h e p i a
Haute école du paysage, d'ingénierie
et d'architecture de Genève

Hes·SO GENÈVE
Haute Ecole Spécialisée
de Suisse occidentale

Bruteforce password attack on FPGAs

< Insérez ici votre illustration >

(non obligatoire)

Projet de semestre présenté par

Abivarman KANDIAH

Informatique et systèmes de communication avec orientation
Systèmes informatiques embarqués

mars, 2024

Professeur-e HES responsable

Andres UPEGUI POSADA, Stéphane
Gaétan KÜNG

Mandant

ELCA Security

Légende et source de l’illustration de couverture :

TABLE DES MATIÈRES

La table des matières doit reprendre tous les niveaux de titre et sous-titre du mémoire, y compris les pages initiales (page des remerciements, énoncé du sujet, résumé, table des annexes et autres tables), ainsi que les références documentaires, etc.

h e p i a
Haute école du paysage, d'ingénierie
et d'architecture de Genève

Hes·SO GENÈVE
Haute Ecole Spécialisée
de Suisse occidentale

RÉSUMÉ

Lorsqu'un utilisateur doit s'authentifier auprès d'un service, il doit fournir un mot de passe préalablement défini. Pour des raisons de sécurité, le système stocke ce mot de passe en le passant par une fonction de hachage. Ainsi, lors de l'authentification, le système compare le hash du mot de passe entré par l'utilisateur avec le hash stocké pour vérifier son identité. Une fonction de hachage qui est souvent utilisé pour le stockage de mots de passe est le Bcrypt, qui a comme particularité d'être assez lente, rendant les mots de passe assez résistante aux attaques par bruteforce. Ce rapport a pour but de détailler la mise en oeuvre d'un système visant à attaquer les mots de passe protégés par l'algorithme bcrypt en utilisant un Field-Programmable Gate Array (FPGA). L'objectif principal est de créer une solution extensible et plus performante que les approches traditionnelles basées sur Graphics Processing Unit (GPU). Le système repose sur plusieurs cœurs de calcul parallèles sur le FPGA pour générer les hashs bcrypt. La génération des mots de passe pour le bruteforce est directement réalisée sur le FPGA à l'aide d'un système de compteur pour distribuer les tâches aux cœurs de calcul. Après validation du fonctionnement, des mesures ont été effectuées pour évaluer les performances et les ressources utilisées sur le FPGA. Pour explorer la possibilité de déléguer la génération de mots de passe à un ordinateur, une interface Peripheral Component Interconnect Express (PCIe) a été établie entre la carte FPGA et un ordinateur. Bien que le projet soit fonctionnel, il reste à mettre en place une interface complète entre l'ordinateur et la carte FPGA pour qu'elle soit utilisable lors d'une véritable attaque.

< Insérez ici votre illustration >
(obligatoire)

Candidat-e :

ABIVARMAN KANDIAH

Filière d'études : ISC

Professeur-e(s) responsable(s) :

ANDRES UPEGUI POSADA, STÉPHANE GAÉTAN KÜNG

En collaboration avec : ELCA Security

Travail de bachelor soumis à une convention de stage en entreprise : non

Travail soumis à un contrat de confidentialité : non

LISTE DE ACRONYMES

FPGA Field-Programmable Gate Array. v, 1, 3, 6, 7

GPU Graphics Processing Unit. v

IC Integrated Circuit. 3

PC Personal Computer. 6, 7

PCIe Peripheral Component Interconnect Express. v, 1

VHDL Very High Speed Integrated Circuit Hardware Description Language. 3

LISTE DES ILLUSTRATIONS

Références des URL

— URL01 ce-site.ch/bla/bli/blo/blou.html

— URL03 ce-site.ch/blou/bli/bla.html

— URL04 https://commons.wikimedia.org/w/index.php?curid=906980

— URL06 ce-site.ch/monrapportdestage.pdf

LISTE DES TABLEAUX

N.B. Si vous avez peu de tableaux, vous pouvez les intégrer à la table des illustrations.

Références des URL

— URL02 ce-site.ch/bli/bla/blo/blou

— URL05 ce-site.ch/publications/documents/rapports/rapportsdestage/monrapportdestage.pdf

LISTE DES ANNEXES

INTRODUCTION

Ce travail s'inscrit dans le cadre de mon travail de semestre, réalisé en réponse à une demande d'ELCA Security, une entreprise spécialisée dans la cyber-sécurité. Le projet présenté ici vise à explorer une approche peu commune pour attaquer des mots de passe protégés par l'algorithme bcrypt en utilisant un FPGA. Ce projet a pour objectif final de leur fournir une solution extensible et performante qui puisse être utilisable lors de leurs attaques. L'intérêt technique et scientifique de ce projet réside dans la recherche de solutions plus performantes pour l'attaque de mots de passe, en tirant parti des capacités de traitement parallèle offertes par les FPGA.

Après une recherche approfondie des implémentations existantes du bcrypt sur FPGA, le projet a débuté par une analyse du papier [1] concernant une implémentation déjà existante. En parallèle, une page Wikipédia détaillant le fonctionnement de bcrypt [2] a été utilisée comme ressource principale pour comprendre les spécificités de cet algorithme. De plus, un papier sur une attaque MD5 [3] sur FPGA a été consulté pour enrichir la compréhension des techniques d'attaque sur des dispositifs matériels. Ces ressources documentaires ont été cruciales pour comprendre les différents concepts, orienter les choix de conception et résoudre les problèmes techniques rencontrés.

Dans le cadre de ce projet, j'ai entrepris plusieurs actions significatives. Tout d'abord, j'ai repris le code de l'implémentation existante en VHDL d'un programme d'attaque par bruteforce de mot de passe bcrypt. Après avoir étudié le papier associé et constaté des incohérences dans les testbenches, j'ai refait ces derniers pour valider le programme. Par la suite, j'ai identifié et corrigé des erreurs dans le code afin d'obtenir un programme fonctionnel.

En parallèle, j'ai pu brièvement étudier le fonctionnement du PCIe afin d'y mettre en place une simple interface entre une carte FPGA et un ordinateur.

1. Friedrich WIEMER et Ralf ZIMMERMANN. "High-speed implementation of bcrypt password search using special-purpose hardware". In : *2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14)*. ISSN: 2325-6532. Déc. 2014, p. 1-6. DOI : 10.1109/ReConFig.2014.7032529. URL : https://ieeexplore.ieee.org/document/7032529.

2. *bcrypt*. en. Page Version ID: 1210874707. Fév. 2024. URL : https://en.wikipedia.org/w/index.php?title=Bcrypt&oldid=1210874707.

3. Maruthi GILLELA, Vaclav PRENOSIL et Venkat Reddy GINJALA. "Parallelization of Brute-Force Attack on MD5 Hash Algorithm on FPGA". In : *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*. ISSN: 2380-6923. Jan. 2019, p. 88-93. DOI : 10.1109/VLSID.2019.00034. URL : https://ieeexplore.ieee.org/document/8710753.

Dans ce rapport, je vais commencer par brièvement expliquer les différents notions clés de ce projet, afin de poser une base technique. Je vais par la suite vous présenter une analyse du projet, afin d'y décrire le projet de manière détaillé, expliquer le fonctionnement du Bcrypt et les recherches qui ont été faites afin de trouver une implémentation existante. Puis, je vais détailler la méthodologie de travail, en exposant les différentes étapes de mise en œuvre du projet, les simulations qui ont été faites et les différentes difficultés rencontrées. Enfin, je vais décrire les résultats obtenus lors des différents tests et mesures qui ont été faits.

CHAPITRE 0 : BASE TECHNIQUE

Ce chapitre a pour but d'introduire et expliquer les différents aspects techniques clés de ce projet de semestre. Je vais notamment expliquer brièvement ce qu'est un FPGA et le principe d'une fonction de hachage.

1.1. FPGA

Un FPGA est un Integrated Circuit (IC) dans laquelle on peut programmer et interconnecter des circuits logiques. Contrairement à un processeur, qui est limité par un certain nombre d'instructions et exécute les instructions de manière séquentielle, un FPGA permet d'exécuter de nombreux circuits logiques en parallèle.

Pour programmer un FPGA, on utilise généralement des langages de description matériel tels que le Very High Speed Integrated Circuit Hardware Description Language (VHDL) et le Verilog. Dans ce projet, j'ai personnellement travaillé avec le VHDL.

1.2. FONCTION DE HACHAGE

Une fonction de hachage est une fonction qui va prendre en entrée une donnée a taille variable et va ressortir une donnée de taille fixe.

Une des propriétés fondamentales d'une fonction de hachage est qu'il n'existe pas de fonction mathématique permettant de retrouver la donnée originale à partir d'un hash généré. Même une petite modification apportée à la donnée en entrée conduira à un hash totalement différent en sortie. Cette particularité est essentielle pour sécuriser le stockage des mots de passe, car même si des hash venait a être compromis, il est extrêmement difficile de retrouver les mots de passe originaux à partir de leurs hachages.

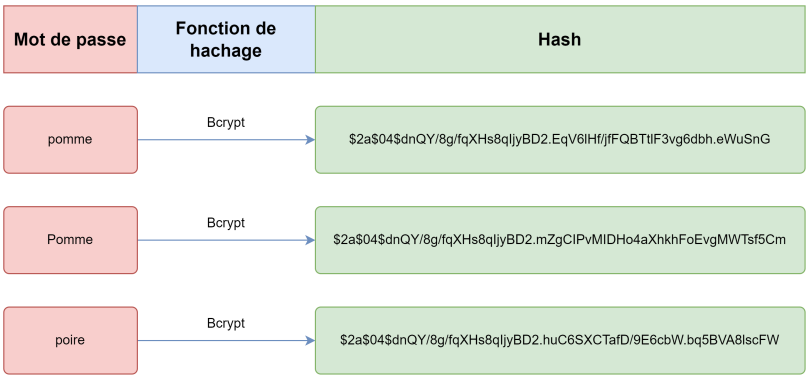

ILLUSTRATION 1.1 – Fonction de hachage. Source : réalisé par Kandiah Abivarman

a. Salt

Certaines fonctions de hachage tel que le bcrypt utilisent ce qu'on appelle un salt, qui est une valeur générée aléatoirement qu'on va donner avec notre mot de passe. Le salt va permettre d'avoir un hash différent, même si deux personnes utilisent le même mot de passe, ajoutant ainsi une couche supplémentaire de sécurité.

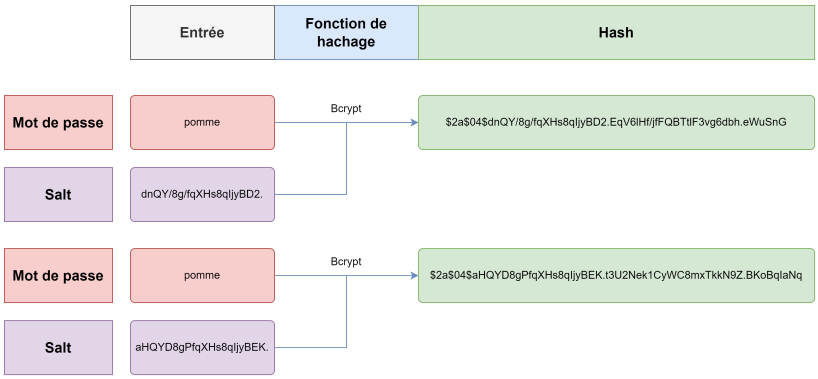

ILLUSTRATION 1.2 – Salt. Source : réalisé par Kandiah Abivarman

b. Attaque par bruteforce

L'attaque par bruteforce consiste à essayer toutes les combinaisons possibles de mots de passe afin de retrouver celui qui correspond au hash compromis. Cette méthode repose sur le fait qu'il est impossible de retrouver directement le mot de passe à partir du hash, obligeant ainsi l'attaquant à tester différentes entrées jusqu'à ce qu'il trouve celle qui génère le hash recherché.

Toutefois, cette méthode peut prendre beaucoup de temps, notamment lorsque les fonctions de hachage utilisées sont conçues pour être lentes à calculer.

CHAPITRE 1 : ANALYSE

Ce chapitre a pour but d'expliquer de manière détaillée l'objectif de ce projet de semestre. Je vais aussi faire part des différentes idées qui sont ressorties lors de nos discussions avec mon professeur. Par la suite, je vais expliquer en quoi consiste la fonction de hachage Bcrypt, son fonctionnement et ses spécificités. Pour finir, je vais rapporter les différentes implémentations sur FPGA que j'ai pu retrouver et celui que j'ai fini par reprendre durant le projet de semestre.

2.1. DESCRIPTION DU PROJET

L'objectif principal de ce projet est d'exploiter le parallélisme offert par les FPGA, afin de calculer les fonctions de hachage nécessitant beaucoup de temps de calculs. Le but étant d'avoir au final un système plus efficient que les solutions actuelles lors d'une attaque par bruteforce. Il est aussi nécessaire d'avoir une certaine communication entre le Personal Computer (PC) de l'attaquant et le FPGA, afin que l'attaquant puisse fournir le hash qu'il souhaite casser.

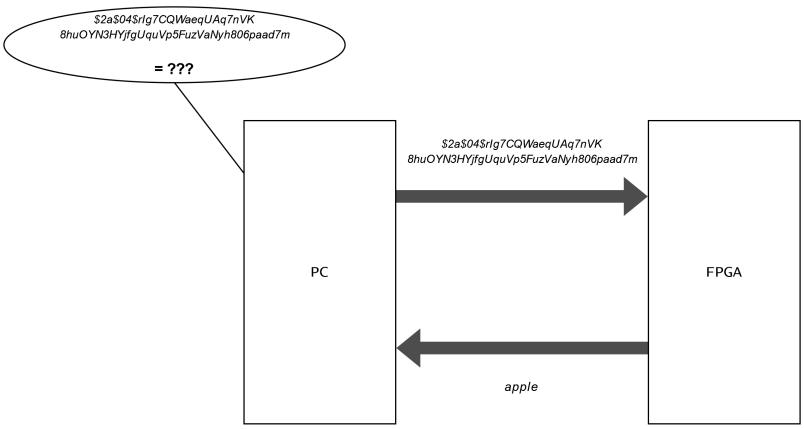

ILLUSTRATION 2.1 – Diagramme général. Source : réalisé par Kandiah Abivarman

Une des première question qu'on s'était posée avec les professeurs était la manière pour générer les mots de passe lors d'une attaque. Dans notre cas, nous avons deux possibilités, soit nous générons les mots de passe directement depuis le PC et devons les transmettre à la carte FPGA afin de procéder au hachage, sinon la génération se fait directement dans le FPGA. La différence est que dans le premier cas, on peut être possiblement limité au niveau logiciel par le PC ou au niveau de la communication avec le FPGA. Toutefois, au niveau de la génération des mots de passe on aura plus de flexibilité permettant d'autres types d'attaques comme par

exemple une attaque par dictionnaire [4].

Pour le projet de semestre, j'ai décidé de partir plutôt vers la génération sur FPGA, afin d'avoir une première solution entièrement fonctionnelle sur FPGA sans dépendance avec un PC. La génération sur PC est toutefois envisageable par la suite pour le projet de bachelor.

2.2. BCRYPT

a. Algorithme

b. Format du Hash

2.3. IMPLÉMENTATIONS EXISTANTES

4. TO DO

CHAPITRE 2 : CONCEPTION

3.1. BCRYPT SUR FPGA

a. Bcrypt Core

b. Password Generator

c. Bcrypt Quadcore

d. Bcrypt Cracker

3.2. INTERFACE PCIE SUR FPGA

CHAPITRE 3 : RÉSULTATS

4.1. BCRYPT CRACKER

a. Validation

b. Mesures

4.2. INTERFACE PCIE

a. Validation

CONCLUSION

Votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte, votre texte.

ANNEXES

Imprimer idéalement cette page sur une page de couleur. Chaque annexe doit commencer sur une nouvelle page et doit être numérotée : Annexe 1 puis Annexe 2, etc.

ANNEXE 1

ANNEXE 2

ANNEXE 3

RÉFÉRENCES DOCUMENTAIRES

bcrypt. en. Page Version ID: 1210874707. Fév. 2024. URL : https://en.wikipedia.org/w/index.php?title=Bcrypt&oldid=1210874707.

GILLELA, Maruthi, Vaclav PRENOSIL et Venkat Reddy GINJALA. "Parallelization of Brute-Force Attack on MD5 Hash Algorithm on FPGA". In : *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*. ISSN: 2380-6923. Jan. 2019, p. 88-93. DOI : 10.1109/VLSID.2019.00034. URL : https://ieeexplore.ieee.org/document/8710753.

WIEMER, Friedrich et Ralf ZIMMERMANN. "High-speed implementation of bcrypt password search using special-purpose hardware". In : *2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14)*. ISSN: 2325-6532. Déc. 2014, p. 1-6. DOI : 10.1109/ReConFig.2014.7032529. URL : https://ieeexplore.ieee.org/document/7032529.

Sites Web consultés – Code repris d'ailleurs – Notices techniques – Articles de presse – Ouvrage imprimés – Ouvrages électroniques – Chapitre dans un ouvrage imprimé – Rapports imprimés – Travaux universitaires – Articles de revues imprimés – Articles de périodiques électroniques – Communication dans un congrès. Pour chacun de ces types de document, les mise en forme sont dans le document « Méthode de citation et de rédaction d'une bibliographie ».

Afin de gagner du temps, pensez à utiliser le logiciel de gestion bibliographique Zotero (et/ou BibTeX si vous utilisez LaTeX) pour la mise en forme et l'édition automatique de vos références à la norme ISO690.