Engineer

Chiffrement

transformation rendant les données illisibles sans la clé

Ce que c’est

Le chiffrement, que certaines publications du NIST désignent aussi par le terme anglais enciphering, transforme des données lisibles, le texte en clair, en texte chiffré en leur appliquant un algorithme publié et une valeur secrète appelée clé. Avec la bonne clé, le déchiffrement restitue le texte en clair, octet pour octet. L’algorithme est public délibérément : AES, l’algorithme de chiffrement utilisé par la plupart des systèmes, a remporté un concours ouvert du NIST en 2000 et fait l’objet d’une norme librement accessible. Seule la clé est secrète.

Les algorithmes de chiffrement appartiennent à deux familles. Un algorithme symétrique emploie la même clé dans les deux sens. AES, spécifié dans FIPS 197, chiffre les données par blocs de 128 bits avec une clé de 128, 192 ou 256 bits. ChaCha20, défini dans la RFC 8439, est un algorithme de chiffrement par flot environ trois fois plus rapide qu’AES sur du matériel dépourvu d’instructions AES. Les algorithmes symétriques sont assez rapides pour un disque entier, mais les deux extrémités doivent posséder la même clé ; la leur transmettre en sécurité est la partie difficile.

La cryptographie à clé publique répond à cette difficulté par une paire de clés : l’une publiée, l’autre conservée secrète. Deux parties peuvent chacune combiner leur clé privée avec la clé publique de l’autre pour obtenir le même secret partagé sans jamais le transmettre. Les opérations à clé publique sont lentes ; les systèmes réels associent donc les deux techniques : un échange à clé publique permet de convenir d’une nouvelle clé symétrique, puis l’algorithme symétrique transporte les données. TLS, le protocole derrière chaque adresse `https://`, fonctionne ainsi.

L’idée moderne la plus importante est l’authentification. Un algorithme de chiffrement seul dissimule le contenu, mais le laisse modifiable. En mode compteur, inverser un bit du texte chiffré inverse le même bit dans le texte en clair, et le destinataire déchiffre le message altéré comme s’il était authentique. Le chiffrement authentifié avec données associées, AEAD, ajoute une étiquette d’authentification que le destinataire vérifie avant de faire confiance au moindre octet. AES-GCM et ChaCha20-Poly1305 sont les deux constructions courantes. TLS 1.3, publié sous la forme de la RFC 8446 en 2018, n’a conservé que les algorithmes AEAD et a imposé la confidentialité persistante à chaque échange à clé publique : une clé de serveur volée plus tard ne permet donc de déchiffrer aucune session enregistrée.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que le chiffrement ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des comptes, mots de passe et droits attribués selon les besoins exacts de chaque poste, puis retirés le jour du départ.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. 1 000 règles RBAC pour sécuriser Linux — 1,000 règles d'accès écrites pour combler une faille de sécurité

1 000 règles RBAC pour sécuriser Linux Contactez-nous →

Pourquoi c’est important

Les données restent rarement sur du matériel contrôlé par leur propriétaire. Elles traversent les réseaux d’autres personnes, résident dans le centre de données d’un tiers et quittent les locaux dans des ordinateurs portables. Le chiffrement permet de les faire passer par ces endroits en sécurité : un disque volé, un câble mis sur écoute ou un bucket dont le contenu a fuité ne livrent que du texte chiffré. Sans la clé, ce texte n’est que du bruit.

Cela déplace le problème plutôt que de le supprimer : une fois les données chiffrées, leur sécurité dépend de celle de la clé. La recommandation du NIST sur la gestion des clés, SP 800-57, les compare à la combinaison d’un coffre : un adversaire qui connaît cette combinaison franchit la protection du coffre le plus solide, et une mauvaise gestion des clés peut facilement compromettre un algorithme robuste. L’essentiel du travail réel porte donc sur les clés : les stocker à part des données, limiter ce que chacune protège, les remplacer selon un calendrier et protéger leur copie de secours aussi soigneusement que l’original.

Les autorités de réglementation font la même distinction. Le RGPD cite la pseudonymisation et le chiffrement des données personnelles parmi les mesures de sécurité de l'article 32. L'article 34 permet au responsable du traitement de ne pas informer individuellement chaque personne touchée par une violation lorsque les données divulguées étaient inintelligibles pour toute personne non autorisée à les consulter ; le chiffrement est l'exemple donné.

Le chiffrement s'applique à deux endroits. Le chiffrement en transit, principalement TLS, protège les données sur le réseau. Le chiffrement au repos les protège dans le stockage. Leur utilisation échappe aux deux : un serveur qui déchiffre un enregistrement pour le traiter détient à la fois le texte en clair et la clé. Quiconque contrôle ce serveur peut donc lire ce qu'il lit.

À quoi cela ressemble en pratique

En transit, le travail quotidien consiste à maintenir la validité des certificats. Lorsque cent applications Docker avaient besoin de certificats SSL/TLS, un flux de travail ACME créait chaque certificat, le renouvelait avant son expiration et le déployait sur des hôtes Ubuntu exécutant Apache et Nginx. L'objectif était de retirer la personne de cette boucle, car c'est elle qui oublie, et un certificat expiré rend une application indisponible aussi sûrement qu'un algorithme de chiffrement défaillant.

Au repos, la question utile est de savoir qui détient la clé. L'implémentation de sauvegardes chiffrées hors de l'hôte de l'entreprise a été conçue pour étendre la reprise au-delà d'un instantané du fournisseur. Elle utilise restic, qui chiffre ce qu'il stocke avec AES-256 en mode compteur et l'authentifie avec Poly1305-AES. Ses scripts permettent d'envoyer les instantanés par SFTP vers un stockage chez un autre fournisseur. Dans cette conception, le fournisseur reçoit du texte chiffré et l'entreprise conserve la phrase secrète dans un fichier protégé. Cela décrit les contrôles implémentés, pas leur activation opérationnelle : le relevé daté du 2026-10-27 indique que les temporisateurs étaient désactivés faute de dépôt configuré et ne consigne aucune restauration réelle achevée. Le chiffrement est une propriété du parcours de stockage proposé ; prouver que la sauvegarde est récupérable exige d'exécuter le test de restauration distinct.

Même le chemin suivi par la phrase secrète avait de l'importance. Elle est lue depuis un fichier, car le superviseur interprète les séquences d'échappement dans les valeurs d'environnement. Une phrase secrète contenant une barre oblique inversée aurait donc différé silencieusement de celle ayant servi à créer

le dépôt. Le processus implémenté pour le test mensuel de restauration est conçu pour restaurer un fichier connu depuis le dépôt et le comparer au résultat attendu. Puisque le test exige la phrase secrète, une exécution réussie vérifierait aussi que la clé permet toujours d'accéder aux données.

Lorsque la plateforme allait recevoir un second hôte, le même raisonnement a déterminé comment chaque secret opérationnel a été séparé par hôte. La chaîne désignant un dépôt de sauvegardes indique un emplacement. Des dépôts restic distincts peuvent avoir des clés maîtresses différentes même s'ils utilisent la même phrase secrète. Une personne qui obtient cette phrase et peut lire les données chiffrées ainsi que les fichiers de clés des deux dépôts peut déverrouiller les deux : la partager lie donc leur exposition. Chaque secret est devenu un dictionnaire dont les clés désignent les hôtes, avec le raisonnement écrit à côté de la configuration. SP 800-57 formule le principe général : limiter ce qu'une clé protège limite les dégâts si elle fuit.

Où les problèmes surviennent

La clé n'existe qu'à un seul endroit. Un algorithme de chiffrement robuste traite le propriétaire qui a oublié le mot de passe exactement comme un attaquant. La documentation de restic le dit clairement : une fois le mot de passe perdu, les données le sont aussi, sans possibilité de récupération. Tout stockage chiffré a besoin d'une seconde copie de sa clé, conservée hors de portée de ce qui détruirait la première.

La clé se trouve à côté des données. Une base chiffrée dont la clé se trouve dans la configuration du même serveur, ou une sauvegarde dont la phrase secrète réside dans le même bucket, n'arrête qu'un voleur qui emporte la moitié des fichiers. Le chiffrement de disque est une version moins évidente du problème : il protège un disque retiré d'une machine éteinte, tandis qu'un serveur en fonctionnement a déjà déverrouillé son disque. Un attaquant présent sur ce serveur lit le texte en clair.

La confidentialité est prise pour de l'intégrité. Un système fondé sur un mode de chiffrement seul dissimule le contenu tout en laissant un attaquant le modifier. Employez une construction AEAD et rejetez tout message dont l'étiquette d'authentification échoue à la vérification, avant d'agir sur le moindre de ses octets.

Un nonce est réutilisé. AES-GCM et ChaCha20-Poly1305 utilisent un nonce, une valeur qui doit rester unique pour chaque message sous une même clé. La RFC 8439 en fait la considération de sécurité la plus importante : deux messages utilisant le même nonce partagent le même flot de clés et une clé d'authentification à usage unique, ce qui révèle le XOR des deux textes en clair. Un compteur est un moyen sûr de produire des nonces, tant qu'un seul compteur est associé à la clé. Deux processus partageant une clé et comptant chacun à partir de zéro produisent précisément la répétition contre laquelle la RFC met en garde.

Un mot de passe remplace une clé. Les personnes choisissent des mots de passe ; les algorithmes de chiffrement ont besoin de clés aléatoires d'une longueur fixe. Une phrase secrète passe d'abord par une fonction de dérivation de clé, comme le fait restic avec scrypt. Cette fonction, spécifiée dans la RFC 7914, est conçue pour exiger beaucoup de mémoire, afin que les essais massifs sur du matériel spécialisé restent coûteux. Les mots de passe de connexion stockés constituent un cas distinct : ils doivent passer par une telle fonction pour être conservés sous forme d'empreintes, puisqu'un serveur capable de les déchiffrer peut aussi les divulguer.

Les métadonnées restent visibles. Le chiffrement dissimule le contenu ; la taille, le moment des échanges et les adresses des deux extrémités restent visibles. TLS 1.3 permet à l'expéditeur de compléter les enregistrements par du bourrage pour masquer leur taille. Ce bourrage est facultatif ; la longueur en révèle donc souvent davantage que le propriétaire ne le pense.

Pour aller plus loin

- Glossaire du NIST : chiffrement — toutes les définitions employées dans les publications du NIST, chacune accompagnée de son document source.
- FIPS 197 : Advanced Encryption Standard (AES) — la spécification AES, sa taille de bloc et ses trois longueurs de clé.
- NIST SP 800-57, partie 1, rév. 5 : recommandation sur la gestion des clés — la création, le stockage, l'utilisation et le retrait des clés, et les conséquences d'une fuite.
- RFC 8446 : protocole Transport Layer Security (TLS), version 1.3 — le protocole derrière `https://`, avec des enregistrements exclusivement AEAD et des échanges de clés assurant la confidentialité persistante.
- RFC 8439 : ChaCha20 et Poly1305 pour les protocoles de l'IETF — un algorithme de chiffrement par flot, un mécanisme d'authentification et leur association AEAD, avec les règles relatives aux nonces.

- restic : références — comment un véritable outil de sauvegarde dérive les clés d'un mot de passe, puis chiffre et authentifie ses données.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que le chiffrement ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/encryption/