Département Génie Informatique (DGI) | Ecole Supérieure Polytechnique (ESP) | UCAD

Veille technologique

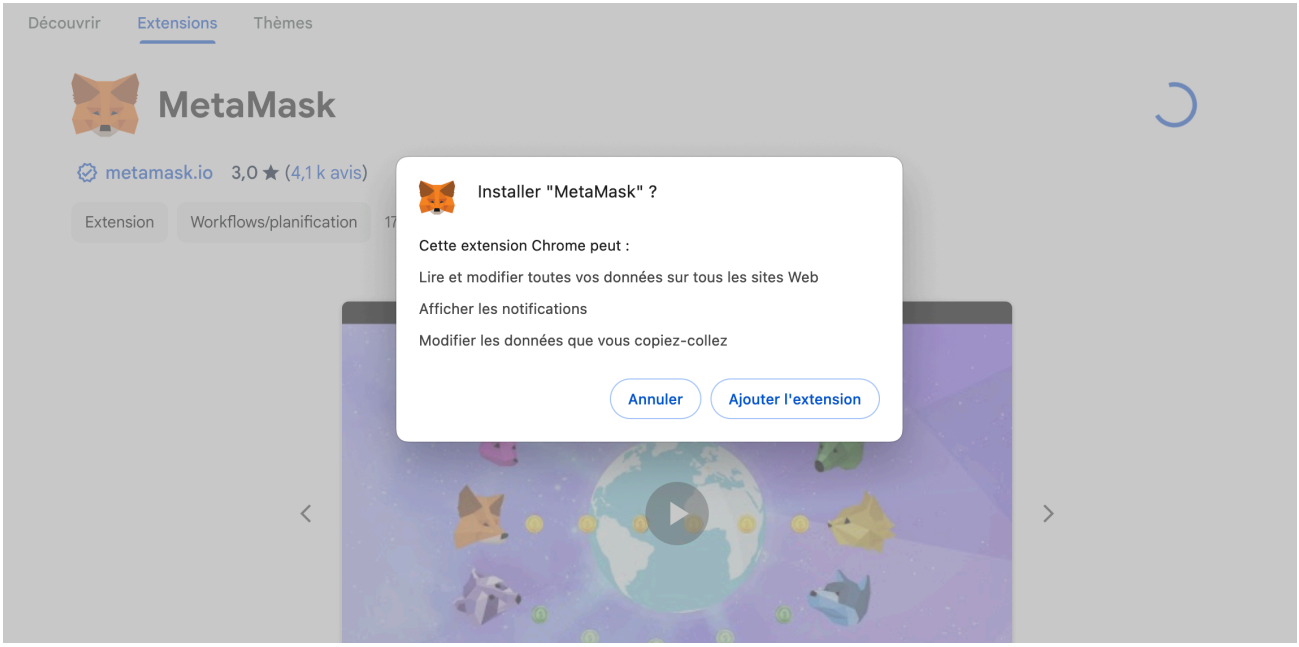
Crypto monnaies et Blockchain Atelier 4 : MetaMask, Ethereum et les Blocs

1. Introduction

Afin de nous donner une compréhension profonde du fonctionnement des blocs dans une application réelle, cet atelier va montrer la logique derrière la construction de blocs en utilisant le portefeuille MetaMask et en créant des transactions Ethereum. Si vous souhaitez trouver un guide plus complet sur MetaMask, vous pouvez consulter les documents de MetaMask sur https://docs.metamask.io/guide/#why-metamask.

En créant des comptes Ethereum et en effectuant plusieurs transactions dans MetaMask, nous examinerons en profondeur certaines propriétés des transactions Ethereum et du cryptage afin de comprendre pleinement l'authenticité et la sécurité des transactions Ethereum.

2. MetaMask 2.1 Introduction



MetaMask est un portefeuille de crypto Ethereum sous forme de plug-in pour les utilisateurs du navigateur Chrome. Disponible sous forme d'extension de navigateur et d'application mobile, MetaMask nous équipe d'un coffre-fort de clés, d'une connexion sécurisée et d'un portefeuille de jetons – tout ce dont nous avons besoin pour gérer nos actifs numériques. MetaMask offre le moyen le plus simple et le plus sûr de se connecter aux applications basées sur la blockchain. Dans cet atelier et le suivant, nous utiliserons MetaMask pour stocker et envoyer des jetons non seulement entre nos comptes, mais aussi dans les contrats intelligents qui seront explorés.

2.2 Configuration de MetaMask

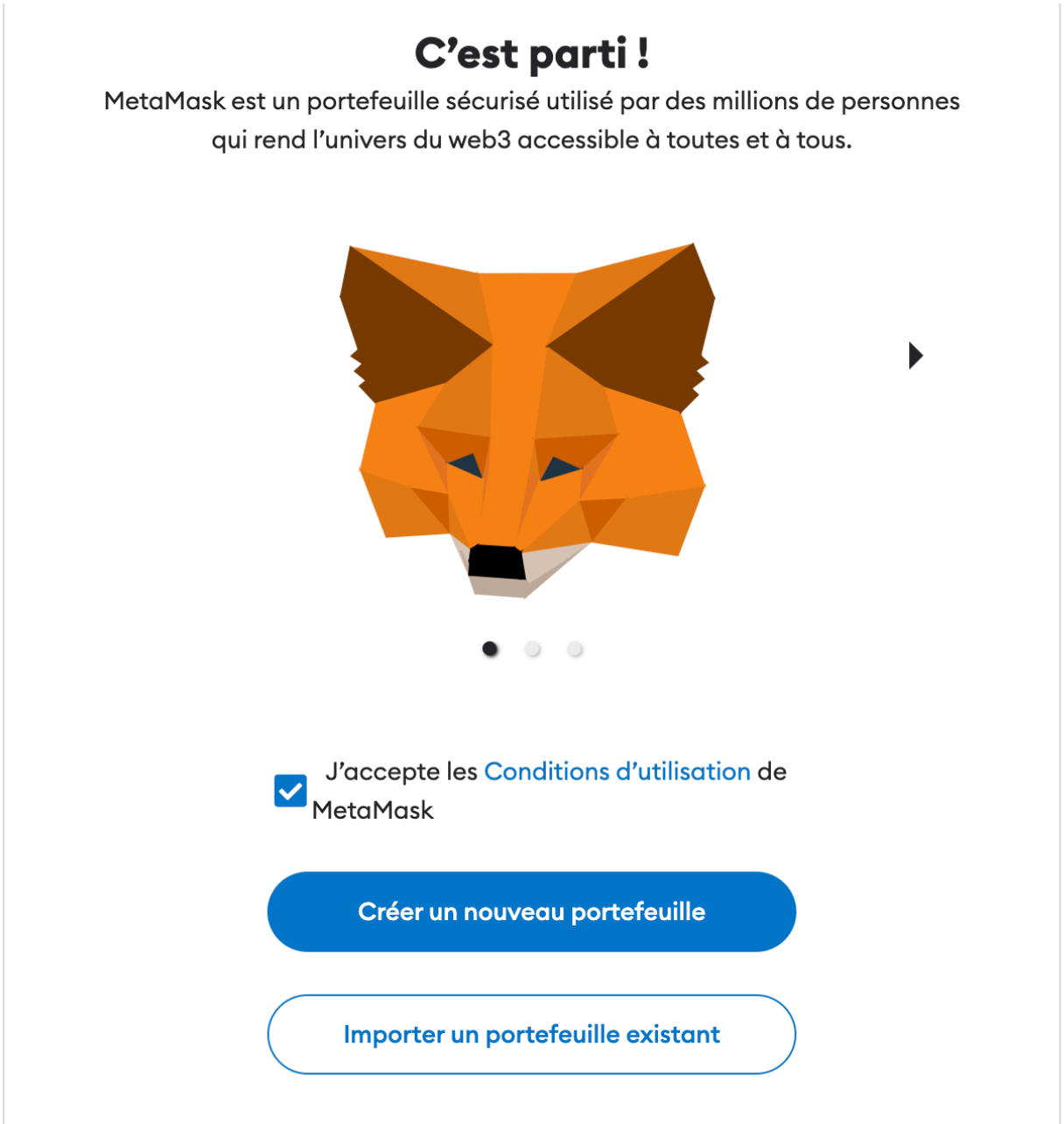
Des informations complètes et un guide d'étude sur MetaMask peuvent être trouvés sur son site officiel https://metamask.io . Nous devons choisir le bon navigateur (Chrome est recommandé) et suivre les instructions d'installation. Voici quelques points clés auxquels nous devons faire attention, lorsque nous créons de nouveaux comptes MetaMask.

Tout d'abord, la création d'un nouveau mot de passe fort est extrêmement importante car il crypte la clé privée. Les clés privées donnent accès à tous nos Ether ou autres jetons, il est donc préférable d'avoir un mot de passe fort.

La phrase secrète de sauvegarde, qui comprend 12 mots mnémotechniques, apparaîtra après la configuration du mot de passe. Nous devons écrire cette phrase sur un morceau de papier ou la

stocker dans un endroit sûr parce que la phrase de sauvegarde secrète facilite la sauvegarde et la restauration de notre compte si nous nous déconnectons de notre compte ou effaçons accidentellement l'historique du navigateur.

Nous sommes maintenant en mesure d'interagir avec MetaMask.



1 Créer un mot de passe — 2 Portefeuille sécurisé — 3 Confirmer la phrase secrète de récupération

Créer un mot de passe

Ce mot de passe permet de déverrouiller votre portefeuille MetaMask uniquement sur cet appareil. MetaMask ne peut pas récupérer ce mot de passe.

Nouveau mot de passe (min 8 caractères) Afficher

1 Créer un mot de passe — 2 Portefeuille sécurisé — 3 Confirmer la phrase secrète de récupération

Confirmer la phrase secrète de récupération

Confirmer la phrase secrète de récupération

1. behind	2. jealous	3.
4.	5. act	6. alter
7. also	8.	9. duck
10. neglect	11. attend	12. eye

Confirmer

2.3 Dépôt d’Ether

Les étapes suivantes peuvent être effectuées sur le site Web de MetaMask ou sur l'interface de son extension (nous pouvons accéder à l'interface à partir de la barre d'outils de l'extension du navigateur, qui se trouve dans le coin supérieur gauche de Chrome). Tout d'abord, nous devons choisir un réseau approprié pour effectuer notre première transaction. Il existe plusieurs options pour les réseaux : réseau principal (main network, mainnet), hôte local, RPC personnalisé et réseaux de test, qui comprennent Ropsten, Kovan, Rinkeby et Goerli. Le réseau par défaut est le réseau principal d’Ethereum où nous pouvons échanger des Ether réels. Dans cet atelier et le suivant, nous allons utiliser le réseau de test Rinkeby (vous pouvez également utiliser d'autres réseaux de test) dans lequel nous pouvons utiliser des Ether de test gratuit à partir de sites Web tiers. Tester sur le réseau principal peut coûter trop d'Ether réels. En cliquant sur "Main Ethereum Network" sur le site ou l'interface, il nous donnera plusieurs choix et nous allons choisir "Rinkeby Test Network" pour notre environnement de développement.

Déposez de l'Ether sur votre compte MetaMask.

En suivant les étapes sur faucet.rinkeby.io, toute personne ayant un compte Twitter ou Facebook peut demander des Rinkeby Ether dans les limites autorisées. Si vous souhaitez utiliser des jetons sur d'autres réseaux, vous pouvez faire recherches sur Google ou cliquer sur "Deposit" dans l'interface de MetaMask.

2.4 Effectuer une transaction

Dans cette section, nous allons effectuer une transaction entre nos comptes. Nous n'avons qu'un seul compte par défaut pour le moment, mais nous devons absolument en créer d'autres.

En cliquant sur l'image du compte en haut à droite de l'interface MetaMask, nous pouvons voir le bouton 'Create Account'. En cliquant sur ce bouton et en entrant le nom du compte, nous retournerons sur le compte où nous avons déposé de l'Ether, en cliquant sur 'Send', nous serons appelé.e.s à fournir une quantité d'Ether, puis à sélectionner le compte vers

lequel nous voulons transférer, et nous choisirons la vitesse d'envoi de l'Ether. En fonction de la vitesse choisie, la transaction prend généralement entre 15 et 30 secondes.

Département Génie Informatique (DGI) | Ecole Supérieure Polytechnique (ESP) | UCAD

Pendant que nous attendons qu'elle soit terminée, nous pouvons retrouver cette transaction dans la "Queue" (ou dans "History" si la transaction est terminée). En cliquant sur "View on Etherscan", nous trouverons les détails de la transaction, notamment le hachage de la transaction, le statut, le bloc, l'horodatage, la date de début et de fin, la valeur de la transaction et les frais de transaction. Comme nous pouvons le voir, notre transaction a été enregistrée dans le réseau Rinkeby, et n'importe qui dans le réseau Rinkeby peut voir nos blocs.

Créez plusieurs comptes et effectuez quelques transactions entre ces comptes.

3. Ethereum:transaction 3.1 Introduction

Dans cette section, nous allons examiner plus en profondeur le fonctionnement des transactions Ethereum et comment s'assurer que ces transactions sont authentiques. En analysant et en comprenant un exemple classique de transaction Ethereum, nous comprendrons pleinement la composition et l'authenticité de la transaction.

3.2 Préparation programmatique de la transaction

Si nous regardons n'importe quelle bibliothèque qui peut être utilisée pour envoyer de manière programmatique une transaction au réseau, comme la bibliothèque **Web3.js** que nous allons utiliser plus tard, nous verrons qu'un objet de transaction contient plusieurs paramètres. Certains d'entre eux sont obligatoires et d'autres sont facultatifs. Regardons de plus près un objet de transaction classique qui ressemble à la transaction que nous avons envoyée dans la section 2.4.

a) **from** : le compte d'où nous envoyons l'Ether ;
b) **to** : (optionnel) le compte sur lequel nous envoyons l'Ether ;
c) **value** : (optionnel) montant d'Ether que nous envoyons en Wei (1 Ether = 10^18 Wei) ; d) **gas** : (optionnel) quantité maximale de gas dans une transaction ;
e) **gasPrice** : (optionnel) montant que l'expéditeur paie par étape de calcul ;
f) **data** : (facultatif) chaînes d'octets ABI ;
g) **nonce** : (facultatif) nombre entier d'un nonce.

Les trois premiers éléments sont relativement faciles à comprendre. Le gas est une unité spéciale dans Ethereum et il fait référence au coût nécessaire pour effectuer une transaction. Le gas Ethereum fonctionne de manière similaire à l'essence dans la voiture. La limite de gas détermine la quantité de gas que nous pouvons utiliser dans une transaction, et si nous avons utilisé tout le gas, la transaction entière sera terminée. La limite de gas fonctionne de la même manière que le réservoir de carburant, le volume du réservoir de carburant ne peut pas être modifié une fois

que la voiture est produite et il en va de même pour la limite de gas qui ne peut pas être modifiée une fois que la transaction commence. Le porte-monnaie MetaMask fixera automatiquement la limite de gas à 21 000 unités lors de transferts simples, mais lorsque la transaction devient plus complexe ou nécessite plus d'étapes de calcul, la limite de gas doit augmenter en conséquence. Selon les différentes étapes de calcul et les fonctions que nous appelons, le prix du gas pour une transaction varie de zéro à quelques milliers de Gwei (1 Gwei = 10^9 Wei). Si vous souhaitez connaître le prix spécifique du gas, veuillez consulter l'annexe G. Fee Schedule sur le site :

https://ethereum.github.io/yellowpaper/paper.pdf

Les données sont une chaîne d'octets ABI (Application Binary Interface) utilisée dans les Smart Contracts lorsqu'ils sont déployés sur la blockchain.

Nous discuterons des données en détail lorsque nous commencerons à concevoir le Smart Contract. Nonce est un nombre entier qui est incrémenté à chaque fois qu'une transaction est envoyée afin d'éviter les attaques par rejeu.

3.3 Ethereum : signature de la transaction

Dans cette section, nous allons élargir nos vues en analysant l'authenticité d'une transaction Ethereum. Les équations suivantes jouent un rôle important dans la production d'une transaction authentique :
a) Un objet de transaction + Clé privée => Transaction signée

b) Clé privée + Algorithme de signature numérique à courbe elliptique (ECDSA) => Clé publique c) Clé publique + Hash Keccak=> Compte Ethereum
d) Transaction signée + ECRECOVER => Compte Ethereum

Comme nous l'avons vu dans la Section 2, un objet de transaction contient différents paramètres. L'algorithme de signature numérique à courbe elliptique (ECDSA) garantit que, outre le propriétaire de la clé privée, personne ne peut générer la clé privée du propriétaire à partir de son compte Ethereum. Si vous êtes intéressé par l'ECDSA, vous trouverez plus d'informations à l'adresse suivante :

https://cryptobook.nakov.com/digital-signatures/ecdsa-sign-verify-messages

La fonction de hachage Keccak est plus connu sous le nom de SHA-3. Elle est largement utilisée pour l'authentification, le cryptage et la génération de nombres pseudo-aléatoires. Pour une explication sur la façon dont Ethereum utilise la fonction de hachage Keccak, en particulier Keccak-256, pour générer l'adresse du compte Ethereum, veuillez consulter la réponse qui est donnée sur le lien suivant :

https://ethereum.stackexchange.com/questions/3542/how-are-ethereum-addresses-generated

Veille technologique

ECRECOVER est l'acronyme de Elliptic Curve Recover Function et transforme la transaction signée en compte Ethereum. Si vous souhaitez obtenir plus d'informations sur son fonctionnement, veuillez consulter le site suivant :

https://gist.github.com/axic/5b33912c6f61ae6fd96d6c4a47afde6d

Nous pouvons générer un compte Ethereum de deux manières différentes qui peuvent nous aider à vérifier l'authenticité d'une transaction Ethereum.

Ajoutez des transactions à une blockchain.

Pour des raisons de sûreté et de sécurité, nous voulons ajouter des transactions à une blockchain que seul le propriétaire de la clé privée peut créer. Regardez la vidéo du tutoriel sur le premier lien et exercez-vous en visitant les autres liens.

https://www.youtube.com/watch?time_continue=1&v=xIDL_akeras&feature=emb_logo https://andersbrownworth.com/blockchain/public-private-keys/keys https://andersbrownworth.com/blockchain/public-private-keys/signatures https://andersbrownworth.com/blockchain/public-private-keys/transaction https://andersbrownworth.com/blockchain/public-private-keys/blockchain