

*Eude et comparaison des principaux systèmes crypto  
fournis par le package de Bouncy Castle  
sous la plateforme Java SDK*

Faculté des Nouvelles Technologies de l'Information et de la Communication  
Département d'Informatique et des Technologies de l'Information  
2ème année Master Informatique Fondamentale  
Bouneghab Noura & Debagh Basma  
Encadreur : Mr. Salah Euschi

**Résumé :**

- Ce travail consiste à comparer les principaux algorithmes crypto fournis par le package crypto open source de Bouncy Castle pour permettre aux développeurs d'applications d'utiliser les algorithmes crypto les plus efficaces pour la prise en charge de la sécurité informatique dans leurs applications à réaliser.
- **Bouncy Castle** est une collection d'API utilisées en cryptographie. Elle comprend les API tant pour le Java JCA/JCE, C#, et une API légère adaptée aux environnements mobiles et les applets Java. Nous avons choisi ce package parce qu'il est open source, riche en algorithmes crypto et connaît un développement continu depuis son apparition sur Internet. Ce package ressemble beaucoup à la librairie C [openssl](#) qui est conforme aux différents standards en vigueur.
- Nous utilisons des Benchmarks crypto pour mesurer ces algorithmes. Les principaux benchmarks sont la sécurité et la rapidité de calcul. La sécurité se mesure en nombre de bits formant la clé, la rapidité de calcul se mesure en temps d'exécution en millisecondes ou en nombre d'octets par seconde.
- **Les mots clé :** Cryptographie, package crypto, Bouncy Castle, Benchmark Crypto, sécurité, Java JCA/JCE, chiffrement, signature numérique.

**INTRODUCTION:**

- Notre projet rentre dans le cadre de la sécurité informatique. La cryptographie n'est pas toute la sécurité mais nous ne pouvons pas réaliser la sécurité informatique sans cryptographie.
- La crypto permet de réaliser les objectifs de sécurité de confidentialité, intégrité, authentification, et de disponibilité via ses outils de chiffrement, signature, fonctions de hachage et Mac (Code d'authentification de message).
- La cryptographie permet de sécuriser les communications et fournir l'authentification, mais l'application de la cryptographie au niveau logiciel dans les applications web peut ralentir le temps d'exécution. Si la cryptographie n'est pas bien implémentée, on pourra avoir un problème de performance. Un mauvais choix des algorithmes crypto et des tailles de clés peuvent affaiblir la sécurité du système. Pour assurer la non-répudiation, une infrastructure de certificats à clé publique tierce partie doit être utilisée. La PKI (Public Key Infrastructure) est le système leader de la sécurité des réseaux et des transactions e-commerce. Elle permet à des utilisateurs d'échanger des données et de l'argent d'une manière sécurisée et confidentielle.
- Les développeurs d'applications web ont besoin d'algorithmes de cryptographie. On trouve ces algorithmes enveloppés dans un ensemble de packages qui offrent des services cryptographiques aux développeurs. Le package est une API qui permet au développeur de sécuriser le contenu de données au lieu de sécuriser uniquement la connexion avec le serveur. Pour sécuriser le contenu des données.

**CONCLUSION.**

- Dans ce travail, nous allons apprendre les concepts de base de la cryptographie moderne ainsi que les algorithmes crypto recommandés par les standards de sécurité afin de rendre nos applications informatiques particulièrement celles qui échangent des données sensibles plus sécurisées.
- Les benchmarks à réaliser nous permettent de choisir l'algorithme approprié à toute opération cryptographique afin de garantir un accès sécurisé aux messages et données échangées entre les utilisateurs

**REFERENCES BIBLIOGRAPHIQUES (A revoir )**

- Bouncycastle.org, The Legion of the Bouncy Castle, SPECIFICATIONS, <http://www.bouncycastle.org/specifications.html>, September 2010.
- VAMPIRE&ECRYPT II, eBACS: ECRYPT Benchmarking of Cryptographic Systems. <http://bench.cr.yp.to/index.html>.
- VAMPIRE&ECRYPT II, eBATS: ECRYPT Benchmarking of Asymmetric Systems <http://bench.cr.yp.to/ebats.html>.
- VAMPIRE&ECRYPT II, eBASC: ECRYPT Benchmarking of Stream Ciphers <http://bench.cr.yp.to/ebasc.html>.
- VAMPIRE&ECRYPT II, eBASH: ECRYPT Benchmarking of All Submitted Hashes <http://bench.cr.yp.to/ebash.html>

**MATERIEL ET METHODES :**

- **La plate forme Java :**
- Très utilisé, plus simple que le C++ orienté objet, robuste et sûr. Dispose d'une machine virtuelle JVM indépendante des machines et leurs OS Très performant.
- Byte code interprété, multi-tâches et dynamique.
- Fournit un Framework JCA/JCE pour la prise en charge de plusieurs fournisseurs de services crypto
- **La package de Bouncy Castle :**
- A télécharger, installer et configurer sous Java JDK 1.7.
- Choix des principaux algorithmes de crypto.
- **NetBeans :** Editeur d'environnement intégré pour l'écriture le test et l'exécution du byte code Java relatifs à nos benchmarks à réaliser.

**RESULTATS :**

Nous souhaitons avoir des résultats qui confirment les recommandations et les règles fondamentales à appliquer dans le choix de ces algorithmes et qui sont fournis par les standards de sécurité en vigueur. Nos résultats seront présentés sous forme de graphe et d'histogramme pour montrer la différence entre chaque famille d'algorithmes crypto.

- **ANALYSE et DISCUSSION :** Nos résultats attendus de notre étude seront comparés aux règles et standards en vigueur et devront confirmer les tendances de la recherche dans ce domaine.