Rapport - SAE 3.04

Partie 2
Liaisons épistolaires

CHABILAN Léni
DEBRAY Antoine

2023-2024

répartition des tâches **3**

Leni Chabilan 3

Partie 2: Un peu d’aide 3

Partie 3: Analyse des messages 3

Antoine Debray 4

Partie 1: Premières tentatives 4

Partie 2: Un peu d’aide - Images 4

Partie 4 et Rapport 4

Réponses aux questions **5**

Partie 1: 5

Partie 2: 5

Partie 4: 5

Répartition des tâches

Leni Chabilan

Partie 2: Un peu d’aide

- Implémentation de l'algorithme AES en Python : Leni a pris en charge l'implémentation de l'algorithme AES avec des clés de 256 bits. Il a assuré que l'implémentation était conforme aux spécifications du projet et a effectué des tests pour garantir son bon fonctionnement.

- Estimation du temps nécessaire pour casser AES, avec prise en compte de la configuration de l'ordinateur.

- Explication succincte d'autres types d'attaques, notamment les attaques par canal auxiliaire et les attaques par fautes

Partie 3: Analyse des messages

- Mise en œuvre du programme utilisant Scapy : Leni a participé à la mise en œuvre du programme utilisant Scapy pour filtrer les messages réseau. Il a contribué à l'élaboration de mécanismes permettant de ne conserver que les messages d'Alice et Bob, ainsi qu'à la phase de déchiffrement des données transportées par ces paquets.

- Documentation associée : Leni a pris part à la rédaction de la documentation associée à la partie 3, en veillant à ce que les explications soient claires et compréhensibles

Antoine Debray

Partie 1: Premières tentatives

- Implémentation de l'algorithme SDES : Antoine a pris en charge l'implémentation de l'algorithme SDES en Python, en se concentrant sur la fonction de chiffrement et de déchiffrement. Il a assuré la cohérence avec les spécifications du projet et a contribué à la validation de l'efficacité de l'algorithme.

Partie 2: Un peu d'aide - Images

- Analyses et résolutions des énigmes des images : Antoine a été responsable de l'analyse des deux images fournies par MrRobot. Il a mis en œuvre les étapes nécessaires pour résoudre les énigmes.

Partie 4 et Rapport

- Mise en œuvre de la Partie 4 : Antoine a été chargé de la mise en œuvre de la partie 4, qui implique des questions supplémentaires pour approfondir les réflexions sur le projet. Il a également contribué à la rédaction du rapport final en rassemblant les différentes sections et en assurant la cohérence globale du document.
- Coordination du rapport final : Antoine a joué un rôle central dans la coordination des différentes parties du rapport, s'assurant que le contenu est présenté de manière fluide et répond aux exigences du projet.

Réponses aux questions

Partie 1:

1- Si RSA est utilisé correctement et avec de très grand nombres premiers, Eve ne pourra pas en venir à bout car cet algorithme fonctionne avec la factorisation de grands nombres premiers. et actuellement, il n'est toujours pas possible de venir à bout de ce chiffrement car il n'existe pas d'algorithme possible de factoriser rapidement de grands nombres premiers.

2- L'algorithme SDES est peu sécurisé en raison de sa taille de clé courte (10 bits), ce qui signifie qu'il y a seulement 2^10 (1024) clés possibles. Une attaque par force brute, où toutes les clés possibles sont testées, devient donc réalisable en un nombre relativement faible d'essais(2^10). Ceci peut être réalisé rapidement avec les ordinateurs que nous avons actuellement.

3- Double SDES n'est pas significativement plus sûr car les clés utilisées dans les deux étapes sont encore trop courtes pour offrir une sécurité robuste.Pour attaquer Double SDES plus rapidement qu'avec une méthode de force brute, Eve pourrait utiliser une attaque de type meet-in-the-middle. Cela implique de chiffrer le message avec toutes les clés possibles pour la première étape, puis déchiffrer avec toutes les clés possibles pour la deuxième étape, stocker les résultats intermédiaires, et chercher les deux messages qui sont identiques.

Le nombres d'essaie maximum pour trouver la clef correspond donc a 2*(2^10) soit encore quelque chose de très rapide pour un ordinateur récent.

Partie 2:

1- Non, c'est généralement considéré comme très sécurisé. La taille de la clé AES de 256 bits offre une sécurité suffisamment élevée pour la plupart des applications.

2- Pour estimer le temps nécessaire pour le cassage d'AES, il faut tout d'abord ce rendre compte du nombre de possibilités. il y a 2^256 possibilités ce qui représente 1.1579209e+77 clés différentes. Or, dans notre cas nous avons un processeur Intel Core i5-13500 qui offre une performance avec une fréquence allant de 4,5 à 4,99 GHz.
Ce qui représente environ 4,5 milliard d'opérations à la seconde. Donc 1.1579209e+77/ 4 500 000 000 égale environ 2,573157556 e+67 de seconde, ou 2,978191616e+62 jours .

3- En effet, il existe plusieurs types d'attaques autres que le cassage brut (brute force) pour compromettre un système de chiffrement. L'une de ces méthodes est appelée l'attaque par dictionnaire (dictionary attack).

Dans une attaque par dictionnaire, l'attaquant utilise une liste préétablie de mots, de phrases courantes, ou de combinaisons possibles de mots de passe comme "dictionnaire". L'attaquant teste ensuite chaque entrée du dictionnaire comme clé potentielle pour le déchiffrement.

Cette méthode est particulièrement efficace lorsque les utilisateurs choisissent des mots de passe faibles ou communs. Elle exploite la tendance humaine à utiliser des mots ou des combinaisons faciles à deviner
Il existe aussi rainbow tables , ou bien Man-in-the-Middle attack comme autre attaque avec des fonctionnalités différentes.

Partie 4:

1- Le fait que Alice et Bob utilisent la même clé et très souvent une mauvaise pratique en cryptographie car quelqu'un qui veut intercepter une discussion il n'aura à trouver qu'une seule clé ce qui est donc plus simple pour lui.

2- Le protocole PlutotBonneConfidentialité semble être inspiré du protocole SSL/TLS (Secure Sockets Layer/Transport Layer Security) utilisé pour sécuriser les communications sur Internet.
La partie associée à la certification des clés dans SSL/TLS est généralement gérée par un processus de vérification des certificats par des tiers de confiance. Ce processus garantit l'authenticité des clés publiques utilisées dans l'échange de clés asymétriques.

PlutotBonneConfidentialité semble ne pas intégrer cette vérification des certificats, ce qui pourrait entraîner des vulnérabilités potentielles.

3- WhatsApp : WhatsApp intègre également le chiffrement de bout en bout. Il utilise le protocole Signal (Signal Protocol) pour chiffrer les messages, les appels vocaux et les appels vidéo, assurant ainsi la confidentialité des communications.
Signal : Signal utilise le chiffrement de bout en bout pour garantir la confidentialité des messages. Il utilise le protocole de signal, qui est basé sur le protocole Double Ratchet pour assurer la confidentialité et la forward secrecy.

4-
Arguments en faveur:
- aiderait à lutter contre les cremes en ligne tels que la pedophilie et le terrorisme.
- permettrait de détecter les activités illégales et donc dans un même temps alerté les autorités plus rapidement de ces activitées

Argument contre:
- pourrait compromettre la vie privée des utilisateurs

- l’Etat et les entreprise pourrait utiliser ses conversations pour surveiller les communication de ses utilisateurs
- effet dissuasif sur les utilisateurs qui veulent protéger leur vie privée et leur sécurité en ligne.

Lien du git : https://github.com/LeniChabilan/SAE-CRYPTO