Travaux Pratiques de cryptographie

LEDUC Alexis

ENSEA IS 2025

Introduction

Ce TP est composé de 5 exercices indépendants portant sur différents aspects de la cryptographie : fonctions de hachage, certificats numériques, cryptanalyse du chiffre de Vigenère, blockchain et reverse engineering. Chaque exercice peut être traité séparément. Votre compte rendu et vos livrables seront rendus à la deadline du 21 décembre 2025 23h59 à l'adresse mail suivante : alexis.leduc.j+ensea@gmail.com

1 Exercice 1 : Découverte des fonctions de hachage (MD5 et SHA-1)

1.1 Objectif

Comprendre le fonctionnement et les propriétés des fonctions de hachage cryptographiques.

1.2 Travail à réaliser

1. Rendez-vous sur le site `https://gchq.github.io/CyberChef/` ou dans un fichier python en installant `pip install pycryptodome`
2. Testez les fonctions de hachage **MD5** et **SHA-1** avec les entrées suivantes :
   — `"ENSEA"`
   — `"eNSEA"`
   — `"eNSeA"`
   — `"EN5EA"`
   Question 1.1 : Que remarquez-vous concernant les hash générés ? Commentez la sensibilité de ces fonctions aux modifications mineures.
3. Essayez maintenant de passer en paramètres un texte très long (plusieurs paragraphes).
   Question 1.2 : Quelle est la taille du hash obtenu ? Que pouvez-vous en déduire sur la propriété de compression des fonctions de hachage ?
4. Modifiez un seul caractère dans votre texte long et recalculez le hash.
   Question 1.3 : Quel phénomène observez-vous ? Comment appelle-t-on cette propriété en cryptographie ?
5. **Question 1.4 :** Recherchez et expliquez pourquoi MD5 et SHA-1 ne sont plus considérés comme sûrs aujourd'hui. Quelles sont les alternatives recommandées ?
6. **Question 1.5 :** Qu'est-ce qu'une fonction de hachage salée (salted hash) ? Pourquoi est-ce important pour le stockage des mots de passe ?

2 Exercice 2 : Analyse d'un certificat numérique

2.1 Objectif

Comprendre la structure et le rôle des certificats X.509 dans les communications sécurisées.

2.2 Travail à réaliser

1. Choisissez un site web de votre choix (par exemple : `google.com`, `github.com`, votre université, etc.)
2. Récupérez son certificat SSL/TLS :
   — Sur un navigateur web, cliquez sur le cadenas dans la barre d'adresse
   — Accédez aux informations du certificat
   — Exportez ou visualisez le certificat
3. **Question 2.1 :** Identifiez et expliquez les champs suivants du certificat :
   — Émetteur
   — Sujet
   — Période de validité (Not Before / Not After)
   — Clé publique et algorithme associé
   — Algorithme de signature
   — Empreinte (Fingerprint)
4. **Question 2.2 :** Qu'est-ce qu'une autorité de certification (CA) ? Quel est son rôle dans l'infrastructure à clés publiques (PKI) ?
5. **Question 2.3 :** Expliquez la chaîne de certification. Remontez jusqu'au certificat racine (Root CA) de votre certificat.
6. **Question 2.4 :** Que se passe-t-il si un certificat est révoqué ? Comment vérifier la révocation d'un certificat (OCSP, CRL) ?
7. **Question 2.5 :** Comparez les extensions présentes dans le certificat (Subject Alternative Name, Key Usage, etc.). À quoi servent-elles ?
8. **Question 2.6 :** Qu'est-ce qu'un certificat auto-signé ?

3 Exercice 3 : Cryptanalyse du chiffre de Vigenère

3.1 Objectif

Apprendre à cryptanalyser un texte chiffré par le chiffre de Vigenère en utilisant l'indice de coïncidence. Vous pouvez vous aider de ce site : `https://www.apprendre-en-ligne.net/crypto/stat/ic.html`

3.2 Rappels théoriques

Le chiffre de Vigenère est un chiffrement polyalphabétique qui utilise une clé répétée. Pour le cryptanalyser sans connaître la clé, on utilise l'**indice de coïncidence** (IC).

3.2.1 Formule de l'indice de coïncidence

Notations :
— N = nombre total de lettres dans le texte
— n_1 = nombre de A dans le texte
— n_2 = nombre de B dans le texte
— n_3 = nombre de C dans le texte
— ...
— n_{26} = nombre de Z dans le texte

Principe de calcul : L'indice de Coïncidence (IC) est la probabilité que deux lettres choisies aléatoirement dans un texte soient identiques.

$$IC = \sum_{i=1}^{26} \frac{n_i(n_i - 1)}{N(N-1)} = \frac{\sum_{i=1}^{26} n_i(n_i - 1)}{N(N-1)}$$

où i parcourt les 26 lettres de l'alphabet (A à Z).

3.2.2 Valeurs de référence

— Pour un texte en français (distribution naturelle) : $IC \approx 0.0778$
— Pour un texte aléatoire équidistribué (26 lettres) : $IC = \frac{1}{26} \approx 0.0385$

3.2.3 Fréquences des lettres en français

E	15,87%	N	7,15%	D	3,39%	Q	1,06%	H	0,77%
A	9,42%	R	6,46%	M	3,24%	G	1,04%	Z	0,32%
I	8,41%	U	6,24%	P	2,86%	B	1,02%	X	0,30%
S	7,90%	L	5,34%	C	2,64%	F	0,95%	Y	0,24%
T	7,26%	O	5,14%	V	2,15%	J	0,89%	K,W	0%

3.3 Texte chiffré à analyser

```
JRDMCQLEGASNAHSHJEVWAVGJSUDWPNUPELWGUAJFZWQRFXVWMWNNIZZWYFKMCML IKWVCQUIQW
GRGHXBYVAXZMRXILQUMGSXIWFWRFGOZWYAWJOQKTBMVVWLVRLVADSMY JIMIJUHSFLM NSHKEVMR
JNAXPZWYIWHEXWVFWZEZSRPWITLWPBYMQCWTBMVDMUSQWVCM EIFKEGM KIPJIT JJEIGTOCJ
ZBLVELWXRJQIVSXVGRLI UVLHXOOJECZMEMKXHFERBSRPAINYMM EWQOVLINDENBAUHAXE
NWPVUMTILMBFW VWMW ZSMTZAWRRQAQFXRFENBDIFTESMKHHULINXVREINB VI IAKEVWVRUISGKXREI
AMLIVFZEVLINMWEQRMREISQWGYWFRINSCGYRINSVJTEZUIPWQYALIEW PA KJCCLENIDCFW HEUSRQW
HETSTNLMEVUI SWPIKAXNLMOVKZBMWADWDYWWCWETRLINKWWAWGEAKEVJIS XGYE USNBARHWV
DIFWPWXTMNSVWFRINSRFGOZW
```

3.4 Travail à réaliser

1. **Question 3.1 :** Calculez l'indice de coïncidence du texte chiffré complet. Que pouvez-vous en déduire sur le type de chiffrement utilisé ?
2. **Question 3.2 :** Pour déterminer la longueur de la clé k :
   — Divisez le texte en sous-séquences en prenant une lettre tous les k caractères (pour $k = 1, 2, 3, ...$)
   — Calculez l'IC de chaque sous-séquence
   — La bonne valeur de k est celle pour laquelle l'IC moyen des sous-séquences est proche de 0.0778

   Quelle est la longueur probable de la clé ?
3. **Question 3.3 :** Une fois la longueur de la clé trouvée, pour chaque position de la clé :
   — Isolez les lettres correspondantes (1 lettre tous les k caractères)
   — Effectuez une analyse fréquentielle
   — Comparez avec les fréquences du français

   Déterminez la clé de chiffrement.
4. **Question 3.4 :** Déchiffrez le message complet.
5. **Question 3.5 :** Expliquez pourquoi l'indice de coïncidence est un outil efficace pour attaquer le chiffre de Vigenère.
6. **Question 3.6 :** Vous avez déchiffrer le message via l'indice de coïncidence. Tentez une autre approche, via la méthode de Kasiski.
7. **Question 3.7 :** Proposez une clé `one-time pad` qui garantie la confidentialité du message.
8. **Question 3.8 :** Quel est l'inconvénient de cette clé ?
9. **Question 3.9 :** Nous souhaitons réemettre un nouveau message avec cette nouvelle clé. Puis-je réutiliser cette clé ?

4 Exercice 4 : Exploration de la blockchain Bitcoin

4.1 Objectif

Se familiariser avec le fonctionnement de la blockchain Bitcoin en analysant concrètement son architecture, ses mécanismes de consensus et ses propriétés cryptographiques.

4.2 Ressources

— `https://www.blockchain.com`
— `https://mempool.space`
— Documentation Bitcoin : `https://bitcoin.org/bitcoin.pdf`

4.3 Analyse de la structure d'un bloc

Connectez-vous sur l'un des explorateurs de blockchain mentionnés ci-dessus et examinez plusieurs blocs récents ainsi que les premiers blocs de la chaîne.

4.4 Questions

4.4.1 Structure du bloc

1. **Question 4.1 :** Quels sont les éléments principaux qui composent un bloc Bitcoin ?
2. **Question 4.2 :** Quelle est la taille moyenne d'un bloc ? Existe-t-il une limite de taille ?
3. **Question 4.3 :** Combien de transactions un bloc peut-il contenir en moyenne ?

4.4.2 En-tête du bloc (Block Header)

1. **Question 4.4 :** Listez les champs présents dans l'en-tête d'un bloc.
2. **Question 4.5 :** Quel est le rôle du champ « Previous Block Hash » ?
3. **Question 4.6 :** À quoi sert le champ « Nonce » ?

4.4.3 Processus de formation d'un bloc

1. **Question 4.7 :** Décrivez les étapes de la création d'un nouveau bloc.
2. **Question 4.8 :** Qu'est-ce que le minage et quel est son rôle dans la formation d'un bloc ?
3. **Question 4.9 :** Qu'est-ce que la « difficulté » (difficulty) et comment évolue-t-elle ?
4. **Question 4.10 :** Combien de zéros initiaux (en moyenne) le hash d'un bloc valide doit-il contenir actuellement (à date nous sommes au bloc N°924561) ?

4.4.4 Récompenses et Halving

1. **Question 4.11 :** Quel était le montant de la première récompense Bitcoin pour le minage d'un bloc ?
2. **Question 4.12 :** Examinez le bloc genesis (bloc #0). Quelle est sa particularité ?
3. **Question 4.13 :** Quel est le message inscrit dans le bloc #0 ?
4. **Question 4.14 :** À quel numéro de bloc a eu lieu le premier halving ?
5. **Question 4.15 :** Quelle était la nouvelle récompense après ce premier halving ?
6. **Question 4.16 :** Tous les combien de blocs se produit un halving ?
7. **Question 4.17 :** Sachant qu'un bloc est créé environ toutes les 10 minutes, calculez approximativement tous les combien d'années a lieu un halving.

8. **Question 4.18 :** Combien de halvings ont déjà eu lieu à ce jour ?
9. **Question 4.19 :** Quelle est la récompense actuelle par bloc ?
10. **Question 4.20 :** En quelle année approximativement le dernier bitcoin sera-t-il miné ?
11. **Question 4.21 :** Calculez le nombre total maximum de bitcoins qui pourront être créés.

4.4.5 Garanties temporelles et consensus

1. **Question 4.22 :** Quel est le temps moyen ciblé entre deux blocs dans Bitcoin ?
2. **Question 4.23 :** Comment le protocole garantit-il que ce temps reste constant en moyenne ?
3. **Question 4.24 :** Tous les combien de blocs la difficulté est-elle réajustée ?
4. **Question 4.25 :** Si la puissance de calcul du réseau double, que se passe-t-il ?
5. **Question 4.26 :** Par combien la puissance de calcul du réseau doit être multipliée pour rajouter un 0 pour valider un bloc ?
6. **Question 4.27 :** Pourquoi faut-il attendre plusieurs confirmations pour considérer une transaction comme définitive ?
7. **Question 4.28 :** Combien de confirmations sont généralement recommandées pour une transaction importante ?
8. **Question 4.29 :** Qu'est-ce qu'une « attaque des 51% » et comment menace-t-elle la sécurité de la blockchain ?

4.4.6 Aspects cryptographiques

1. **Question 4.30 :** Quelle fonction de hachage est utilisée dans Bitcoin ?
2. **Question 4.31 :** Quelle est la taille (en bits) d'un hash Bitcoin ?
3. **Question 4.32 :** Le hash d'un bloc est-il calculé sur l'ensemble du bloc ou seulement sur son en-tête ?
4. **Question 4.33 :** Observez plusieurs hashs de blocs. Que remarquez-vous concernant leur format ?
5. **Question 4.34 :** En quoi consiste exactement la preuve de travail dans Bitcoin ?
6. **Question 4.35 :** Écrivez mathématiquement la condition que doit satisfaire le hash d'un bloc valide.
7. **Question 4.36 :** Estimez le nombre moyen de tentatives (variations du nonce) nécessaires pour trouver un bloc valide actuellement (à date nous sommes au bloc N°924561).

4.4.7 Analyse pratique

Exercice pratique : Choisissez un bloc récent sur l'explorateur et renseignez les informations suivantes :

1. Numéro du bloc : _____
2. Hash du bloc : _____
3. Hash du bloc précédent : _____
4. Timestamp (date et heure) : _____
5. Nombre de transactions : _____
6. Taille du bloc : _____
7. Difficulté : _____
8. Nonce : _____
9. Récompense totale du mineur (block reward + fees) : _____
10. Pool de minage ayant trouvé ce bloc : _____

4.4.8 Questions de synthèse

1. **Question 4.37 :** Expliquez comment la structure en chaîne de blocs garantit l'immutabilité des transactions passées.
2. **Question 4.38 :** Pourquoi dit-on que Bitcoin est un système décentralisé ? Quels acteurs participent au réseau ?
3. **Question 4.39 :** Calculez le débit maximum théorique de transactions par seconde (TPS) du réseau Bitcoin. Comparez avec des systèmes de paiement traditionnels comme Visa.
4. **Question 4.40 :** Discutez des implications environnementales du mécanisme de preuve de travail. Quelles sont les alternatives proposées (ex : Proof of Stake) ?
5. **Question 4.41 :** Analysez l'évolution du hashrate total du réseau sur les 6 derniers mois. Qu'observe-t-on et quelles conclusions peut-on en tirer ?

5 Exercice 5 : Analyse forensique d'une extension Chrome pour les crypto-actifs

5.1 Objectif

Comprendre les mécanismes de protection cryptographique utilisés par Metamask pour sécuriser les clés privées et analyser la chaîne complète de dérivation des clés.

5.2 Données récupérées

Voici l'extrait du fichier JSON récupéré (cf fichier associé) :


```
{
  "ciphertext": "aGQt3HsqhFFf886oMWW9D0jGr7EiTZYSL1o29fwTp0Zg8U9V/KPM+VVfq86a
          CZxhRWS0XoLXLAtn+RCmysOWs04hNHRI7h87zAIpgIpIlpCgX2KEaAa+h3im
          Q9uueSLzpjxOijiCVwc69eqTJtntbr2dlgFM6nG6a62OLDa+A/XaiBBc2pzc
          p42DyReTccaAqbcUk6RknbLhDxjCcg4V+Eeocu3mJVpqlvKsl7wud4xJt0qX
          VaYf9sNWBrmWNGSzS8ta5aPI+/ypZveV7Hq0kBHJN6tVaGT800oElguxKr9p
          5xMO4WJPny6x32qNAbLGKZaG3E1BqvjT04aYY/P7NX05jNXYu6g43OSGVSxN
          ..." ,
  "iv": "V52NqKHOL8C0jbNYOHItgw==",
  "keyMetadata": {
    "algorithm": "PBKDF2",
    "params": {
      "iterations": 600000
    }
  },
  "salt": "JR53k2vFWO11bPrXZLcCYEE01fxSQhTy/8oWaco0bIs="
}
```


5.3 Analyse du format de données

1. **Question 5.1 :** Identifiez et expliquez le rôle de chaque champ du fichier JSON :
   — `ciphertext` :
   — `iv` :
   — `algorithm` :
   — `iterations` :
   — `salt` :
2. **Question 5.2 :** Dans quel encodage sont représentées les valeurs `ciphertext`, `iv` et `salt` ? Quelle est la taille en octets de `iv` et `salt` ?

5.4 Analyse du code source

Metamask est un projet open-source disponible sur GitHub (`https://github.com/MetaMask/metamask-extension`).

1. **Question 5.3 :** Recherchez dans le code source de Metamask quel algorithme de chiffrement symétrique est utilisé pour protéger les données.
2. **Question 5.4 :** Expliquez la différence entre les modes de chiffrement suivants :
   — AES-CBC
   — AES-GCM
   — AES-CTR
   Quel mode utilise Metamask et pourquoi ?

3. **Question 5.5 :** À partir du code source ou de la documentation, décrivez le processus complet de déchiffrement :
   (a) Entrées nécessaires :
   (b) Étapes de dérivation de clé :
   (c) Algorithme de déchiffrement :
   (d) Sorties obtenues :

5.5 Dérivation de clé avec PBKDF2

PBKDF2 (Password-Based Key Derivation Function 2) est une fonction de dérivation de clé définie dans RFC 2898.

1. **Question 5.6 :** Pourquoi utilise-t-on un nombre d'itérations élevé (600 000 dans notre cas) ?
2. **Question 5.7 :** Quel est le rôle du sel (*salt*) dans PBKDF2 ? Pourquoi est-il essentiel pour la sécurité ?
3. **Question 5.8 :** Supposons que le mot de passe utilisé soit `"123PetitsChats"`. Écrivez un programme (Python - pip install pycryptodome) qui :
   (a) Décode le sel en Base64
   (b) Applique PBKDF2 avec SHA-256
   (c) Utilise 600 000 itérations
   (d) Génère une clé de 256 bits (32 octets)

5.6 Graines mnémoniques (BIP39)

Une *seed* (graine) est une séquence de mots qui permet de régénérer toutes les clés privées d'un portefeuille.

1. **Question 5.9 :** Qu'est-ce que le standard BIP39 ? Expliquez son fonctionnement en détaillant :
   — La génération de l'entropie
   — La création de la phrase mnémonique
   — La conversion en seed binaire
2. **Question 5.10 :** Combien de mots peut contenir une phrase mnémonique BIP39 ? Quelle est la relation entre le nombre de mots et la sécurité ?
3. **Question 5.11 :** Une fois les données déchiffrées, vous obtiendrez une phrase mnémonique comme l'exemple suivant :

   ```
   witch collapse practice feed shame open despair creek road again ice least
   ```

   Expliquez comment cette phrase se matérialise en une seed de 512 bits (64 octets). Quel algorithme est utilisé ?
4. **Question 5.12 :** Implémentez un programme qui :
   (a) Prend en entrée la phrase mnémonique
   (b) Utilise PBKDF2 avec la passphrase `"mnemonic"` comme sel.
   (c) Utilise l'algorithme défini précédemment.
   (d) Fait 2048 itérations
   (e) Génère la seed de 512 bits
5. **Question 5.13.1 :** Donnez les 64 premiers octets de la seed en hexadécimal.
6. **Question 5.13.2 :** Décodez le portefeuille.

5.7 Dérivation hiérarchique (BIP32/BIP44)

1. **Question 5.14 :** Qu'est-ce qu'un portefeuille HD (Hierarchical Deterministic) ? Quels sont ses avantages ?
2. **Question 5.15 :** Expliquez la notation de chemin de dérivation. Que signifie :

   `m/44'/60'/0'/0/0`

   Décomposez chaque niveau et expliquez sa signification.
3. **Question 5.16 :** À partir de la seed calculée précédemment :
   (a) Calculez la clé privée maître (*master private key*)
   (b) Dérivez la clé pour le chemin `m/44'/60'/0'/0/0` (compte Ethereum)
   (c) Donnez la clé privée en hexadécimal
4. **Question 5.17 :** La courbe elliptique utilisée est `secp256k1`. Rappelez :
   — L'équation de la courbe
   — Les paramètres du générateur G
   — L'ordre du groupe n

5.8 Calcul des clés publiques et adresses

1. **Question 5.18 :** À partir de la clé privée d obtenue, calculez la clé publique correspondante sur la courbe elliptique secp256k1.
2. **Question 5.19 :** Calculez l'adresse Ethereum du portefeuille à partir de la clé publique.
3. **Question 5.20 :** Vérifiez votre résultat en utilisant un outil en ligne ou une bibliothèque cryptographique (web3.js, ethers.js, ou web3.py).

5.9 Analyse forensique des transactions blockchain

Une fois les adresses publiques dérivées selon le chemin BIP44 (`m/44'/60'/0'/0`), il est possible d'analyser l'historique des transactions sur la blockchain Ethereum pour reconstituer l'activité du portefeuille.

5.9.1 Extraction des adresses du portefeuille

1. **Question 5.21 :** À l'aide de la bibliothèque `HDWallet` (Python), générez les 5 premières adresses dérivées du portefeuille selon les chemins :
   — `m/44'/60'/0'/0/0`
   — `m/44'/60'/0'/0/1`
   — `m/44'/60'/0'/0/2`
   — `m/44'/60'/0'/0/3`
   — `m/44'/60'/0'/0/4`
2. **Question 5.22 :** Pour chaque adresse générée, vérifiez son activité sur la blockchain Ethereum en utilisant :
   — Etherscan (`https://etherscan.io`)
   — API Etherscan (`https://api.etherscan.io`)
   — Ou un nœud Ethereum (web3.py)

5.9.2 Recherche et documentation des transactions

Pour chaque adresse active identifiée, documentez les informations suivantes :

1. **Question 5.23 :** Créez un tableau forensique pour chaque transaction :

Champ	Valeur
Hash de transaction	
Adresse source	
Adresse destination	
Montant (ETH)	
Montant (USD à l'époque)	
Date et heure (UTC)	
Numéro de bloc	
Gas utilisé	
Prix du gas (Gwei)	
Frais totaux (ETH)	
Statut	Success / Failed
Type de transaction	Transfer / Contract Call / Contract Creation
Input Data	

2. **Question 5.24 :** Pour chaque transaction identifiée, capturez les éléments de preuve suivants :
   — Capture d'écran de la transaction sur Etherscan
   — Export JSON de la transaction via l'API
   — Vérification du bloc contenant la transaction
   — Horodatage blockchain vs horodatage système

5.9.3 Analyse des patterns de transactions

1. **Question 5.25 :** Analysez les caractéristiques temporelles :
   — Quelle est la première transaction du portefeuille ? (date, montant, source)
   — Quelle est la dernière transaction ? (date, montant, destination)
   — Identifiez les périodes d'activité intense
   — Y a-t-il des patterns temporels récurrents ? (heures, jours de la semaine)
2. **Question 5.26 :** Calculez les statistiques financières :
   — Solde actuel de chaque adresse
   — Volume total des transactions entrantes (ETH)
   — Volume total des transactions sortantes (ETH)
   — Frais totaux payés en gas
   — Nombre total de transactions
3. **Question 5.27 :** Identifiez les contreparties :
   — Listez toutes les adresses ayant interagi avec le portefeuille
   — Identifiez les exchanges (Binance, Coinbase, Kraken, etc.)
   — Identifiez les smart contracts utilisés (DEX, NFT, DeFi)
   — Recherchez les adresses étiquetées (labeled addresses)

5.9.4 Marqueurs forensiques

1. **Question 5.28 :** Identifiez les marqueurs forensiques suivants :
   Marqueurs d'identité :
   — Première source de financement (CEX, faucet, autre wallet)
   — Utilisation de services KYC (exchanges centralisés)
   — Réutilisation d'adresses vs nouvelles adresses
   — Liens avec des adresses connues ou étiquetées
   Marqueurs comportementaux :
   — Fréquence des transactions
   — Montants typiques transférés

— Horaires d'activité (fuseau horaire probable)
— Utilisation de round numbers vs montants précis
— Temps moyen entre les transactions

Marqueurs techniques :
— Nonce sequence (détection de transactions parallèles)
— Prix du gas payé (utilisateur pressé ou patient)
— Utilisation de gas limit standard ou personnalisé
— Présence de data dans les transactions
— Type de signature (EIP-155, EIP-2930, EIP-1559)

Marqueurs de mixing/anonymisation :
— Utilisation de mixers (Tornado Cash, etc.)
— Transactions vers des bridges cross-chain
— Utilisation de privacy coins
— Patterns de dispersion (splitting)

2. **Question 5.29 :** Analysez la confidentialité du portefeuille :
   — Le portefeuille réutilise-t-il les mêmes adresses ?
   — Y a-t-il des liens évidents entre les différentes adresses dérivées ?
   — Des transactions regroupent-elles plusieurs adresses (common input ownership) ?
   — Le portefeuille interagit-il avec des services désanonymisants ?

5.9.5 Graphe de transactions et visualisation

1. **Question 5.30 :** Créez une visualisation du graphe de transactions :
   — Utilisez un outil comme GraphViz ou NetworkX (Python)
   — Représentez les adresses comme des nœuds
   — Représentez les transactions comme des arêtes dirigées
   — Annotez les montants et les dates
   — Mettez en évidence les adresses d'exchanges identifiés
2. **Question 5.31 :** Générez une timeline des activités :
   — Créez un graphique temporel des transactions
   — Identifiez les périodes d'inactivité
   — Corréllez avec des événements externes (prix ETH, actualités crypto)

5.9.6 Rapport forensique

1. **Question 5.32 :** Rédigez un rapport forensique structuré comprenant :

   (a) **Résumé exécutif**
   — Adresses identifiées
   — Période d'activité
   — Volume total de transactions
   — Principales conclusions

   (b) **Méthodologie**
   — Outils utilisés
   — Sources de données
   — Processus de vérification

   (c) **Résultats détaillés**
   — Liste complète des transactions
   — Analyse des contreparties
   — Marqueurs forensiques identifiés

   (d) **Chaîne de custody**
   — Source des données chiffrées

   - Méthode de déchiffrement
   - Horodatage de l'analyse
   - Hash des éléments de preuve

(e) **Conclusions et recommandations**

2. **Question 5.33 :** Pour chaque élément de preuve numérique collecté :
   - Calculez son hash SHA-256
   - Documentez la date et l'heure de collecte
   - Notez la source (URL API, bloc number, etc.)
   - Conservez une copie brute (JSON) et une copie formatée

5.9.7 Aspects légaux et éthiques

1. **Question 5.34 :** Discutez des considérations légales :
   - Quelle est la légalité de l'analyse de blockchain publique ?
   - Quelles sont les limites de l'utilisation de ces informations ?
   - Comment respecter le RGPD lors d'une analyse forensique ?
   - Dans quels cas peut-on associer une adresse à une identité ?
2. **Question 5.35 :** Problématiques éthiques :
   - Transparence vs vie privée sur blockchain
   - Responsabilité de l'analyste forensique
   - Utilisation des données à des fins d'investigation légitime

5.10 Sécurité et recommandations

1. **Question 5.36 :** Analysez les vulnérabilités potentielles du système :
   - Force brute du mot de passe
   - Attaques par dictionnaire
   - Extraction de la mémoire
2. **Question 5.37 :** Proposez des améliorations pour renforcer la sécurité d'un portefeuille Metamask.
3. **Question 5.38 :** Comparez PBKDF2 avec d'autres fonctions de dérivation modernes (Argon2, scrypt). Quelle serait la meilleure option aujourd'hui ?

Ressources complémentaires

Exercice 1

— CyberChef : `https://gchq.github.io/CyberChef/`
— NIST sur les fonctions de hachage : `https://csrc.nist.gov/`

Exercice 2

— Documentation sur les certificats X.509 : RFC 5280
— Let's Encrypt documentation : `https://letsencrypt.org/docs/`

Exercice 3

— Cryptanalyse de Vigenère : méthode de Kasiski
— Indice de coïncidence : `https://www.apprendre-en-ligne.net/crypto/stat/ic.html`
— Méthode de Kasiski : `https://www.apprendre-en-ligne.net/crypto/vigenere/decodevig.html`
— CyberChef pour les tests : `https://gchq.github.io/CyberChef/`

Exercice 4

— Explorateurs blockchain : `https://www.blockchain.com`, `https://mempool.space`
— Bitcoin Whitepaper : `https://bitcoin.org/bitcoin.pdf`
— Documentation Bitcoin : `https://developer.bitcoin.org/`

Exercice 5

— Code source Metamask : `https://github.com/MetaMask/metamask-extension`
— BIP39 specification : `https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki`
— BIP32 specification : `https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki`
— BIP44 specification : `https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki`
— Documentation web3.py : `https://web3py.readthedocs.io/`