client* et en nombre d' échanges

&

Utilisabilité

Le moins de contraintes sur l'utilisateur

Quel modèle de sécurité ?

Qu’est-ce qu’un modèle de sécurité cryptographique ?

Des hypothèses

Éléments externes sur lesquels un protocole cryptographique peut s’appuyer

- Tiers de confiance
- Source d’aléa publique
- Oracle aléatoire
- etc.

Qu'est-ce qu'un modèle de sécurité cryptographique ?

Des hypothèses

Éléments externes sur lesquels un protocole cryptographique peut s'appuyer

- Tiers de confiance
- Source d'aléa publique
- Oracle aléatoire
- etc.

Une capacité d'attaque

Ensemble des pouvoirs de l'adversaire

- Observation d'un canal
- Modifications sur ce canal
- Démarrage d'instances du protocole
- Oracle de chiffrement/déchiffrement
- Puissance de calcul
- L'adversaire veut rester « discret »
- etc.

Qu'est-ce qu'un modèle de sécurité cryptographique ?

Des hypothèses

Éléments externes sur lesquels un protocole cryptographique peut s'appuyer

- Tiers de confiance
- Source d'aléa publique
- Oracle aléatoire
- etc.

Une capacité d'attaque

Ensemble des pouvoirs de l'adversaire

- Observation d'un canal
- Modifications sur ce canal
- Démarrage d'instances du protocole
- Oracle de chiffrement/déchiffrement
- Puissance de calcul
- L'adversaire veut rester « discret »
- etc.

Un but

Ce que l'adversaire cherche à obtenir ou à provoquer

- Déchiffrement d'un message
- Modification aléatoire d'un message
- Impersonification
- Déni de service
- etc.

Le modèle de base.

Bellare-Rogaway

(Dolev-Yao étendu)

L'adversaire **contrôle tous les intermédiaires**, et peut **démarrer des protocoles**.

Maîtrise du réseau

Lire les paquets, les modifier, les supprimer, les réordonner, en insérer, etc.

Discrétion

L'adversaire ne veut pas être détecté
Modèle « Honnête mais curieux » étendu

Besoin de tiers de confiance ?

La plupart des solutions reposent sur des tiers de confiance pour l'authentification :

- S/MIME → autorités de certification
- PGP → système de signature par les pairs
- WhatsApp → annuaire central chez WhatsApp

Dans un huis clos, c'est sur la **confiance entre interlocuteurs** que repose la sécurité

Besoin de tiers de confiance ?

La plupart des solutions reposent sur des tiers de confiance pour l'authentification :

- S/MIME → autorités de certification
- PGP → système de signature par les pairs
- WhatsApp → annuaire central chez WhatsApp

Dans un huis clos, c'est sur la **confiance entre interlocuteurs** que repose la sécurité

Modèle de confiance

On ne demande de faire confiance à personne d'autre que ceux en qui on a déjà confiance
→ ce n'est pas à la messagerie **d'imposer des tiers de confiance**

L’utilisateur est son propre adversaire.

L’utilisateur n’est pas un expert

Chaque choix utilisateur peut être fait de travers, car il ne comprend pas les implications.

→ la sécurité ne peut pas reposer sur des choix utilisateurs

Pas de mot de passe

- Mal choisi dans 50% des cas
- Uniquement pour de la sur-sécurité

« Security by design »

On ne laisse aucun choix de sécurité à l’utilisateur, ou alors tous les choix sont bons.

Modèle de sécurité

L’utilisateur est son propre adversaire
Chacun de ses choix est mauvais

Sécurité persistante.

Le vol de device est un risque réel pour une application mobile :

- On ne peut pas considérer l'OS comme une sécurité suffisante
  → donne accès à tout ce qui est sur le téléphone
- Cela ne doit donner accès à rien de plus
  → les contacts et messages effacés doivent l'être définitivement

Les clés long terme **ne servent jamais à chiffrer** des données sensibles/du contenu

Sécurité persistante.

Le vol de device est un risque réel pour une application mobile :

- On ne peut pas considérer l'OS comme une sécurité suffisante
  → donne accès à tout ce qui est sur le téléphone
- Cela ne doit donner accès à rien de plus
  → les contacts et messages effacés doivent l'être définitivement

Les clés long terme **ne servent jamais à chiffrer** des données sensibles/du contenu

Modèle de sécurité des clés long terme

L'adversaire peut voler des clés long terme sans atteinte à la sécurité des échanges passés
→ propriété de **« forward secrecy »**

Multi-utilisateurs et multi-instances.

Les modèles cryptographiques considèrent Alice et Bob, isolés du reste du monde :

- Une messagerie a des millions d'utilisateurs
- L'adversaire n'en cible pas un en particulier
  → modèle d'attaque **« 1 parmi N »**
- Chaque utilisateur est en contact avec des dizaines de contacts
  → modèle d'attaque **multi-instances**

Protocoles sans humain

Des milliers d'instances en parallèle
Sur des milliers d'utilisateurs

Protocoles avec humain

Quelques instances en parallèle
Sur quelques utilisateurs

Le « bon » modèle de sécurité.

Modèle de sécurité

Comme dans un huis-clos, tout le monde extérieur est hostile mais souhaite rester invisible/indétectable

Hypothèses

- Les serveurs font (à peu près) ce qu'on leur demande
- Les utilisateurs se connaissent et se font confiance
- Le device est « sain » au moment des échanges

Capacité d'attaque

Les serveurs :

- observent les méta-données
- font des analyses statistiques
- fond des copies
- modifient certains messages
- tentent du MitM
- etc.

But

Récupérer une information quelconque sur les échanges :

- qui parle à qui ?
- à quelle fréquence ?
- pour dire quoi ?

Tour d’horizon
des solutions existantes.

Chiffrement du mail : S/MIME et PGP.

Intégré au client mail

- Risque d'erreur humaine
- Risque de bug (attaque EFAIL)
- Très loin du « privacy by design »

Signature optionnelle

- Normal pour la répudiabilité
- Problématique pour authentifier l'envoyeur
- Problème insoluble avec des clés long terme seulement

S/MIME

- Confiance en des autorités de certification
- Certaines peu fiables
- Difficile de contrôler auxquelles on fait confiance

PGP

- Possibilité de ne pas vérifier les clés (fingerprint)
  → plus aucune sécurité
- Peu d'utilisateurs se signent leurs clés

Usages avec IMAP

- Les données restent (chiffrées) sur le serveur
- Besoin de pouvoir déchiffrer dans le future
  → pas de « forward secrecy »

Métadonnées en clair

- Le protocole SMTP a besoin de métadonnées en clair
- Elles sont facilement modifiables/forgeables
- Ne sont pas toutes signées

Chiffrement du mail : S/MIME et PGP.

Intégré au client mail

- Risque d'erreur humaine
- Risque de bug (attaque EFAIL)
- Très loin du « privacy by design »

Signature optionnelle

- Normal pour la répudiabilité
- Problématique pour authentifier l'envoyeur

S/MIME

- Confiance en des autorités de certification
- Certaines peu fiables
- contrôler
- on fait confiance

Le mail ne sera jamais sûr...

PGP

- Possibilité de ne pas vérifier les clés (fingerprint)
  → plus aucune sécurité
- Peu d'utilisateurs se signent leurs clés

- Les données restent (chiffrées) sur le serveur
- Besoin de pouvoir déchiffrer dans le future
  → pas de « forward secrecy »

nées en clair

- Le protocole SMTP a besoin de métadonnées en clair
- Elles sont facilement modifiables/forgeables
- Ne sont pas toutes signées

Messageries instantanées modernes (eg. Signal).

Chiffrement end-2-end

- Messageries créées pour corriger les problèmes du mail → chiffrement avec toutes les bonnes propriétés
- Tout leur marketing est centré sur le chiffrement

Authentification faillible

- La distribution des clés passe par un serveur central → un hack et la sécurité s'effondre
- MitM facile (et indétectable) pour un serveur malveillant

Aucun anonymat

- On commence par « uploader » son carnet d'adresse
- Le serveur sait qui parle à qui, quand et à quelle fréquence

Et plein de petites spécificités…

WhatsApp

- Demande de re-chiffrement sélectif
- Analyse des contenus sur le device

Threema

- Pas de forward secrecy end-2-end

Telegram

- Chiffrement optionnel (et exotique)
- Pas de chiffrement pour les groupes

Wire

- Multi-device utilisant plusieurs clés
  → pas de MitM, ajouter un device suffit

Quels moyens cryptographiques pour sécuriser Olvid ?

Quels moyens cryptographiques pour sécuriser Olvid ?

On repart de zéro en appliquant les 3 principes précédemment énoncés.

« Minimal Disclosure »

Toujours divulguer le minimum d'information au minimum de personnes.

Efficacité

Côté serveur, *côté client* et en nombre d' échanges

&

Utilisabilité

Le moins de contraintes sur l'utilisateur

Principes généraux.

- Pour échanger avec quelqu'un, il faut récupérer sa clé et **l'authentifier**
- Cette clé permet d'établir un **« canal évolutif »** avec toutes les propriétés voulues
- Les échanges sur ce canal sont entièrement **chiffrés end-2-end**
  → seule la clé publique du destinataire est en clair
- Les messages déposés sur le serveur le sont de façon **anonyme**
- À réception d'un message, une **notification push** (iOS ou Android) est envoyée
- Le **destinataire s'authentifie** auprès du serveur pour télécharger les messages

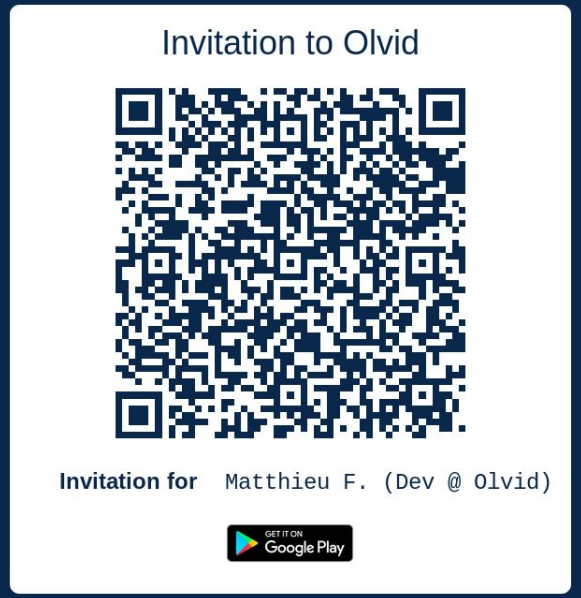
Échange de clés.

Envoi d'une URL (ou code QR), **non-confidentielle** :

https://invitation.olvid.io/AwAAAIEAAAAAXmh0dHBzOi8vc2VydmVyLmRldi5vbHZpZC5pbwAAyCJAgWak7cRQdAJUgS2Ly-qa6MossL52cpKE83u1G58AbkPFTOXKixt-sDhvY6GNrFBI8SzedVV4ktovqdHspAoAAAAAGU1hdHRoaWV1IEYuIChEZXYgQCBPbHZpZCk

Suivi d'un **protocole de SAS** (Short Authentication String) :

- Diffie-Hellman authentifié via les clés long terme
- Code PIN de vérification à échanger sur **un canal authentique**
  - Les utilisateurs s'appellent ou se voient
  - Ils échangent deux fois 4 chiffres
- **Les deux clés sont authentifiées** et liées à leur propriétaire



Échange de clés.

Envoi d'une URL (ou code QR), **non-confidentielle** :

https://invitatio
LmRldi5vbHZp
pKE83u1G58A
AAAAAGU1hc

Suivi d'un **protoc**

- Diffie-Helln
- Code PIN
  - Les ut
  - Ils éch
- **Les deux clés sont authentifiées** et liées à leur propriétaire

Ce que permet la cryptographie ?

- Transformer un **canal authentique** en **canal sécurisé**
  → Sans canal authentique, la cryptographie ne sert à rien !
- On veut utiliser le canal authentique le moins possible
  → 4 chiffres, une et une seule fois
- Moins que ça ? Ce n'est pas possible…

Olvid.

Olvid

F. (Dev @ Olvid)

GET IT ON Google Play

Chiffrement des messages.

Deux types de chiffrement :

Asymétrique
(clé long terme)

- Uniquement pour les protocoles cryptographiques
- Permet d'établir le canal Oblivious

Symétrique
(canal Oblivious)

- Clés à usage unique
  → avec double ratcheting
- Un préfixe aléatoire pour identifier la clé
- Chiffrement authentifié

Dans les deux cas, les chiffrés ont la même forme (ils sont indistinguables) :

- Un en-tête contenant une clé symétrique chiffrée (key-wrapping)
- Un message (d'application ou de protocole) et des pièces jointes chiffrés
- La clé publique du destinataire

Forward secrecy de l'anonymat.

On n'a pas de forward secrecy pour ce qui est chiffré avec la clé long terme.

Peut-on avoir une forward secrecy de l'anonymat ?

Forward secrecy de l'anonymat.

On n'a pas de forward secrecy pour ce qui est chiffré avec la clé long terme.

Peut-on avoir une forward secrecy de l'anonymat ?

OUI

- Ne jamais envoyer son identité sur un canal asymétrique
- Utiliser des clés éphémères pour l' échange de clés

Forward secrecy de l'anonymat.

On n'a pas de forward secrecy pour ce qui est chiffré avec la clé long terme.

Peut-on avoir une forward secrecy de l'anonymat ?

OUI

- Ne jamais envoyer son identité sur un canal asymétrique
- Utiliser des clés éphémères pour l' échange de clés

NON

- Cela ne peut pas être efficace
- Impossibilité d'avoir des notifications push sur la clé éphémère (lien entre les identités)
- On se retrouve à faire du polling…

Forward secrecy de l'anonymat.

On n'a pas de forward secrecy pour ce qui est chiffré avec la clé long terme.

Peut-on avoir une forward secrecy de l'anonymat ?

OUI

- Ne jamais envoyer son identité sur un canal asymétrique
- Utiliser des clés éphémères pour l' échange de clés

NON

- Cela ne peut pas être efficace
- Impossibilité d'avoir des notifications push sur la clé éphémère (lien entre les identités)
- On se retrouve à faire du polling…

En pratique ?

On ne perd pas réellement de sécurité : en cas de vol de device, l'anonymat est forcément compromis (carnet d'adresse, autres applications, etc.)

Envoi de message anonyme.

- Le **destinataire doit être connu** pour une distribution efficace
- L'anonymat impose que **l'expéditeur soit anonyme**
  → pas d'authentification pour l'envoi de messages
- Pour lutter contre les **DDoS**, on demande une preuve de calcul

Envoi de message anonyme.

- Le **destinataire doit être connu** pour une distribution efficace
- L'anonymat impose que **l'expéditeur soit anonyme**
  → pas d'authentification pour l'envoi de messages
- Pour lutter contre les **DDoS**, on demande une preuve de calcul

Preuve de calcul

- Le serveur donne un challenge
  → génération du challenge facile
- Le device calcul la réponse
  → calcul de la réponse complexe
- Le serveur n'accepte le message que s'il est accompagné d'une preuve de calcul
  → vérification instantanée

Principes

- Trouver un XOR de colonnes d'une matrice égal à une certaine somme
- La matrice est générée à partir d'une graine et d'un PRNG
- Calcul avec des accès mémoire aléatoires
  → supercalculateur et smartphone sont « à armes égales »

Notifications push.

Élément indispensable pour l'instantanéité

- un challenge à implémenter : iOS et Android prévoient un contenu en clair
- un risque pour la sécurité : un serveur/adversaire de plus

De quelles informations Apple/Google ont-ils besoin ?

Notifications push.

Élément indispensable pour l'instantanéité

- un challenge à implémenter : iOS et Android prévoient un contenu en clair
- un risque pour la sécurité : un serveur/adversaire de plus

De quelles informations Apple/Google ont-ils besoin ?

RIEN

- Juste un **« token »** de notification push fourni par l'OS

MAIS...

- Pour gérer plusieurs identités sur un device : un **identifiant aléatoire** pour que l'application fasse le lien

Apple/Google ne doivent pas pouvoir faire
le lien entre une clé publique et une identité réelle.

Authentification serveur.

Utilisation d'un protocole « zero-knowledge »

- L'utilisateur prouve au serveur qu'il connaît la clé secrète associée à sa clé publique
  → à base de signature, mais plus compliqué qu'une signature de challenge
- La clé publique est en fait une paire de clés publiques : chiffrement et signature

Un tiers ne peut pas savoir quand quelqu'un a reçu un message :

- Uniquement le serveur
- Et Apple/Google pour les notifications push

Anonymat des conversations de groupe.

L'anonymat est une notion complexe

Il ne faut pas que le serveur puisse **lier deux utilisateurs**

Conversation de groupe

- N utilisateurs reçoivent un message en même temps
- Analyse statistique des timestamps → corrélation forte entre membres d'un même groupe

Pas d'anonymat de groupe possible

Efficacité

Plutôt que d'envoyer N messages :

- un seul message
- une seule copie des pièces jointes
- N en-têtes différents

Quitte à ne pas avoir d'anonymat, autant être efficace !

Chiffrement post-quantique.

Faire du chiffrement post-quantique n'est pas utile aujourd'hui :

- Les données sont **chiffrés en symétrique**
- Les clés long terme servent uniquement pour de **l'authentification**

Il faut en revanche prévoir une migration le jour venu :

- Comme pour SSL/TLS, un type de clé est toujours apposé à une clé
- Aujourd'hui nous utilisons des courbes elliptiques
- Facile d'inclure un KEM post-quantique et une signature post-quantique
  Attention : le code QR devra référencer une URL où télécharger les clés

Venez l'essayer !

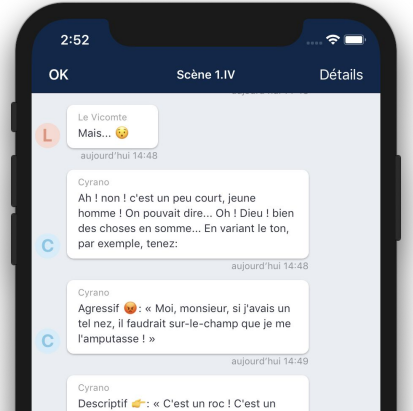
En bêta depuis

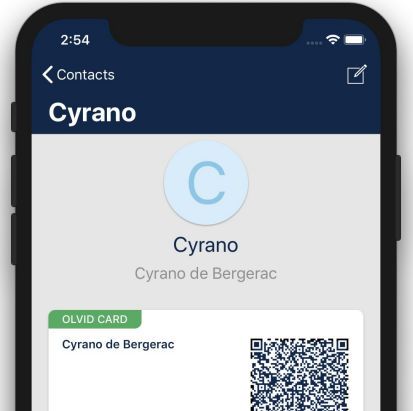
6 mois

100+

utilisateurs

Et vous ?





Cédric SYLVESTRE
OLVID, Business Development
cedric.sylvestre@olvid.io
Tél. : 06 77 58 23 41

96 bis boulevard Raspail, 75006 Paris
https://olvid.io/

Merci.