(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** ***3 141 538***

(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **22 11306**

(51) Int Cl[8] : ***G 06 F 16/10*** *(2023.01), G 06 F 21/78, 21/64, 21/31*

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 28.10.22.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 03.05.24 Bulletin 24/18.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Se reporter à la fin du présent fascicule*

(60) **Références à d'autres documents nationaux apparentés :**

( ) **Demande(s) d'extension :**

(71) **Demandeur(s) :** *ASTRACHAIN Société par Actions Simplifiée* — FR.

(72) **Inventeur(s) :** SEGHAIER Gilles, DOGET Julien et CORIAT Florent.

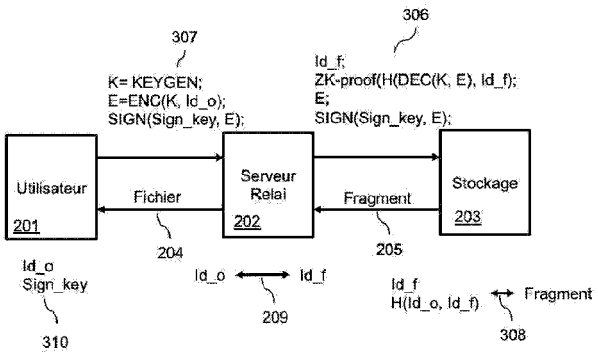
(73) **Titulaire(s) :** ASTRACHAIN Société par Actions Simplifiée.

(74) **Mandataire(s) :** SANTARELLI.

(54) PROCEDE ET DISPOSITIF DE STOCKAGE EN LIGNE REPARTI DE FICHIERS DANS UN CONTEXTE ZERO CONFIANCE.

(57) La présente invention concerne un procédé et un système de stockage sécurisé zéro connaissance géré par un serveur relai connecté à une pluralité de services de stockage en ligne, le fichier à stocker étant divisé en une pluralité de fragments sauvegardés sur la pluralité de services de stockage en ligne, chaque service de stockage en ligne étant à même de vérifier l'identité de l'utilisateur requérant un fichier, l'identifiant du fichier original et le lien entre l'identifiant du fichier original et l'identifiant de fragment sans acquérir de connaissance sur l'identifiant du fichier original.

[Fig. 3b]



FR 3 141 538 - A1

Description

Titre de l'invention : PROCEDE ET DISPOSITIF DE STOCKAGE EN LIGNE REPARTI DE FICHIERS DANS UN CONTEXTE ZERO CONFIANCE

[0001] La présente invention concerne le domaine du stockage de fichiers informatiques en ligne auprès d'une pluralité de services de stockage dans un contexte où la sécurité du service n'implique pas de relation de confiance entre les différents acteurs.

[0002] Les services de stockage en ligne de fichiers informatiques se sont développés ces dernières années. Les grands noms de l'informatique, proposent presque tous leur propre service. Aux côtés de ses grands acteurs, de multiples sociétés ont également développé leur propre service à destination tant des particuliers que des entreprises. Ces dernières sont nombreuses aujourd'hui à adopter ces services pour sauvegarder leurs données.

[0003] L'utilisation d'un service de stockage en ligne permet en effet d'éviter la mise en place et la gestion au quotidien d'un stockage, de ses sauvegardes et de sa sécurité. Il permet également de mutualiser les coûts du service entre les différents clients.

[0004] La sécurité informatique est également un sujet de plus en plus sensible. Les exemples de piratage d'entreprises et de services en ligne se multiplient. Même sans intention malveillante, un exemple récent d'incendie dans un centre de données d'un important acteur de services en ligne a pu provoquer la perte de leurs données pour un nombre important de ses clients.

[0005] Dans ce contexte, l'amélioration de la sécurité et de la fiabilité des services de stockage en ligne est un souci constant.

[0006] L'invention se place dans le contexte où un service sécurisé de stockage en ligne est géré par un serveur relai (*proxy server* en anglais) qui fait le lien entre l'utilisateur du service et une pluralité de services de stockage en ligne. Le terme serveur est ici utilisé de manière fonctionnelle et peut référer à diverses implémentations physiques du serveur relai comprenant un ou plusieurs serveurs physiques. Le terme utilisateur désigne un terminal informatique utilisé par l'utilisateur humain pour accéder au service. Lorsque nous utilisons ce terme d'utilisateur dans la description des procédés proposés, il s'agit d'actions effectuées par le terminal de l'utilisateur sous son contrôle et non pas d'actions faites par l'utilisateur humain.

[0007] L'utilisateur du service sécurisé de stockage en ligne ne communique qu'avec le serveur relai et n'a pas connaissances des différents services de stockage en ligne utilisés. De même, chaque service de stockage en ligne ne communique qu'avec le serveur relai et n'a pas connaissance des autres services en ligne ni de l'utilisateur.

[0008] Le fonctionnement général du service sécurisé de stockage en ligne est illustré par la [Fig.1]. L'utilisateur 101 transmet au serveur relai 102 un fichier 106 à stocker. Le serveur relai 102 applique au fichier 106 une fonction de dispersion générant des fragments 107, 108, 109 à partir du fichier initial 106. Ces fragments sont stockés par le serveur relai 102 sur les différents services de stockage en ligne 103, 104, 105. C'est l'opération de téléversement (*upload* en anglais).

[0009] Lorsque l'utilisateur 101 veut retrouver le fichier stocké, il soumet une requête en téléchargement (*download* en anglais) au serveur relai 102. Le serveur relai 102 requiert alors les différents fragments 107, 108, 109 aux différents services de stockage en ligne 103, 104, 105. A partir de ces fragments 107, 108, 109, il reconstitue le fichier initial 106 et le transmet à l'utilisateur 101.

[0010] La figure illustre trois services de stockage en ligne. Cette figure n'est qu'un exemple et le nombre de services en ligne utilisé par un mode de réalisation particulier de l'invention peut être quelconque avec un minimum de deux services. De manière similaire, la figure illustre la sauvegarde d'un seul fragment par service de stockage en ligne. Dans la pratique, plusieurs fragments peuvent être stockés sur un même service de stockage en ligne.

[0011] Avantageusement, ces fragments comportent un certain degré de redondance pour permettre la reconstitution du fichier initial à partir seulement d'un sous-ensemble de l'intégralité des fragments disponibles pour ce fichier. Cette propriété apporte la résilience du système face à la perte potentielle ou la compromission d'un service de stockage en ligne.

[0012] Avantageusement, les communications entre l'utilisateur et le serveur relai ainsi qu'entre le serveur relai et les services de stockage en ligne utilisent des tunnels chiffrés permettant de garantir la confidentialité des échanges. Chaque acteur du système est authentifié auprès des acteurs avec lesquels il communique. L'utilisateur, en particulier, doit être authentifié auprès du système qui lui délivre un jeton d'accès qui peut être vérifié par le serveur relai et/ou les services de stockage en ligne.

[0013] L'utilisateur dispose également d'une clé de signature lui permettant de signer ses messages. Cette signature peut être vérifiée, et donc les messages authentifiés comme provenant bien de l'utilisateur par le serveur relai et les services de stockage en ligne. Cette clé de signature peut être symétrique ou asymétrique.

[0014] Avantageusement, la fonction de dispersion permet de garantir que la connaissance d'un seul fragment ne permet pas à un service de stockage en ligne de reconstituer le fichier initial. Dans le cas où le système utilise de la redondance et où plusieurs fragments sont stockés sur un même service de stockage en ligne, le système assure que le nombre de fragments stockés sur un même service de stockage en ligne ne permet pas de reconstituer le service.

[0015] En termes de sécurité, il est avantageux de proposer un service de stockage dit zéro confiance entre les différents acteurs. Ainsi la compromission de l'un des acteurs ne compromet pas la sécurité de la solution. Dans le cas présent, la zéro confiance se traduit par trois contraintes que l'on souhaite voir respecter par le système.

[0016] Selon une première contrainte, un service de stockage en ligne doit être capable de vérifier qu'une requête reçue du serveur relai provient bien de l'utilisateur légitime. Cette contrainte doit garantir que le serveur relai n'est pas en capacité de forger une requête valide qui ne proviendrait pas effectivement de l'utilisateur légitime. Ainsi, la compromission du serveur relai, par exemple par son piratage par des individus malveillants, ne permet pas aux pirates d'obtenir les fichiers des utilisateurs.

[0017] Selon une seconde contrainte, un service de stockage en ligne ne doit pas être en capacité d'identifier que deux fragments qui sont stockés par son service sont issus d'un même fichier original. Ainsi, chaque fragment est indépendant et ne peut être associé à d'autres fragments du même fichier original afin d'interdire toute reconstruction de ce fichier original, même partielle, par le service de stockage en ligne.

[0018] Selon une troisième contrainte, un service de stockage en ligne doit être en mesure de vérifier la validité du lien entre le fragment qu'il possède et le fichier original. Ceci en respectant la seconde contrainte qui ne lui permet pas d'obtenir ce lien.

[0019] L'invention vise à offrir un service sécurisé de stockage en ligne sur la base d'un serveur relai et d'une pluralité de services de stockage en ligne respectant ces contraintes.

[0020] Selon un aspect de l'invention il est proposé un procédé de stockage sécurisé d'un fichier original caractérisé en ce qu'il comprend les étapes suivantes :

- réception du fichier original, transmis par un dispositif utilisateur, par un serveur relai connecté à une pluralité de services de stockage en ligne, le fichier original étant associé à un identifiant de fichier original ;
- génération par le serveur relai d'une pluralité de fragments à partir du fichier original, chaque fragment étant associé à un identifiant de fragment ;
- transmission pour sauvegarde par le serveur relai de chaque fragment à un service de stockage en ligne parmi la pluralité de service de stockage en ligne ;

où pour télécharger le fichier original stocké, le procédé comprend les étapes suivantes :

- réception par le serveur relai d'une requête pour le fichier original comprenant l'identifiant de fichier original et un ensemble de données utilisateur ;

pour chaque fragment :

- transmission d'une requête pour le fragment comprenant l'identifiant de fragment et un ensemble de donnée de fragment au service de stockage en ligne ;
- vérification par le service de stockage en ligne, à partir des données de fragment,

sans acquérir de connaissance sur l'identifiant de fichier original, que :
- la requête provient de l'utilisateur ayant transmis la requête pour le fichier original ;
- l'identifiant de fragment de la requête de fragment correspond à l'identifiant du fichier original ;
- lorsque la vérification est positive transmission du fragment au serveur relai ;
- génération du fichier original à partir des fragments reçus ; et
- transmission du fichier original au dispositif utilisateur.

[0021] Selon un mode de réalisation :

- l'ensemble de données utilisateur et l'ensemble de données fragment comprennent une version chiffrée et signée de l'identifiant de fichier original permettant de vérifier que la requête provient de l'utilisateur ayant transmis la requête pour le fichier original.

[0022] Selon un mode de réalisation :

- le serveur relai génère une preuve zéro connaissance basée sur l'identifiant du fichier original et sur l'identifiant de fragment permettant de vérifier que l'identifiant de fragment de la requête de fragment correspond à l'identifiant du fichier original sans apporter de connaissance sur l'identifiant original, cette preuve zéro connaissance étant comprise dans l'ensemble de données fragment.

[0023] Selon un mode de réalisation :

- le dispositif utilisateur génère un premier engagement basé sur l'identifiant de fichier original, cet engagement étant compris dans les données utilisateur et dans les données fragment ;
- le serveur relai génère un second engagement conjoint basé l'identifiant de fichier original et sur l'identifiant de fragment, cet engagement conjoint étant compris dans les données fragment ;
- le service de stockage en ligne vérifie conjointement les premier et second engagements.

[0024] Selon un autre aspect de l'invention il est proposé un programme d'ordinateur comprenant des instructions adaptées à la mise en œuvre de chacune des étapes du procédé selon l'invention lorsque ledit programme est exécuté sur un ordinateur.

[0025] Selon un autre aspect de l'invention il est proposé un moyen de stockage d'informations, amovible ou non, partiellement ou totalement lisible par un ordinateur ou un microprocesseur comportant des instructions de code d'un programme d'ordinateur pour l'exécution de chacune des étapes du procédé selon l'invention.

[0026] Selon un autre aspect de l'invention il est proposé un système comprenant un dispositif utilisateur, un serveur relai et une pluralité de service de stockage en ligne configurés pour mettre en œuvre un procédé selon l'invention.

[0027] Aux dessins annexés, donnés à titre d'exemples non limitatifs :

[0028] la [Fig.1] illustre l'architecture générale et le fonctionnement d'un service sécurisé de

stockage en ligne selon un mode de réalisation de l'invention;

[0029] les figures 2a et 2b illustrent le fonctionnement général du téléversement et du téléchargement d'un fichier;

[0030] Les figures 3a et 3b illustrent un premier mode de réalisation de l'invention ;

[0031] La [Fig.4] illustre un second mode de réalisation

[0032] La [Fig.5] est un bloc-diagramme schématique d'un dispositif de traitement de l'information pour la mise en œuvre d'un ou plusieurs modes de réalisation de l'invention.

[0033] Les figures 2a et 2b illustrent le fonctionnement général du téléversement, [Fig.2a], et du téléchargement, [Fig.2b], d'un fichier.

[0034] Sur ces figures un seul service de stockage en ligne 203 est représenté. Le même fonctionnement s'applique à la pluralité de services de stockage en ligne connectés au serveur relai 202.

[0035] Le téléversement du fichier est illustré par la [Fig.2a]. L'utilisateur 201 transmet le fichier original à sauvegarder 204 au serveur relai 202. Ce dernier attribue un identifiant 207 Id_o au fichier original, le transmet à l'utilisateur 201 qui le stocke 210.

[0036] Le serveur relai applique la fonction de dispersion pour générer les fragments à partir du fichier original. Le fonctionnement de la fonction de dispersion n'est pas le sujet du présent document. La figure illustre un fragment 205 de cette pluralité de fragments. Le fragment 205 est transmis au service de stockage en ligne 203 qui lui attribue un identifiant de fragment Id_f 206 et mémorise le lien 208 entre le fragment 205 et son identifiant Id_f 206. Le service de stockage en ligne transmet cet identifiant de fragment 206 au serveur relai qui mémorise l'association 209 entre l'identifiant du fichier original Id_o et l'identifiant de fragment Id_f. Les autres fragments générés par le serveur relai reçoivent le même traitement non représenté sur la figure.

[0037] Le téléchargement illustré par la [Fig.2b] présente un fonctionnement symétrique. L'utilisateur 201 transmet au serveur relai 202 une requête comprenant l'identifiant Id_o 207 du fichier qu'il souhaite télécharger. Le serveur relai retrouve l'identifiant de fragment associé Id_f qu'il utilise pour construire une requête comprenant cet identifiant qu'il transmet au service de stockage en ligne 203. A l'aide de l'identifiant de fragment Id_f, le service de stockage en ligne 203 retrouve le fragment associé 205 et le transmet au serveur relai 202. Ce dernier, quand il a récupéré la totalité des fragments, ou au moins un nombre suffisant si la redondance est mise en œuvre, reconstruit le fichier original 204 en utilisant la fonction inverse de la fonction de dispersion. Le serveur relai peut alors transmettre le fichier original 204 ainsi obtenu à l'utilisateur.

[0038] Dans ce contexte, les trois contraintes pour atteindre le niveau de sécurité désiré zéro confiance se traduisent par les contraintes suivantes :

[0039] Selon la première contrainte, le service de stockage en ligne doit être en mesure de vérifier que la requête reçue pour le fragment provient bien de l'utilisateur légitime. Selon la seconde contrainte, le service de stockage en ligne ne doit pas être en mesure de connaître l'identifiant Id_o du fichier original. Selon la troisième contrainte, le service de stockage en ligne doit être en mesure de vérifier que l'identifiant de fragment demandé Id_f correspond bien à l'identifiant du fichier original Id_o, et cela sans connaître cet identifiant du fichier original Id_o selon la seconde contrainte.

[0040] Les solutions proposées dans ce document pour permettre de répondre aux contraintes de sécurité exposées sont basée sur l'utilisation d'outils cryptographiques. Parmi ces outils nous citerons les fonctions suivantes :

[0041] La fonction KEYGEN qui permet de générer une clé cryptographique. La clé générée peut être une clé symétrique ou asymétrique.

[0042] Une clé symétrique est une clé partagée entre deux acteurs qui permet le chiffrement d'un contenu pour obtenir un contenu chiffré. La même clé est nécessaire pour déchiffrer le contenu chiffré et retrouver le contenu d'origine. Cette clé constitue un secret partagé entre l'émetteur du contenu chiffré qui procède au chiffrement et le récepteur du contenu chiffré qui procède au déchiffrement.

[0043] Une clé asymétrique est composée d'une paire de clés, une clé dite privée et une clé dite publique. Pour échanger du contenu chiffré entre deux acteurs, chacun possède sa propre clé asymétrique. Chaque acteur garde secret sa clé privée et diffuse sa clé publique. Dans ce cas, l'émetteur qui procède au chiffrement d'un contenu chiffre celui-ci à l'aide de sa clé privée et de la clé publique du récepteur. Ce dernier déchiffre le contenu à l'aide de sa propre clé privée et de la clé publique de l'émetteur.

[0044] La fonction ENC(k, m) représente le chiffrement d'un contenu 'm' à l'aide de la clé 'k'. Il s'agit ici d'une clé symétrique, typiquement issue de la fonction KEYGEN. Une clé asymétrique peut également être utilisée.

[0045] La fonction DEC(k, c) représente le déchiffrement d'un contenu chiffré 'c' à l'aide de la clé 'k'. C'est la fonction inverse de la fonction ENC(k, m).

[0046] La fonction SIGN(k, m) produit une signature cryptographique du contenu 'm' à l'aide de la clé 'k'. Cette fonction ne chiffre pas le contenu et le laisse lisible. L'émetteur d'un contenu 'm' peut signer ce contenu à l'aide de cette fonction. Il transmet alors le contenu 'm' et la signature ainsi calculée. Le récepteur du contenu 'm' peut vérifier la signature, par exemple à l'aide de la même fonction et de la même clé ou des fonctions de vérification selon le mode de réalisation. Si la signature ainsi obtenue correspond à la signature reçue, cela signifie que le contenu 'c' n'a pas été altéré et qu'il a bien été signé par le possesseur de la clé 'k'. Dans le mode de réalisation privilégié, la clé de signature est différente de la clé utilisée pour chiffrer/déchiffrer les messages.

[0047] La fonction HASH(m) est une fonction mathématique non inversible qui produit à partir d'un contenu 'm' de longueur quelconque une valeur de taille fixe. La probabilité que deux contenus différents m1 et m2 produise la même valeur de hash est suffisamment faible pour que l'on considère la fonction HASH comme permettant de vérifier l'intégrité du contenu 'm'.

[0048] La fonction ZK-proof(m) utilisée par un producteur P, produit une preuve que P connaît le secret S, m étant le résultat de la fonction f connue appliquée à S. La preuve pouvant être vérifiée par un vérificateur V qui peut alors vérifier que P connaît S, sans qu'aucune connaissance sur S ne soit transmise à V. Différentes fonctions ZK-proof sont connues et peuvent être utilisées indifféremment dans les différents modes de réalisation de l'invention. Par exemple, des fonctions ZK-proof sont décrites dans les documents suivants : « Pinocchio: Parno, B., Howell, J., Gentry, C., and Raykova, M. Pinocchio: Nearly practical verifiable computation. In 2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013 (2013), pp. 238–252 », ou « Ligero: Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In ACM CCS 2017, pages 2087–2104. ACM Press, 2017 ».

[0049] Les figures 3a et 3b illustrent un premier mode de réalisation de l'invention. La [Fig.3a] illustre le téléversement selon ce mode de réalisation, tandis que la [Fig.3b] illustre le téléchargement.

[0050] Pour réaliser le téléversement, l'utilisateur 201, transmet le fichier original 204 à stocker au serveur relai 202. Ce dernier applique la fonction de dispersion et génère les fragments issus du fichier original. Il associe également le fichier original à l'identifiant de fichier original 207 Id_o qu'il transmet en retour à l'utilisateur 201. Nous décrivons le traitement d'un de ces fragments, le fragment 205 qui est transmis au service de stockage en ligne 203 pour stockage. Le service de stockage en ligne 203 mémorise le fragment et l'associe à un identifiant de fragment Id_f qui est renvoyé au serveur relai. Le serveur relai calcule alors un hash des deux identifiants, l'identifiant du fichier original Id_o et de l'identifiant du fragment Id_f, qui correspond à l'application de la fonction de hash aux deux identifiants : « HASH(Id_o, Id_f) » également noté en raccourci « H(Id_o, Id_f) ». Cette valeur de hash est transmise au service de stockage en ligne qui le mémorise avec l'identifiant Id_f et l'associe au fragment via l'association 308. Alternativement, l'identifiant de fragment Id_f peut être généré directement par le serveur relai qui peut alors calculer le hash directement afin d'éviter un aller-retour avec le service de stockage. L'utilisateur dispose également d'une clé de signature « Sign_key » qu'il mémorise 310 ainsi que l'identifiant de fichier original Id_o. Cette clé est typiquement une clé asymétrique grâce à laquelle, l'utilisateur peut signer E à l'aide de sa clé privée. Le serveur relai et

le service de stockage en ligne sont capables de vérifier cette signature à l'aide de la clé publique, mais ne sont pas capables de générer une éventuelle fausse signature sans connaître la clé privée de l'utilisateur.

[0051] Concernant le téléchargement, l'utilisateur génère une clé K à l'aide de la fonction KEYGEN. A l'aide de cette clé, l'utilisateur chiffre l'identifiant du fichier original Id_o pour obtenir un identifiant chiffré E=ENC(K, Id_o). Il utilise ensuite sa clé de signature pour signer l'identifiant chiffré E ainsi obtenu.

[0052] Ensuite, l'utilisateur transmet au serveur relai, référence 307, la clé K, l'identifiant chiffré E et sa signature SIGN(Sign_key, E).

[0053] Le serveur relai est alors en mesure de vérifier que l'identifiant chiffré E est bien émis par l'utilisateur en vérifiant la signature électronique de E. Il peut également dé-chiffrer l'identifiant du fichier d'origine à l'aide de la clé K. Il peut alors retrouver l'identifiant du fragment Id_f à l'aide de l'association mémorisée, référence 209, entre l'identifiant du fichier original et l'identifiant du fragment. Enfin, le serveur relai calcule la preuve ZK-proof(H(DEC(K, E), Id_f)).

[0054] Le serveur transmet alors au service de stockage en ligne, référence 306, l'identifiant Id_f du fragment demandé, la preuve ZK-proof(H(DEC(K, E), Id_f)), l'identifiant chiffré E et sa signature par l'utilisateur SIGN(Sign_key, E).

[0055] Le service de stockage en ligne est alors en mesure de vérifier la preuve ZK-proof(H(DEC(K, E), Id_f)), ce qui permet de vérifier que le serveur relai possède bien la connaissance de la clé K sans avoir accès à cette clé K. Il faut comprendre que ici, la fonction f(x) de la preuve correspond à la fonction H(DEC(x, E), Id_f). Cela permet au service de stockage en ligne de valider que la valeur chiffrée de Id_o, c'est-à-dire E a bien été utilisée pour le calcul. Il peut également vérifier que cette valeur chiffrée provient bien de l'utilisateur en vérifiant la signature de E. La valeur de hash permet quant à elle de vérifier le lien entre Id_o et Id_f, et ceci toujours sans acquérir de connaissance sur Id_o.

[0056] Il peut être constaté que les trois contraintes zéro-connaissance sont respectées par le système lors de l'étape de téléchargement. Le service de stockage en ligne peut vérifier que la requête reçue provient bien de l'utilisateur légitime par la vérification de la signature de la valeur chiffrée de Id_o, première contrainte. Le service de stockage en ligne n'obtient aucune information sur l'identifiant du fichier original Id_o, seconde contrainte. Le service de stockage en ligne peut finalement vérifier que l'identifiant de fragment demandé, Id_f, correspond bien à l'identifiant du fichier original Id_o, via le hash des deux identifiants, toujours sans connaître Id_o, troisième contrainte. Le système proposé permet également de garantir que le serveur relai ne peut pas forger une requête ne provenant pas de l'utilisateur car il ne peut pas générer une signature valide de E.

[0057] Une fois ces contraintes vérifiées, le service de stockage en ligne transmet le fragment demandé 205 au serveur relai qui reconstitue, à l'aide de l'ensemble des fragments ainsi récupérés, le fichier initial 204 pour le transmettre à l'utilisateur.

[0058] Le système proposé permet de garantir la sécurité des données des utilisateurs, même en cas de compromission du serveur relai ou d'un ou des services de stockage en ligne utilisés.

[0059] La [Fig.4] illustre un second mode de réalisation. Contrairement au premier mode de réalisation qui est basé sur une preuve de calcul (*proof of computation* en anglais), ce second mode de réalisation est basé sur une preuve de connaissance (*proof of knowledge* en anglais).

[0060] Pour éviter le stockage de Id_o, un engagement conjoint (*joint commitment* en anglais) des deux identifiants Id_o et Id_f permet au serveur relai de prouver au service de stockage en ligne le lien entre Id_o et Id_f sans révéler d'information sur Id_o. Ce faisant, il est possible de satisfaire les seconde et troisième contraintes.

[0061] Il est également possible d'utiliser un second engagement par l'utilisateur pour permettre au serveur relai de prouver au service de stockage en ligne la connaissance de l'identifiant de fichier original Id_o. L'utilisateur peut également signer ce second engagement pour apporter en plus la preuve que cet identifiant de fichier original est bien issu de l'utilisateur. La vérification séparée des deux engagements ne permet pas de vérifier que la valeur de l'identifiant du fichier original Id_o utilisée lors de chacun des engagements est la même. Pour ce faire et ainsi permettre le respect des trois contraintes, il est nécessaire de procéder à une unique étape de vérification conjointe des deux engagements. Ainsi, la vérification permet de garantir, en cas de succès, que le même identifiant Id_o a été utilisée lors de la génération des deux engagements. La vérification de cet identifiant Id_o est alors complète et permet de satisfaire également la première contrainte.

[0062] La preuve de connaissance permettant la vérification conjointe des deux engagements peut, par exemple, être basée sur un ensemble de représentations linéaires décrite par [MAU09] (Ueli M. Maurer: Unifying Zero-Knowledge Proofs of Knowledge. AFRICACRYPT 2009) qui généralise les preuves de Schnorr [SCH89] (Schnorr, C.P.: Efficient identification and signatures for smart cards. CRYPTO 1989).

[0063] Dans un exemple de réalisation, la preuve de connaissance peut fonctionner comme suit :

[0064] Soit P le prouveur et V le vérificateur. Soient les espaces $G = Z_q^4$ et $H = E^2$ ou E est un groupe cyclique d'ordre q. Soient h_1, h_2, h_3 trois générateurs publics distincts de H, par distincts il faut comprendre que leur logs discrets dans H ne doivent pas être liés mathématiquement.

[0065]

(g_1, g_2, g_3, g_4), la valeur $[(g_1, g_2, g_3, g_4)] = (h_1^{g_2} h_3^{g_4}, h_1^{g_1} h_2^{g_2} h_3^{g_3})$. Il est à noter ici que les deux composantes du couple définissant l'homomorphisme de groupe vont permettre d'exprimer chacun respectivement les deux engagements utilisés dans ce mode de réalisation. C'est la forme de cet homomorphisme de groupe qui va permettre la vérification conjointe des deux engagements permettant de valider que la même valeur de Id_o est utilisée dans le calcul des deux engagements.

[0066] Soit $x = (x_1, x_2, x_3, x_4) \in G$ un secret connu de P, mais pas de V. La valeur associée à x dans le groupe H est la valeur $[x] = (h_1^{x_2} h_3^{x_4}, h_1^{x_1} h_2^{x_2} h_3^{x_3}) \in H$. Cette valeur est supposée connue de V qui va l'utiliser pour vérifier que P est bien en possession de x. La vérification peut se dérouler de la manière suivante :

[0067] Dans une première étape, P génère un nombre aléatoire correspondant à l'aléa de mise en gage du prouveur P $k = (k_1, k_2, k_3, k_4) \in G$ et transmet à V l'engagement associé $K = [k] = (h_1^{k_2} h_3^{k_4}, h_1^{k_1} h_2^{k_2} h_3^{k_3}) \in H$.

[0068] Dans une seconde étape V génère un challenge aléatoire $c \in Z_q$, et transmet cette valeur à P. Alternativement, il est possible de rendre la preuve non interactive, en remplaçant cette étape par une transformation de Fiat-Shamir.

[0069] Dans une troisième étape P calcule :

[0070] $r = k + cx = (k_1 + cx_1, k_2 + cx_2, k_3 + cx_3, k_1 + cx_1) \in G$;

[0071] Cette valeur r est transmise à V. Ce dernier peut alors calculer la valeur correspondant à r dans H et vérifier que cette valeur $[r]$ est bien égale à $K.[x]^c$.

[0072] Cet exemple de preuve de connaissance peut être utilisée dans le système de stockage de fichiers. D'autres preuves de connaissance pourraient également être utilisées de manière similaire.

[0073] La [Fig.4] illustre un mode de réalisation basé sur les preuves de connaissances comme celles qui vient d'être décrite.

[0074] Concernant le téléversement, ce dernier est similaire au téléversement utilisé dans le mode de réalisation précédent sauf sur le calcul d'un engagement conjoint $L = h_1^{Id_f} h_2^{Id_o} h_1^{Rand_f}$ par le serveur relai des deux identifiants Id_o et Id_f. Rand_f étant un nombre aléatoire généré par le serveur relai. Cet engagement conjoint est transmis au service de stockage en ligne et mémorisé par ce dernier associé, référence 408, au fragment. La valeur aléatoire Rand_f est mémorisée, référence 409, par le serveur relai.

[0075] Concernant le téléchargement, lors d'une première étape 410, l'utilisateur calcule un premier engagement $E = h_1^{Id_o} h_3^{Rand_n}$ à l'aide d'un nombre aléatoire $Rand_n$ qu'il génère pour l'occasion. L'utilisateur signe cet engagement E à l'aide de sa clé de

signature Sign_key. L'utilisateur transmet alors l'identifiant de fichier original Id_o, la signature de l'engagement E et la valeur aléatoire Rand_n utilisée pour le calcul de l'engagement E au serveur relai.

[0076] Lors d'une étape 407, les valeurs de Id_o, de la valeur aléatoire Rand_n et la signature du premier engagement $E = h_1^{Id_o} h_3^{Rand_n}$ sont transmises au serveur relai. Il n'est pas nécessaire de transmettre l'engagement lui-même, car ce dernier peut être calculé par le serveur relai à l'aide des valeurs transmises. Il faut rappeler ici que les valeurs h_i générateur du groupe H sont supposées connues de l'ensemble des acteurs du système.

[0077] Lors de l'étape 411, le serveur relai calcule le premier engagement E, ainsi que le second engagement $L = h_1^{Id_o} h_2^{Id_f} h_3^{Rand_f}$, à l'aide d'une valeur aléatoire Rand_f qui correspond à celle utilisée lors du téléversement. Une autre valeur aléatoire $(r_1, r_2, r_3, r_4) \in G$ est générée et utilisée pour le calcul de la valeur correspondante K correspondant à l'engagement de k, $K = [k] = \left(h_1^{r_2} h_3^{r_4}, h_1^{r_1} h_2^{r_2} h_3^{r_3}\right) \in H$. Le serveur relai calcule alors la valeur $c = HASH, h_3, E, L, K$ qui correspond au challenge du vérificateur V et finalement la valeur *r=k+c.(Id_f, Id_o, Rand_f, Rand_n)* qui correspond au témoin de preuve.

[0078] Lors de l'étape 408, les valeurs Id_f, l'engagement de k K, le témoin de preuve r, E et la signature de E par l'utilisateur sont transmises au service de stockage en ligne.

[0079] Lors de l'étape 412, le service de stockage est alors en mesure de procéder aux calculs suivants :

[0080] Le service de stockage en ligne vérifie la signature du premier engagement E, ce qui permet de valider que la valeur Id_o utilisée est bien issue de l'utilisateur. Il calcule ensuite le challenge $c = HASH, h_3, E, L, K$, la valeur $Z_1 = \left(E^c, L^c\right)$, la valeur $Z_2 = [r] = h_3^{r_4+cRand_n}, h_1^{r_1+cId_f} h_2^{r_2+cId_o} h_3^{r_3+cRand_f}$. Finalement, la vérification que $KZ_1 = Z_2$ permet de vérifier de manière conjointe les deux engagements E et L, ce qui permet de valider la connaissance par le serveur relai de l'identifiant de fichier original et le lien entre cet identifiant et l'identifiant du fragment demandé Id_f. Les trois contraintes sont donc vérifiées, ce qui permet de valider le transfert du fragment 205 au serveur relai. Ce dernier peut alors, une fois qu'il a reçu tous les fragments de tous les services de stockage en ligne, reconstituer le fichier original 204 et le transmettre à l'utilisateur.

[0081] La [Fig.5] est un bloc-diagramme schématique d'un dispositif de traitement de l'information 500 pour la mise en œuvre d'un ou plusieurs modes de réalisation de l'invention. Le dispositif 500 de traitement de l'information peut être un périphérique tel qu'un micro-ordinateur, un poste de travail ou un terminal mobile de télécommu-

nication. Le dispositif 500 comporte un bus de communication connecté à :

[0082] - une unité centrale de traitement 501, tel qu'un microprocesseur, notée CPU;

[0083] - une mémoire à accès aléatoire 502, notée RAM, pour mémoriser le code exécutable du procédé de réalisation de l'invention ainsi que les registres adaptés à enregistrer des variables et des paramètres nécessaires pour la mise en œuvre du procédé selon des modes de réalisation de l'invention ; la capacité de mémoire du dispositif peut être complétée par une mémoire RAM optionnelle connectée à un port d'extension, par exemple ;

[0084] - une mémoire morte 503, notée ROM, pour stocker des programmes informatiques pour la mise en œuvre des modes de réalisation de l'invention ;

[0085] - une interface réseau 504 est normalement connectée à un réseau de communication sur lequel des données numériques à traiter sont transmis ou reçus. L'interface réseau 504 peut être une seule interface réseau, ou composée d'un ensemble d'interfaces réseau différentes (par exemple filaire et sans fil, interfaces ou différents types d'interfaces filaires ou sans fil). Des paquets de données sont envoyés sur l'interface réseau pour la transmission ou sont lues à partir de l'interface de réseau pour la réception sous le contrôle de l'application logiciel exécuté dans le processeur 501 ;

[0086] - une interface utilisateur 505 pour recevoir des entrées d'un utilisateur ou pour afficher des informations à un utilisateur ;

[0087] - un dispositif de stockage 506 tel que décrit dans l'invention et noté HD ;

[0088] - un module d'entrée/sortie 507 pour la réception / l'envoi de données depuis / vers des périphériques externes tels que disque dur, support de stockage amovible ou autres.

[0089] Le code exécutable peut être stocké dans une mémoire morte 503, sur le dispositif de stockage 506 ou sur un support amovible numérique tel que par exemple un disque. Selon une variante, le code exécutable des programmes peut être reçu au moyen d'un réseau de communication, via l'interface réseau 504, afin d'être stocké dans l'un des moyens de stockage du dispositif de communication 500, tel que le dispositif de stockage 506, avant d'être exécuté.

[0090] L'unité centrale de traitement 501 est adaptée pour commander et diriger l'exécution des instructions ou des portions de code logiciel du programme ou des programmes selon l'un des modes de réalisation de l'invention, instructions qui sont stockées dans l'un des moyens de stockage précités. Après la mise sous tension, le CPU 501 est capable d'exécuter des instructions de la mémoire RAM principale 502, relatives à une application logicielle. Un tel logiciel, lorsqu'il est exécuté par le processeur 501, provoque l'exécution des procédés décrits.

[0091] Dans ce mode de réalisation, l'appareil est un appareil programmable qui utilise un logiciel pour mettre en œuvre l'invention. Toutefois, à titre subsidiaire, la présente invention peut être mise en œuvre dans le matériel (par exemple, sous la forme d'un

circuit intégré spécifique ou ASIC).

Revendications

[Revendication 1] Procédé de stockage sécurisé d'un fichier original caractérisé en ce qu'il comprend les étapes suivantes :
- réception du fichier original, transmis par un dispositif utilisateur, par un serveur relai connecté à une pluralité de services de stockage en ligne, le fichier original étant associé à un identifiant de fichier original ;
- génération par le serveur relai d'une pluralité de fragments à partir du fichier original, chaque fragment étant associé à un identifiant de fragment ;
- transmission pour sauvegarde par le serveur relai de chaque fragment à un service de stockage en ligne parmi la pluralité de service de stockage en ligne ;
où pour télécharger le fichier original stocké, le procédé comprend les étapes suivantes :
- réception par le serveur relai d'une requête pour le fichier original comprenant l'identifiant de fichier original et un ensemble de données utilisateur ;
pour chaque fragment :
- transmission d'une requête pour le fragment comprenant l'identifiant de fragment et un ensemble de donnée de fragment au service de stockage en ligne ;
- vérification par le service de stockage en ligne, à partir des données de fragment, sans acquérir de connaissance sur l'identifiant de fichier original, que :
- la requête provient de l'utilisateur ayant transmis la requête pour le fichier original ;
- l'identifiant de fragment de la requête de fragment correspond à l'identifiant du fichier original ;
- lorsque la vérification est positif transmission du fragment au serveur relai ;
- génération du fichier original à partir des fragments reçus ; et
- transmission du fichier original au dispositif utilisateur.

[Revendication 2] Procédé selon la revendication 1, caractérisé en ce que :
- l'ensemble de données utilisateur et l'ensemble de données fragment comprennent une version chiffrée et signée de l'identifiant de fichier original permettant de vérifier que la requête provient de l'utilisateur ayant transmis la requête pour le fichier original.

[Revendication 3] Procédé selon la revendication 1, caractérisé en ce que :
- le serveur relai génère une preuve zéro connaissance basée sur l'identifiant du fichier original et sur l'identifiant de fragment permettant de vérifier que l'identifiant de fragment de la requête de fragment correspond à l'identifiant du fichier original sans apporter de connaissance sur l'identifiant original, cette preuve zéro connaissance étant comprise dans l'ensemble de données fragment.

[Revendication 4] Procédé selon la revendication 1, caractérisé en ce que :
- le dispositif utilisateur génère un premier engagement basé sur l'identifiant de fichier original, cet engagement étant compris dans les données utilisateur et dans les données fragment ;
- le serveur relai génère un second engagement conjoint basé l'identifiant de fichier original et sur l'identifiant de fragment, cet engagement conjoint étant compris dans les données fragment ;
- le service de stockage en ligne vérifie conjointement les premier et second engagements.

[Revendication 5] Programme d'ordinateur comprenant des instructions adaptées à la mise en œuvre de chacune des étapes du procédé selon l'une quelconque des revendications 1 à 4 lorsque ledit programme est exécuté sur un ordinateur.

[Revendication 6] Moyen de stockage d'informations, amovible ou non, partiellement ou totalement lisible par un ordinateur ou un microprocesseur comportant des instructions de code d'un programme d'ordinateur pour l'exécution de chacune des étapes du procédé selon l'une quelconque des revendications 1 à 4.

[Revendication 7] Système comprenant un dispositif utilisateur, un serveur relai et une pluralité de service de stockage en ligne configurés pour mettre en œuvre un procédé selon l'une des revendications 1 à 4.

[Fig. 1]

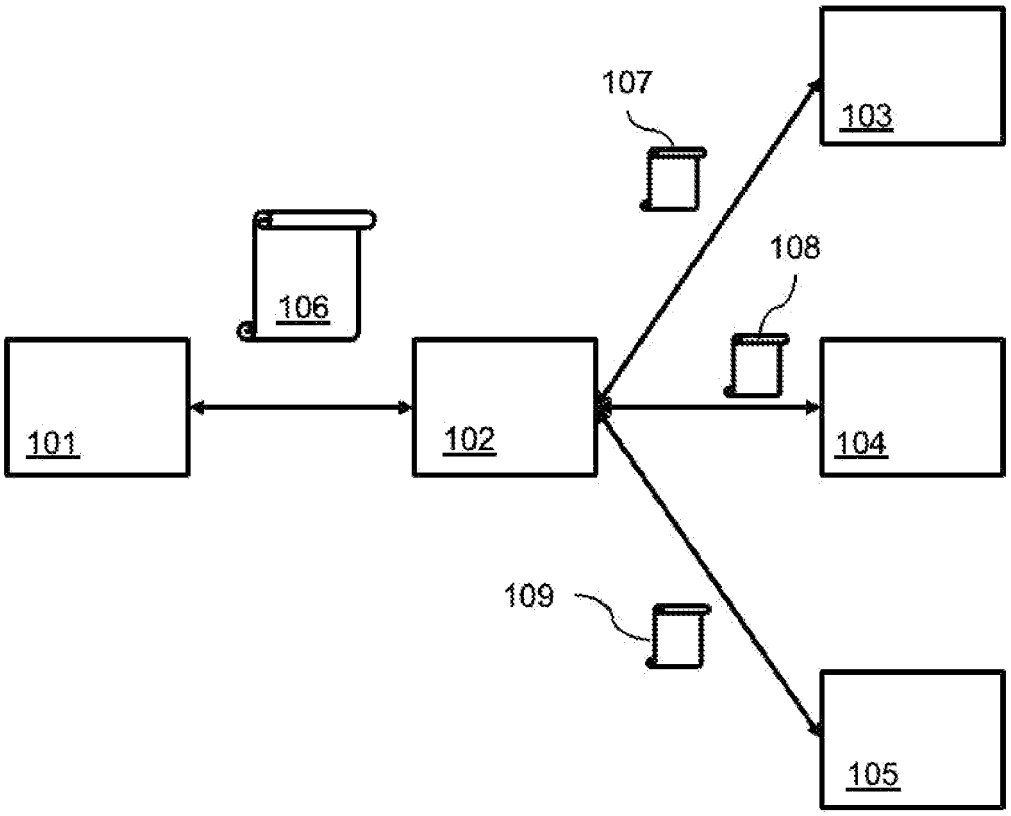


Fig. 1

[Fig. 2a]

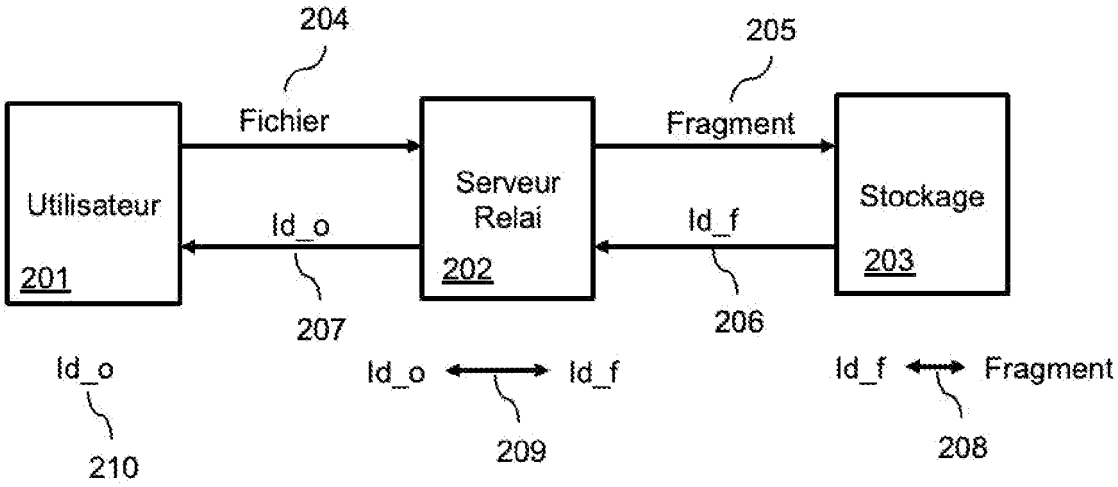


Fig. 2a

[Fig. 2b]

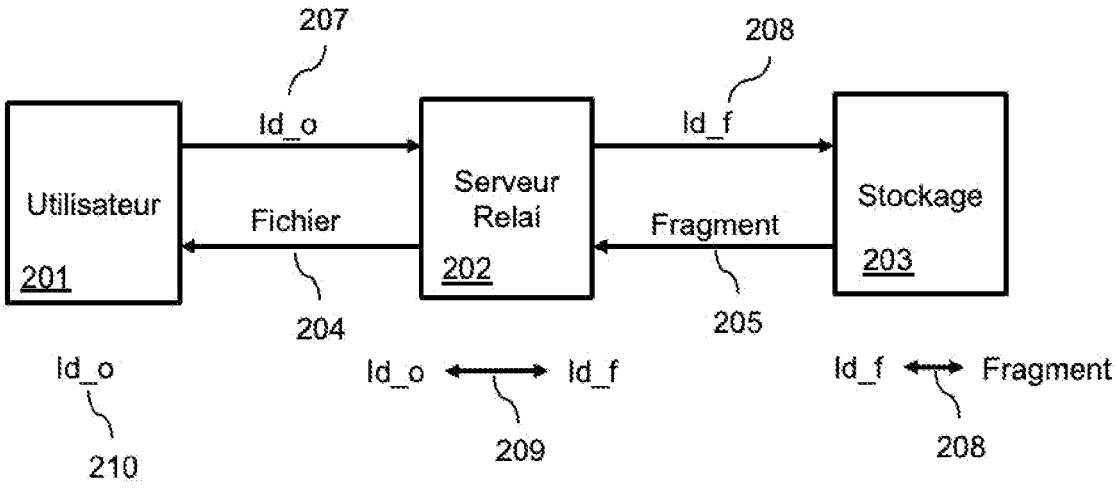


Fig. 2b

[Fig. 3a]

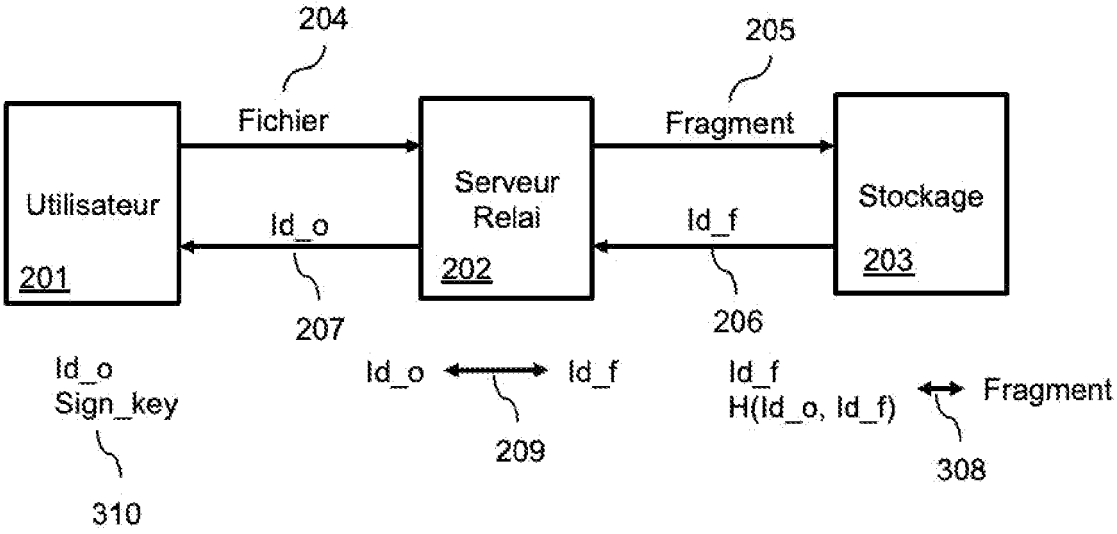


Fig. 3a

[Fig. 3b]

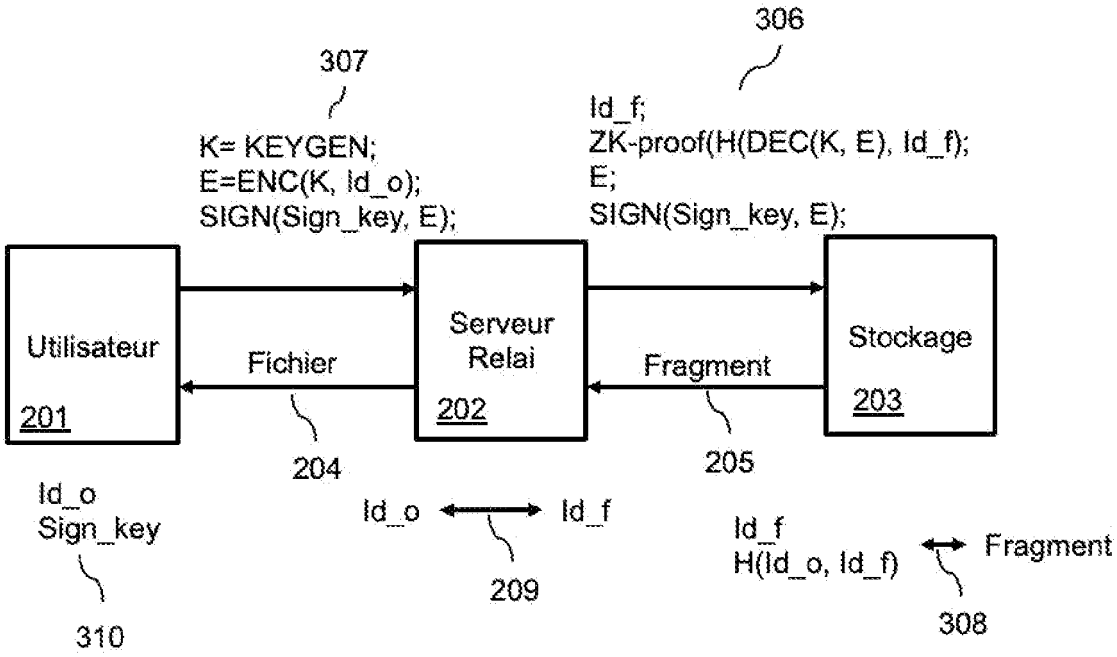
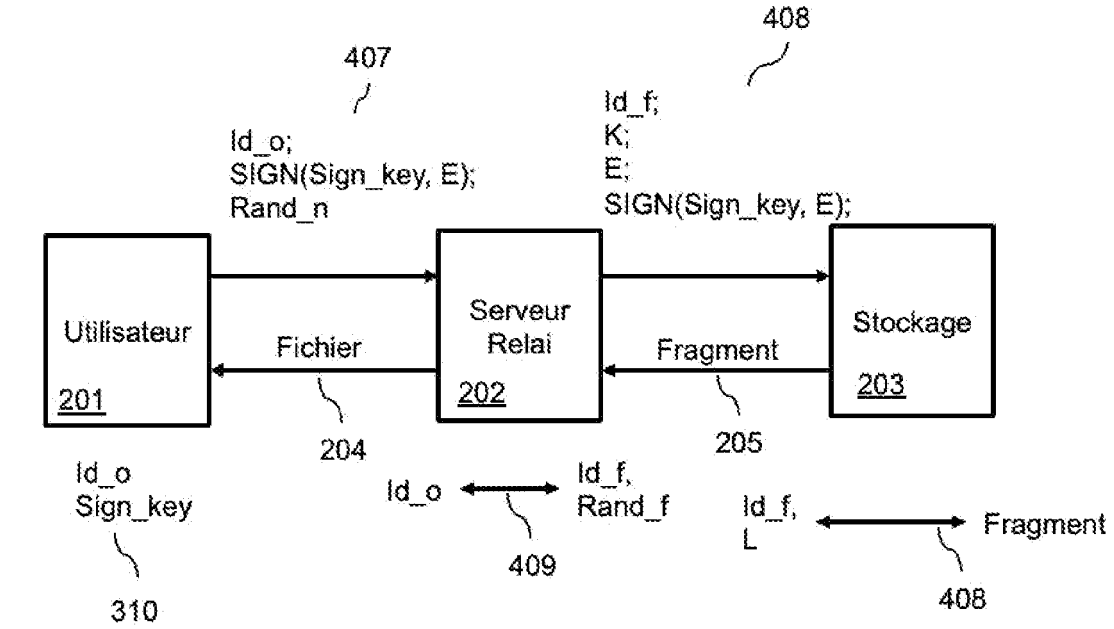

306
307
Id_f;
ZK-proof(H(DEC(K, E), Id_f);
E;
SIGN(Sign_key, E);
K= KEYGEN;
E=ENC(K, Id_o);
SIGN(Sign_key, E);
Utilisateur
201
Fichier
204
Serveur
Relai
202
Fragment
205
Stockage
203
Id_o
Sign_key
310
Id_o
Id_f
209
Id_f
H(Id_o, Id_f)
Fragment
308

Fig. 3b

[Fig. 4]


408
407
Id_f;
K;
E;
SIGN(Sign_key, E);
Id_o;
SIGN(Sign_key, E);
Rand_n
Utilisateur
201
Fichier
Serveur
Relai
202
Fragment
Stockage
203
204
205
Id_o
Sign_key
310
Id_o
Id_f,
Rand_f
409
Id_f,
L
Fragment
408

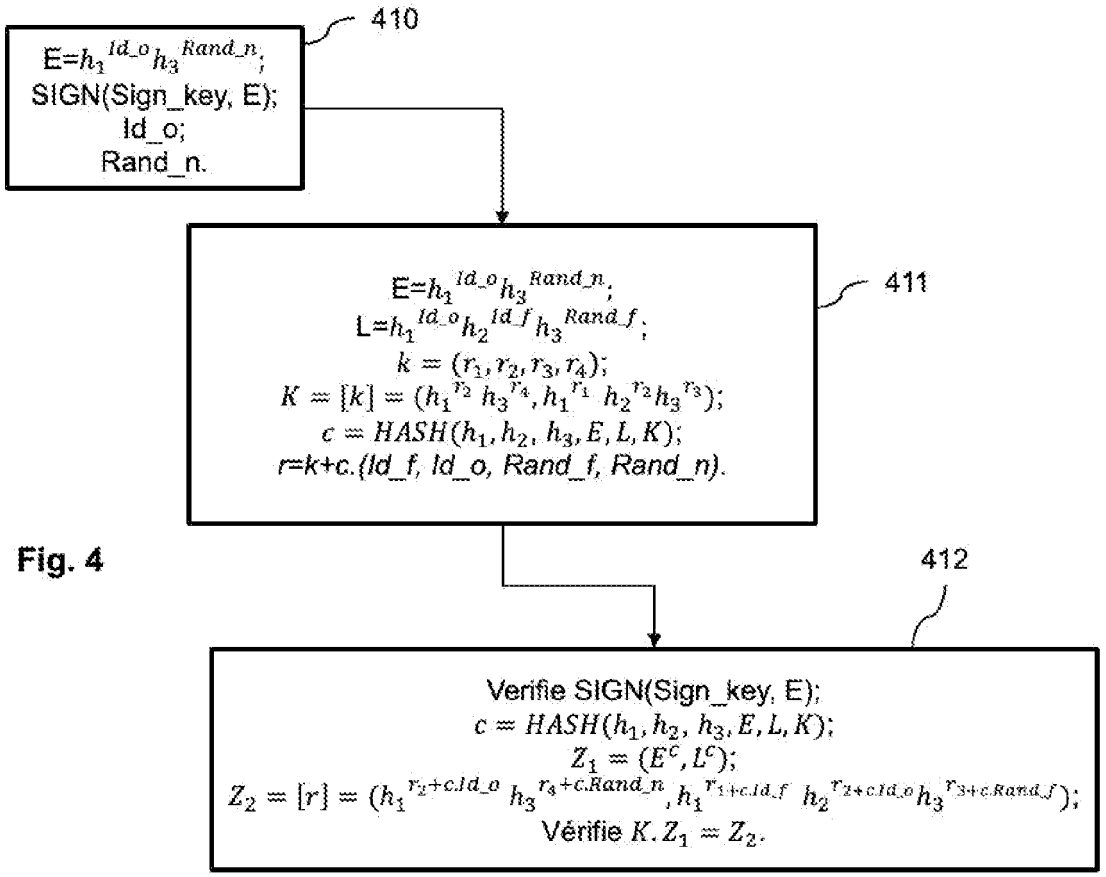

410
E=$h_1^{Id\_o} h_3^{Rand\_n}$;
SIGN(Sign_key, E);
Id_o;
Rand_n.
411
E=$h_1^{Id\_o} h_3^{Rand\_n}$;
L=$h_1^{Id\_o} h_2^{Id\_f} h_3^{Rand\_f}$;
$k = (r_1, r_2, r_3, r_4)$;
$K = [k] = (h_1^{r_2} h_3^{r_4}, h_1^{r_1} h_2^{r_2} h_3^{r_3})$;
$c = HASH(h_1, h_2, h_3, E, L, K)$;
r=k+c.(Id_f, Id_o, Rand_f, Rand_n).
412
Verifie SIGN(Sign_key, E);
$c = HASH(h_1, h_2, h_3, E, L, K)$;
$Z_1 = (E^c, L^c)$;
$Z_2 = [r] = (h_1^{r_2+c.Id\_o} h_3^{r_4+c.Rand\_n}, h_1^{r_1+c.Id\_f} h_2^{r_2+c.Id\_o} h_3^{r_3+c.Rand\_f})$;
Vérifie $K.Z_1 = Z_2$.

Fig. 4

[Fig. 5]

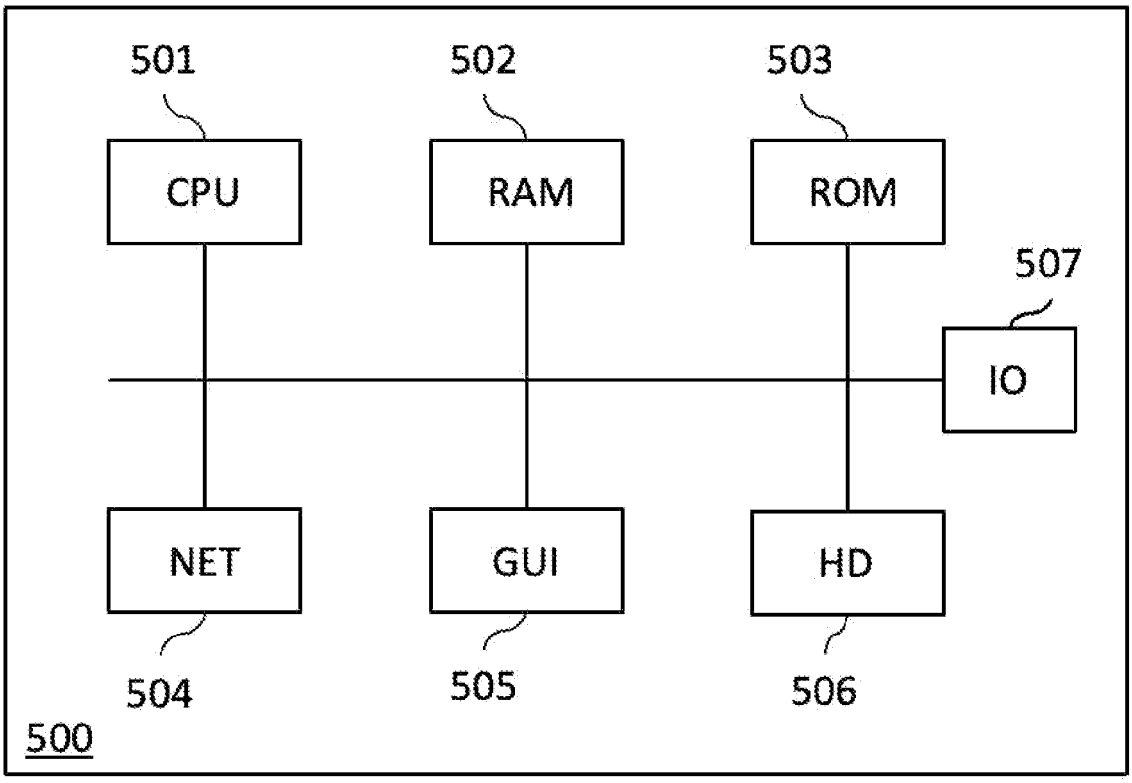


Fig. 5

RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

N° d'enregistrement national

FA 913885
FR 2211306

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
A	CN 110 417 750 B (BEIJING JIANWANG FUTURE TECH CO LTD; WANG YINING; WANG JIAN) 3 juillet 2020 (2020-07-03) * alinéa [0085] - alinéa [0103] * * alinéa [0133] * * alinéa [0141] * * alinéa [0153] - alinéa [0169] * -----	1-7	G06F16/10 G06F21/78 G06F21/64 G06F21/31
A	Adil H: "Off-Chain Data Storage: Ethereum & IPFS", , 17 octobre 2017 (2017-10-17), pages 1-4, XP093050530, Extrait de l'Internet: URL:https://web.archive.org/web/20210604233919/https://didil.medium.com/off-chain-data-storage-ethereum-ipfs-570e030432cf [extrait le 2023-05-30] * page 2 * -----	1-7	
			DOMAINES TECHNIQUES RECHERCHÉS (IPC)
			G06F

Date d'achèvement de la recherche	Examinateur
30 mai 2023	Meis, Marc

CATÉGORIE DES DOCUMENTS CITÉS

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons
.....
& : membre de la même famille, document correspondant

EPO FORM 1503 12.99 (P04C14)

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 2211306 FA 913885

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.
Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du 30-05-2023
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
CN 110417750 B	03-07-2020	AUCUN	

EPO FORM P0465