

Three small diamond-shaped icons in the top right corner: a black one, a grey one, and a white one.

Prédictions sécurisées à l'aide de calculs multipartites

Marius A. Theo LB. Maxence C. Romain N.

Sommaire

01

Introduction

Vue d'ensemble du calcul
sécurisé multipartite

02

Le projet

Présentation de nos choix
et implémentations

03

Difficultés rencontrées

Retour d'expériences sur
notre projet

04

Conclusion

Récapitulatif





01

Introduction

Vue d'ensemble du calcul sécurisé multipartite

Introduction



Qu'est-ce que le calcul multipartite sécurisé ?

- Technique cryptographique avancée.
- Préserve la confidentialité des données
- Utilisé dans les domaines de la santé, la finance et la recherche

Les principaux défis et considérations ?

- Assurer la confidentialité, l'intégrité et la disponibilité
- Performance et efficacité des calculs
- Gestion des risques liés aux adversaires semi-honnêtes et malveillants





02

Le projet

Présentation de nos choix et implémentations

Analyse et implémentation



Phase 1 : Analyse

- Compréhension du sujet
- Recherche de librairies adéquates pour le projet
- Analyse de ces librairies



Phase 2 : It's time to code

- Implémentation des différents protocoles trouvés
- Implémentation du système de prédiction et d'entraînement

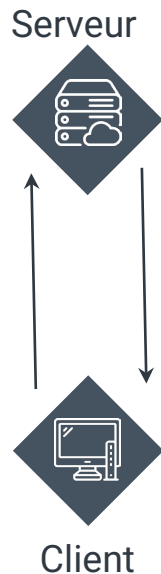


Fonctionnement



Etablissement de la connection TLS

- Négociation de la version TLS et des algorithmes de chiffrement
- Authentification du serveur



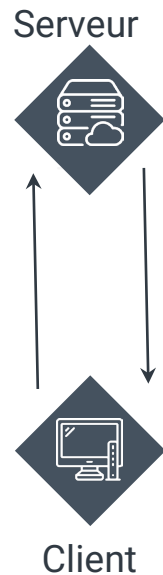
- Échange de clés entre le serveur.
- Dérivation des clés symétriques

Fonctionnement



Transfert de fichiers (1)

- Calcul de hash SHA-256 pour chaque « chunk »
- Envoi du hash



- Vérification côté serveur
 - Renvoi du résultat

Fonctionnement



Transfert de fichiers (2)

- Calcul de hash HMAC côté client du fichier entier
- Envoi du hash

Serveur



Client

- Vérification côté serveur
 - Renvoi du résultat



03

Difficultés rencontrées

Retour d'expériences sur notre projet



Difficultés rencontrées



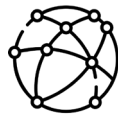
Rust

Apprentissage
express du langage



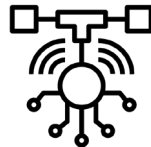
SMC

Compréhension fine
du sujet



Réseau

Gestion efficace du
réseau



Protocole

Implémentation des
différents protocoles



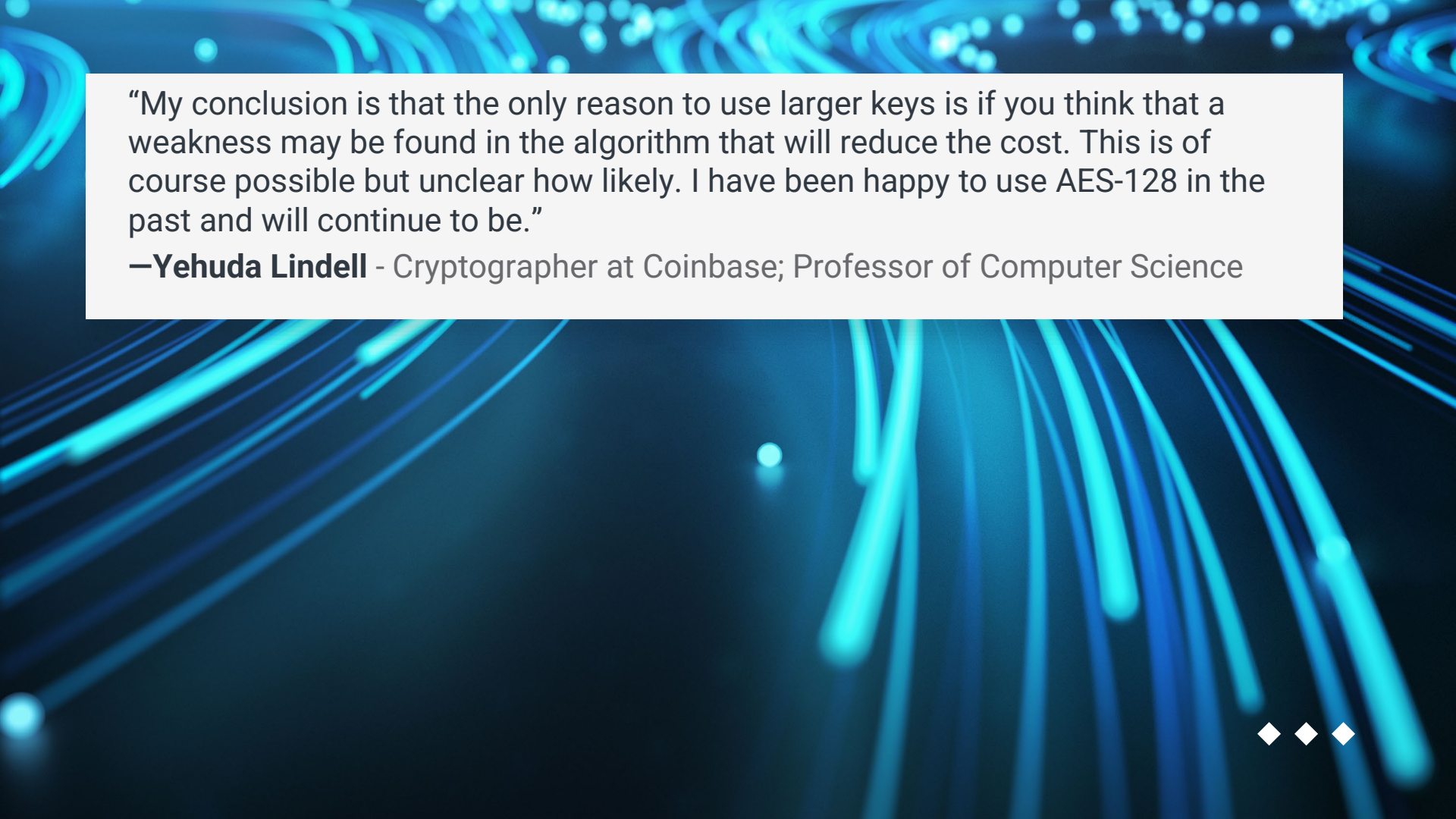
04

Conclusion

Vue d'ensemble du projet

Récapitulatif

- Utilisation du langage **Rust** et de la bibliothèque **Tonic** pour la communication **gRPC**
- Communication **TLS** sécurisée entre les parties
- Génération de certificats avec **OpenSSL** et utilisation de clés **RSA** de **2048 bits**
- Vérification d'intégrité grâce à **SHA-256** et protocole **HMAC**
- Recommandations futures :
 - ▶ Explorer d'autres modèles et algorithmes d'apprentissage automatique
 - ▶ Étudier d'autres bibliothèques et protocoles SMC
 - ▶ Étendre le projet pour un environnement totalement malveillant



“My conclusion is that the only reason to use larger keys is if you think that a weakness may be found in the algorithm that will reduce the cost. This is of course possible but unclear how likely. I have been happy to use AES-128 in the past and will continue to be.”

—**Yehuda Lindell** - Cryptographer at Coinbase; Professor of Computer Science



MERCI !

Des questions ?

Théo Le Bever

Marius André

Maxence Crouzy

Romain Nakusi

