# Échange de clé authentifié par mot de passe post-quantique

Guillaume Chirache Christopher Calvet Thomas Sauvage Emre Ucar
Timothée Fisher

Proposition détaillée de Projet Scientifique Collectif
Octobre 2023

## Table des matières

# 1 Contexte et objectif du projet

La cryptographie, c'est-à-dire le fait de chiffrer des données et communications afin de les rendre illisibles pour les tiers non autorisés, est une composante essentielle de la sécurité informatique moderne. Elle est aujourd'hui utilisée par pratiquement tous les sites web, serveurs de messagerie et applications de communication pour se prémunir des interceptions réseau, et les appareils (ordinateurs, téléphones) vendus sont quasiment tous chiffrés par défaut.

Le cas classique est celui de la **cryptographie symétrique**, qui regroupe les protocoles où le chiffrement et le déchiffrement se font par la même clé secrète. L'algorithme le plus communément utilisé est AES, publié en 2000, et il est en pratique utilisé pour quasiment toutes les communications aujourd'hui.

Son pré-requis est néanmoins que les deux personnes qui communiquent (appelons-les Alice et Bob comme on le fait classiquement) doivent partager une clé secrète commune. Il faut donc également définir un protocole d'**échange de clé**. En pratique, l'idée est en général d'utiliser du **chiffrement asymétrique**, où la clé de chiffrement (clé publique) est différente dans la clé de déchiffrement (clé secrète ou clé privée). Si Alice veut communiquer avec Bob, elle génère une clé symétrique, la chiffre avec la clé publique de Bob avant de la lui transmettre. Bob n'a plus qu'à déchiffrer la clé symétrique avec sa clé privée, et ils peuvent l'utiliser pour s'échanger des messages.

Ce schéma requiert néanmoins qu'Alice soit sûre que la clé publique dont elle dispose soit celle de Bob, sous peine que quelqu'un se fasse passer pour lui. Cela peut être garantir par une infrastructure à clés publiques (PKI) : c'est-à-dire une autorité de confiance (en pratique une chaîne d'autorité de confiance) qui fournit les clés publiques à utiliser. Les sites web utilisent des certificats SSL fournis par des autorités de certification garantissant que le site web visité correspond bien au nom de domaine affiché.

Néanmoins, la gestion d'une PKI est relativement complexe, car elle requiert des capacités de calcul et de communication (accès à Internet) qui ne sont pas réunies dans tous les cas de figure. On peut citer par exemple les objets connectés, les cartes à puce ou les passeports biométriques. Dans ces cas, il est nécessaire de trouver un autre moyen de s'assurer de l'identité de l'autre partie. C'est le rôle d'un **échange de clé authentifié par mot de passe (PAKE)** : les deux parties disposent d'un mot de passe commun, et l'échange de clé se fait en utilisant ce mot de passe. PAKE est utilisé pour la protection des données sensibles des puces des passeports de tous les pays européens et de la nouvelle carte d'identité française. [1][2]

Pour rendre les PAKE utilisables en pratique, on doit s'autoriser des mots de passe de faible entropie [1] (i.e. des mots de passe faibles), ce qui implique que le nombre de tentatives que puisse faire un attaquant soit restreint. Les attaques *online* (en présence du composant à protéger) peuvent être facilement limitées, et les attaques *offline* (fondée sur la cryptanalyse de communications interceptées) doivent être rendues impossibles par les propriétés du PAKE.

Tous ces mécanismes de cryptographie sont robustes et, malgré des évolutions dues principalement à la puissance de calcul croissante des ordinateurs, sont utilisés depuis une vingtaine d'années. Leur robustesse s'appuie sur des problématiques mathématiques NP-complets (non résolubles en temps polynomial [2]), comme le calcul du logarithme discret ou la factorisation d'entiers en facteurs premiers.

Cependant, l'émergence possible d'ordinateurs quantiques avec un grand nombre de qubits efficaces est susceptible de changer la donne, en permettant l'exécution d'algorithmes non utilisables sur un ordinateur classique. Si AES n'est pas menacé (il suffit de doubler la taille des clés pour résister à l'algorithme de Grover), les algorithmes de chiffrement asymétrique (et donc de PAKE) sont quant à eux cassés par les algorithmes comme celui de Shor. Il est donc nécessaire de trouver des alternatives post-quantiques à ces algorithmes, c'est-à-dire des algorithmes de chiffrement exécutés sur un ordinateur classique, mais résistant aussi bien aux attaques par des ordinateurs classiques que quantiques. [3][4][5]

L'objet de ce PSC est d'implémenter un PAKE post-quantique, et plus précisément le protocole CAKE, proposé en 2023 par une équipe comprenant Mme Mélissa Rossi, chercheuse à l'ANSSI. Ce

1. Si un mot de passe est généré uniformément parmi $N$ possibilités, il a une entropie de $\log_2(N)$. Par exemple, un PIN à 6 chiffres a une entropie d'environ 20 bits, ce qui est très faible.
2. Sous réserve que $\mathrm{P} \subsetneq \mathrm{NP}$.

travail de recherche contient une preuve de validité, mais il s'agit d'un protocole très amont, et de nombreuses questions pratiques d'implémentation restent à résoudre. [6]

# 2 État de l'art

Le concept de PAKE existe depuis les années 1990, et un certain nombre de protocoles ont été proposés, avec des degrés de preuve différents. Les plus utilisés dans l'industrie sont EKE et OEKE. Ils sont basés sur le problème du logarithme discret, et sont donc cassés par les ordinateurs quantiques. [2][7]

L'objet de l'article de Mme Rossi est de montrer que, sous réserve qu'ils vérifient un certain nombre de bonnes propriétés (comme l'indistinguabilité des clés publiques générées), il est envisageable de transformer un KEM en PAKE. De plus, cet article montre que **CRYSTALS-Kyber** vérifie ces propriétées. [6]

L'idée de PAKE ainsi construit (appelé CAKE) est de qu'Alice génère une paire de clés $(pk, sk)$, puis envoie la clé publique chiffrée à Bob à l'aide d'un identifiant de session et du mot de passe. Bob la déchiffre, puis utilise `Encaps` à partir de $pk$ pour obtenir une clé symétrique $K$ et son chiffré $c$. Il envoie à Alice $c$, toujours chiffré avec identifiant de session et mot de passe, ce qu'Alice peut déchiffrer. Elle peut ainsi utiliser $sk$ pour obtenir $K$, et la clé finale symétrique $SK$ est dérivée des clés désormais communes (en particulier $K$).

L'équipe de Melissa Rossi a prouvé la sécurité de ce modèle dans le cadre théorique de l'*Universal Composability* (UC), reconnu comme l'un des plus solides. Cependant, ce travail reste très amont, et de nombreuses questions pratiques doivent être résolues. En particulier, le *cipher* utilisé pour les chiffrements avec le mot de passe doit être un **ideal cipher**, c'est-à-dire un algorithme qui se comporte comme une permutation aléatoire, ce qui n'est pas évident pour de telles données. Il reste donc à résoudre ces problèmes techniques, puis à proposer une implémentation solide dans un langage de programmation comme le C.

# 3 Objectifs intermédiaires et échéancier

Afin de mener à bien ce projet, nous avons défini plusieurs objectifs intermédiaires. En jalonnant le projet de cette manière, nous espérons pouvoir avancer efficacement, guider correctement notre travail et garder une vision claire de l'avancement du projet.

Dans un premier temps, après avoir pris connaissance du protocole CAKE et de ses spécifications, nous allons implémenter un prototype en Python. Ce prototype nous permettra de nous familiariser avec le protocole et de nous assurer que nous avons bien compris les spécificités du problème de l'implémentation. Nous pourrons l'utiliser pour identifier les points d'ombre et donc en partie les questions à poser à Mme Rossi.

Après avoir implémenté ce prototype sommaire, nous souhaitons comprendre et résoudre les problèmes identifiés lors de la phase de familiarisation. Nous devrons arrêter notre étude sur des choix d'implémentation précis qui permettront de passer d'une idée théorique à une implémentation exploitable par des utilisateurs industriels. En particulier, nous devrons choisir comment retranscrire les concepts cryptographiques en code, comme par exemple celui d'*oracle* ou d'*ideal cypher*.

Une fois ces choix effectués, nous pourrons implémenter le protocole CAKE dans son intégralité dans un langage de programmation de plus bas niveau, comme le C. Nous pourrons alors effectuer des tests de performance et de sécurité sur notre implémentation. Nous devrons également veiller à correctement documenter notre travail afin de permettre à d'autres personnes de l'utiliser et de l'exploiter dans des applications concrètes.

Enfin, nous souhaitons nous-même utiliser notre implémentation dans une application concrète. Nous voulons pour cela construire un démonstrateur comme un lecteur de passeport biométrique ou une carte à puce. Nous pourrons ainsi tester notre implémentation dans un contexte réel et nous assurer de sa pertinence.

# 4 Méthodologie, organisation du travail et répartition des tâches

Étant un groupe de cinq, nous avons la possibilité de nous répartir le travail, mais la difficulté et que le sujet est très technique et que les problèmes sont souvent interconnectés. Nous devons donc communiquer régulièrement afin de tous rester à jour.

Nous ferons (et avons déjà commencé à faire) des réunions bimensuelles avec notre tutrice, Mme Rossi. Celles-ci permettront de faire le point sur notre avancement, de discuter des difficultés rencontrées, et de pouvoir être orientés dans la bonne direction.

Nous travaillons les autres mercredis ensemble, de manière à partager une vision commune du cadre mathématiques et de bénéficier des expériences de chacun (nous venons de filières différentes), et de manière isolée le reste du temps. Nous avons mis en place un espace Google Drive, une bibliographie Zotero et un dépôt Git pour faciliter le partage de nos notes et réflexions.

L'essentiel du travail en ce début de projet consiste, selon les directions fixées par notre tutrice, à résoudre les problèmes techniques d'implémentation. En pratique, cela nécessite de nous former aux notions employées dans la cryptographie, à faire de la recherche documentaire pour trouver des résultats analogues à ceux que nous voulons (ce qui est facilité par le fait que l'*open access* soit la norme dans la recherche en cryptographie), et à rédiger formellement les spécifications, de manière à pouvoir les présenter, en garder une trace claire, et faciliter leur programmation une fois cette étape venue.

La deuxième phase consistera à programmer le protocole, en commençant par une version Python de démonstration (déjà ébauchée), puis une version C, nécessaire pour le exigences de performance des usages industriels. Nous ferons ce travail de groupe comme tout projet de développement, c'est-à-dire en utilisant un dépôt GitLab commun, et en veillant à documenter et rendre suffisamment modulaire le code pour que la reprise de notre travail par les autres membres du projet ne pose pas de difficulté. Cette démarche est facilitée par le fait que plusieurs d'entre nous ont été entraînés à cette démarche dans le cadre de développement pour le Binet Réseau.

Enfin, la réalisation d'un démonstrateur physique se fera en commun, les difficultés principales étant le choix du matériel, sa configuration, et le code d'un programme cumulant communication NFC et notre CAKE en C.

# 5 Contribution de l'ANSSI

Nous avons initialement formé notre groupe de PSC autour de l'envie partagée de travailler sur un sujet de cybersécurité qui puisse avoir des applications concrètes au delà du cadre purement scolaire. Nous nous sommes tournés vers l'ANSSI – l'Agence nationale de la sécurité des systèmes d'information – à l'aide de relations personnelles.

Nous avons ainsi été mis en contact avec Mme Mélissa Rossi, chercheuse à l'ANSSI, qui nous a proposé de travailler sur le protocole CAKE dont elle est co-autrice. Cela nous offre l'occasion de travailler sur un protocole à un stade encore très amont, et donc de participer activement au développement de la cryptographie post-quantique.

L'encadrement par une chercheuse experte du domaine nous permet de bénéficier de ses conseils afin de mieux le comprendre, de même que les difficultés d'implémentation que nous pouvons rencontrer. En qualité de tutrice, elle nous aiguille dans notre travail et nous aide à identifier les objectifs importants afin de mener à bien le PSC.

Il est également prévu que nous rencontrions d'autres chercheurs de l'ANSSI spécialistes des réseaux de Feistel. Ces constructions sont potentiellement de bons candidats pour le chiffrement symétrique de notre implémentation finale. L'idée serait de bénéficier de leurs connaissance des propriétés de ces réseaux pour aboutir à une construction sécurisée tout en évitant l'écueil d'une erreur d'implémentation. Un rendez-vous à déjà été fixé en janvier pour un premier entretien.

## 6 Résultats préliminaires

Tout d'abord, nous avons étudié en détail l'article de Mme Rossi, pour comprendre le fonctionnement du protocole CAKE proposé par les chercheurs.

Nous avons développé une première version du protocole en Python (en nous appuyant sur une implémentation pré-existante de Kyber). Nous nous sommes toutefois heurté à un problème, qui constituera l'un des problèmes centraux du PSC : quel algorithme utiliser pour chiffrer la clé publique qu'Alice envoie à Bob ?

Cet algorithme doit en effet se rapprocher le plus possible d'un **ideal cipher**. Cela signifie que la répartition des textes chiffrés produits par l'algorithme doit se rapprocher le plus possible d'une permutation aléatoire. De plus, il doit vérifier certaines propriétés pour éviter les attaques *offline* : si les distributions des textes chiffrés avaient des formes différentes selon la clé, un attaquant potentiel pourrait en déduire des informations sur la clé utilisée (i.e. éliminer des mots de passe). Concrètement, cela signifie qu'une simple utilisation d'AES ne peut convenir. Un des objectifs de notre projet, auquel nous travaillons actuellement, est de dégager les conséquence de cette assertion pour pouvoir trouver un protocole de chiffrement qui peut convenir.

Notre tutrice nous a en outre suggéré d'explorer la piste des réseaux de Feistel généralisés. Nous avons donc commencé à nous familiariser avec ce type de chiffrement. Elle compte également nous faire rencontre des chercheurs de l'ANSSI, experts en la matière, pour nous aider à résoudre ce problème. [8]

Enfin, en nous appuyant sur un énoncé de travaux pratiques que nous a envoyé Mme Rossi, nous avons implémenté une version simplifiée du protocole Kyber dans le langage Sage (une version enrichie de Python). Nous souhaitions en effet avoir quelques éléments de compréhension du fonctionnement de Kyber, même si, grâce à l'implémentation pré-existante du protocole en Python que nous utilisons, il n'est pas dérangeant de le considérer comme une « boite noire ».

## Références


[1] Ministère de l'Intérieur et des Outre-mer : La puce de la nouvelle carte nationale d'identité, 2021.

[2] Victor Boyko, Philip MacKenzie et Sarvar Patel : Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. *In* Bart Preneel, éditeur : *Advances in Cryptology — EUROCRYPT 2000*, Lecture Notes in Computer Science, pages 156–171, Berlin, Heidelberg, 2000. Springer.

[3] Peter W. Shor : Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, octobre 1997. arXiv :quant-ph/9508027.

[4] ANSSI : ANSSI views on the Post-Quantum Cryptography transition, 2022.

[5] Samuel Jaques, Michael Naehrig, Martin Roetteler et Fernando Virdia : Implementing Grover Oracles for Quantum Key Search on AES and LowMC. *In* Anne Canteaut et Yuval Ishai, éditeurs : *Advances in Cryptology – EUROCRYPT 2020*, volume 12106, pages 280–310. Springer International Publishing, Cham, 2020.

[6] Hugo Beguinet, Céline Chevalier, David Pointcheval, Thomas Ricosset et Mélissa Rossi : GeT a CAKE : Generic Transformations from Key Encaspulation Mechanisms to Password Authenticated Key Exchanges, 2023. Publication info : Published elsewhere. 21st International Conference on Applied Cryptography and Network Security (2023).

[7] S.M. Bellovin et M. Merritt : Encrypted key exchange : password-based protocols secure against dictionary attacks. *In Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 72–84, Oakland, CA, USA, 1992. IEEE Comput. Soc. Press.

[8] Viet Tung Hoang et Phillip Rogaway : On Generalized Feistel Networks. *In* David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu

Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum et Tal Rabin, éditeurs : *Advances in Cryptology – CRYPTO 2010*, volume 6223, pages 613–630. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.