Mini-ChaCha

Ce rapport résume l'ensemble du travail réalisé pour implémenter un variant simplifié de ChaCha20 (Mini-ChaCha) et démontrer l'insécurité d'une clé de 32 bits par attaque brute force.

Logique et méthode :

Nous avons commencé par implémenter Mini-ChaCha, une version réduite de ChaCha20 avec des paramètres de sécurité volontairement faibles pour l'analyse. Ensuite nous avons développé un script d'attaque par brute force séquentiel pour démontrer concrètement que des clés de 32 bits sont vulnérables aux attaques sur du matériel standard. L'attaque a été faite par un PC et pas vraiment un laptop donc avec des performances meilleures mais rien de vraiment significatif à notre avis.

1- Implémentation Mini-ChaCha:

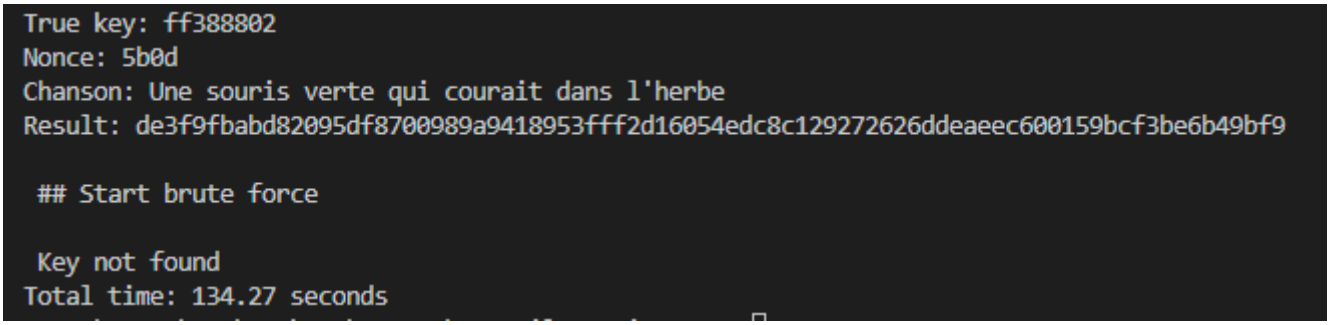
- **State** : 16 bytes (4×4 matrice de valeurs 8-bits) au lieu de 64 bytes
- **Key** : 32 bits (4 bytes, dupliquée pour former 8 bytes)
- **Nonce** : 16 bits (2 bytes)
- **Counter** : 16 bits
- **Rounds** : 5 double-rounds

2- Opérations cryptographiques:

- `quarter_round()` : opérations ARX (Addition, Rotation, XOR) sur 8 bits
- `doubleround()` : quarter-rounds sur colonnes puis diagonales
- `mini_chacha_block()` : génération de blocs de flot de bits de 16 bytes

3- Démonstration d'attaque:

- **Méthode séquentielle**: parcours de l'espace des clés de 0 à N, garantit de trouver toute clé dans la plage testée



4- Mesure de l'attaque:

- Extrapolation pour l'espace complet (2^{32} = 4,294,967,296 clés)
- Temps écoulé par million de clés testées = ~130s

On peut donc extrapoler et estimer que pour rechercher l'espace complet, l'algorithme prendrai ~155 heures (6.5 jours). En conclusion, une clé de 32bits n'est effectivement pas très sûre.

Code Source :

Le dépôt Git associé est disponible à : https://github.com/ThemlaouiHou/travail-pratique-1-

Fichiers principaux :

- `mini_chacha.py` : Implémentation complète de Mini-ChaCha
- `brute_force.py` : Script d'attaque par brute force séquentiel

LLM:

ChatGPT a été utilisé pour :

- Explications sur le fonctionnement de l'algorithme.
- Aide avec certaines synthaxes python.
- Suggestions d'implémentation et d'amélioration des méthodes encrypt() et mini_chacha-block() de MiniChacha.
- Suggestion de procédure pour l'attaque séquentielle (squelette)
- Aide au débugging