(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

COURBEVOIE

(11) **N° de publication :** ***3 138 714***

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

(21) **N° d'enregistrement national :** **22 08140**

(51) Int Cl[8] : ***G 06 F 16/27*** *(2022.01), G 04 F 10/00, G 06 F 21/53*

(12) **BREVET D'INVENTION** **B1**

(54) MÉTHODE DE GÉNÉRATION DE PREUVE DE TEMPS ÉCOULÉ ENTRE ÉVÈNEMENTS DANS UN RÉSEAU DE NŒUDS ASYNCHRONES.

(22) **Date de dépôt :** 05.08.22.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 09.02.24 Bulletin 24/06.

(45) **Date de la mise à disposition du public du brevet d'invention :** 04.10.24 Bulletin 24/40.

(56) **Liste des documents cités dans le rapport de recherche :**

Se reporter à la fin du présent fascicule

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

◯ **Demande(s) d'extension :**

(71) **Demandeur(s) :** *COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES Etablissement Public* — FR.

(72) **Inventeur(s) :** HENNEBERT Christine et PAULIN Dylan.

(73) **Titulaire(s) :** COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES Etablissement Public.

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

FR 3 138 714 - B1

Description

Titre de l'invention : MÉTHODE DE GÉNÉRATION DE PREUVE DE TEMPS ÉCOULÉ ENTRE ÉVÈNEMENTS DANS UN RÉSEAU DE NŒUDS ASYNCHRONES

Domaine technique

[0001] La présente invention concerne de manière générale les méthodes de preuve de temps écoulé entre deux évènements dans un réseau asynchrone. Elle trouve notamment application dans un mécanisme de consensus relatif à un historique de transactions sur un registre distribué (*ledger*).

ÉTAT DE LA TECHNIQUE ANTÉRIEURE

[0002] De nombreux mécanismes de consensus ont été proposés pour valider des transactions avant de les incorporer dans un registre distribué (*ledger*), les plus connus étant la preuve de travail ou PoW (*Proof of Work*) et la preuve d'enjeu ou PoS (*Proof of Stake*). Ces mécanismes servent généralement plusieurs objectifs : celui de vérifier l'intégrité des transactions, c'est-à-dire que celles-ci ne sont ni falsifiées ni corrompues, celui d'authentifier les entités ayant émis ces transactions, et enfin celui de garantir la correction de l'historique de ces transactions. En effet, la modification de l'historique de différentes transactions peut conduire à des situations très dommageables : double dépense, débit d'un compte en cryptomonnaie en absence de crédit suffisant, violation d'une condition d'exécution d'un contrat intelligent. La preuve de travail permet de garantir de manière vérifiable (mais au prix d'une certaine latence) un historique correct des transactions mais requiert une consommation considérable d'énergie. Qui plus est, la preuve de travail comme la preuve d'enjeu suppose de mettre en place un système de récompense des mineurs qui n'est pas approprié à certains contextes d'application.

[0003] La preuve de temps écoulé et/ou la vérification de temps écoulé est un sujet qui suscite beaucoup d'intérêt dans la mesure où il est au cœur de la sécurité et de la confiance dans la technologie des chaines de blocs. Les recherches actuelles portent sur les fonctions dites fonctions de durée vérifiables (ou de manière équivalente des fonctions de délai ou d'intervalle de temps vérifiables) ou VFD (*Verifiable Delay Functions*). Ces fonctions font appel à un triplet d'algorithmes : un algorithme d'initialisation (*Setup*) d'un couple de paramètres d'évaluation (e_k) et de vérification (v_k) en fonction d'un niveau de sécurité (λ) et d'une durée donnée (T), un algorithme d'évaluation d'une fonction (*Eval*) à partir d'une valeur d'entrée (x) et du paramètre d'évaluation (e_k), la fonction d'évaluation donnant une valeur de sortie (y) et une preuve (π) et enfin un algorithme de vérification (*Verify*) fournissant un résultat

booléen à partir du paramètre de vérification (v_k), des valeurs d'entrée et de sortie (x,y) et de la preuve (π). Le principe d'une telle fonction de vérification de durée est qu'il n'est pas possible d'anticiper la valeur des paramètres e_k et v_k avant l'expiration de la durée T, et que toute parallélisation de calcul pour exécuter l'algorithme d'initialisation plus rapidement conduit à un résultat d'évaluation non distinguable d'un nombre aléatoire.

[0004] On pourra trouver une présentation des fonctions de durée vérifiables dans l'article de D. Boneh *et al.* intitulé « Verifiable delay functions », publié dans Advances in Cryptology, , CRYPTO 2018, Springer Int'l Publishing, pp. 757-788. Les fonctions de durée vérifiables peuvent être implémentées de différentes manières.

[0005] Une première approche consiste à construire une telle fonction au moyen d'une enclave sécurisée ou TEE (*Trusted Execution Environment*). Plus précisément, une telle enclave exécute l'algorithme d'évaluation précité. Elle reçoit la valeur d'entrée x ainsi qu'un couple de paramètres d'évaluation et de vérification (e_k,v_k), et fournit en sortie, d'une part, une valeur hachée y de x, au moyen d'une fonction de hachage dont la complexité dépend du paramètre de difficulté λ , et, d'autre part, une preuve π obtenue comme signature de y au moyen de la clé privée d'un cryptosystème asymétrique stocké dans l'enclave. L'enclave est synchronisée sur une horloge externe pour effectuer l'évaluation de la durée T. Toutefois, cette horloge est un point d'attaque singulier (*single point of failure*) et peut faire l'objet d'agissements malveillants susceptibles de corrompre le résultat de l'évaluation.

[0006] Une seconde approche consiste à effectuer séquentiellement des tirages d'un nombre aléatoire , à effectuer un hachage de la valeur d'entrée x concaténée avec le nombre aléatoire ainsi obtenu, soit $x\|\alpha$ jusqu'à ce qu'une certaine condition sur le résultat de hachage soit vérifiée, ladite condition dépendant du degré de difficulté λ. C'est sur ce principe que se base la preuve de travail utilisée dans les chaines de blocs Bitcoin et Ethereum, très consommatrice en énergie. D'autres fonctions d'évaluation basées sur des SNARK (*Succinct Non-Interactive ARguments of Knowledge*) récursifs (SNARK de SNARK). Celles-ci font appel à des fonctions de hachage récursives utilisant des techniques de chiffrement homomorphe, également très consommatrices en énergie.

[0007] Enfin, une troisième approche pour implémenter la fonction d'évaluation est de faire appel à une construction polynomiale consistant à calculer séquentiellement (λ fois) le carré d'une fonction de hachage dans un corps de Galois. Comme la précédente, cette approche nécessite de faire d'importants calculs non parallélisables et non optimisables, ce qui conduit donc encore à une importante consommation en énergie.

[0008] L'algorithme de preuve de temps écoulé PoET (*Proof of Elapsed Time*) est un mécanisme de consensus de type Nakamoto pour chaines de blocs à accès contrôlé (*permissioned blockchains*) qui nécessite une consommation d'énergie sensiblement

plus faible que dans Bitcoin. La preuve de travail (PoW) est remplacée ici par un système de loterie qui génère un temps d'attente aléatoire pour chaque mineur (ou nœud validateur) de la chaine. Pour ce faire, chaque mineur dispose d'une enclave sécurisée (TEE) qui lui fournit un temps aléatoire certifié qu'il doit attendre avant d'être autorisé à soumettre son prochain bloc au consensus. N'importe quel nœud participant de la chaine doit pouvoir vérifier que le temps d'attente a bien été généré par le tirage aléatoire certifié dans l'enclave sécurisée et que ce temps d'attente a bien été respecté par le mineur.

[0009] L'algorithme PoET a été déployé dans la chaine de blocs Hyperledger Sawtooth avec succès. Toutefois, les enclaves sécurisées (SGX) utilisent un temporisateur externe situé à l'extérieur de la zone d'exécution de confiance (TEE) et sont susceptibles de subir certaines attaques matérielles visant à corrompre les temps d'attente générés.

[0010] Un objet de la présente invention est par conséquent de proposer une méthode de preuve de temps écoulé entre transactions émises par un nœud vers un registre distribué (*ledger*) et, de manière plus générale, entre évènements intervenant dans un nœud de réseau asynchrone, qui ne présente pas les inconvénients précités, notamment qui soit relativement simple, faiblement consommatrice en énergie et robuste aux attaques matérielles.

Exposé de l'invention

[0011] La présente invention est définie par une méthode de preuve de temps écoulé entre évènements successifs intervenant dans un nœud d'un réseau asynchrone, ledit nœud comprenant un système embarqué partitionné en une zone de confiance et une zone non sécurisée, la zone de confiance comprenant un environnement d'exécution sécurisé ou TEE dans lequel l'intégrité et la confidentialité du code et des données sont assurées, et un composant matériel de confiance ou TPM comportant une première horloge, dit horloge de contrôle donnant le temps écoulé à partir de sa dernière initialisation et un compteur monotone indiquant le nombre d'initialisations de l'horloge de contrôle, l'environnement TEE disposant d'une seconde horloge, dite horloge système, et recevant d'une part l'horloge de contrôle du TPM et d'autre part une troisième horloge, dite horloge de précision , d'un périphérique extérieur au système embarqué, l'horloge de contrôle présentant un degré de sécurité supérieur à l'horloge système et à l'horloge de précision, l'horloge de précision présentant une précision et une stabilité supérieures à celles de l'horloge de contrôle et de l'horloge système, ladite méthode étant originale en ce que :

une application résidant dans l'environnement TEE mesure le temps auquel intervient un évènement observé par le nœud, la mesure étant effectuée à l'aide de la première, de la seconde et de la troisième horloges, pour fournir une première, une

seconde et une troisième mesures de temps relatives à l'évènement observé ;
ladite application détermine, à partir des mesures ainsi obtenues pour l'évènement observé et de celles obtenues pour un évènement précédent, une première, une seconde et une troisième mesures de temps écoulé entre ledit évènement observé et l'évènement précédent ;
ladite application vérifie la cohérence des première, seconde et troisième horloges en comparant les mesures de temps écoulé ainsi obtenues et si elles ne diffèrent pas plus d'une marge de tolérance prédéterminée, forme un message de preuve de temps écoulé entre l'évènement observé et l'évènement précédent, le message de temps écoulé comprenant la valeur du compteur monotone, les mesures de temps relatives à l'évènement observé et les dérives des seconde (ε_{TEE}) et troisième horloge (ε_{GPS}) par rapport à la première horloge sur l'intervalle de temps écoulé ;
ladite application signe ledit message avec une clé privée du nœud avant de le transmettre sur le réseau asynchrone.

[0012] Avantageusement, les dérives de l'horloge de contrôle et de l'horloge système par rapport à l'horloge de précision sont intégrées dans le temps par ladite application et que, lorsque l'une d'elles excède une valeur maximale prédéterminée, celle-ci est corrigée.

[0013] La présente invention concerne également une méthode de preuve de temps écoulé entre transactions successives émises par un nœud, dit nœud client, à destination d'un registre distribué, caractérisée en ce qu'elle met en œuvre une méthode de preuve de temps écoulé entre évènements successifs telle que définie ci-dessus, l'évènement observé étant la formation d'une transaction par ledit nœud client.

[0014] Avantageusement, la transaction émise par le nœud client comprend en outre un haché du code de l'application au moment où la transaction est formée.

[0015] Selon un exemple de réalisation, la transaction émise par le nœud client comprend en outre la valeur d'un compteur de transactions, ledit compteur étant incrémenté à chaque transaction émise.

[0016] La transaction émise par le nœud client peut comprend en outre une mesure du temps auquel la transaction précédente a été émise, la mesure ayant été effectuée au moyen de l'horloge de contrôle et étant signée par une clé privée du module TPM.

[0017] Selon une variante, préalablement à l'émission des transactions successives, le nœud client émet une transaction d'enregistrement à destination du registre distribué, ladite transaction d'enregistrement comprenant une valeur initiale du compteur monotone, et indique les valeurs initiales de l'horloge de contrôle, de l'horloge système et de l'horloge de précision.

[0018] La vérification peut être effectuée par un nœud tiers du réseau asynchrone, dit nœud validateur, pour une nouvelle transaction émise par le nœud client, ladite méthode étant

caractérisée en ce que le nœud validateur extrait de la transaction courante la valeur du compte monotone et vérifie dans le registre distribué qu'elle est identique à celle de la transaction précédente, puis extrait les dérives des seconde et troisième horloges et les compare à une marge de tolérance prédéterminée pour vérifier la cohérence des horloges du nœud client.

[0019] Avantageusement, préalablement à la vérification de la cohérence des horloges du nœud client, le nœud vérificateur s'assure à l'aide de la signature de la transaction que celle-ci a bien été émise par le nœud client.

[0020] Le nœud vérificateur extrait ensuite de la transaction du nœud client la valeur de l'horloge de précision et la compare à la valeur de sa propre horloge de précision à la réception de cette transaction, et vérifie que la différence de la seconde avec la première est supérieure à zéro et inférieure un temps de propagation maximal pré-déterminé.

[0021] De préférence, le nœud vérificateur est un nœud présentant un degré de fiabilité supérieur à une valeur prédéterminée, le degré de fiabilité d'un nœud étant une fonction croissante du nombre de transactions validées avec succès et du temps écoulé depuis son enregistrement dans le registre distribué.

[0022] Si la vérification a été effectuée avec succès, le nœud validateur émet une transaction, dite de transaction de validation, comprenant un haché de la transaction validée et le temps auquel elle a été reçue, mesurée par l'horloge de précision du nœud validateur.

[0023] Le nœud validateur peut vérifier préalablement la cohérence de ses première, seconde et troisième horloges avant d'émettre la transaction de validation vers le registre distribué.

[0024] La transaction de validation comprend typiquement les mesures du temps auquel est effectuée la validation au moyen des première, seconde et troisième horloges, la valeur du compteur monotone du nœud validateur ainsi que les dérives des seconde (ε_{TEE}^{val}) et troisième horloge (ε_{GPS}^{val}) du nœud validateur par rapport à la première horloge du nœud validateur sur l'intervalle de temps écoulé depuis la dernière transaction émise par le nœud validateur.

[0025] Avantageusement, la transaction de validation comprend en outre un haché du code exécutant la vérification dans l'environnement TEE du nœud validateur, au moment où la transaction du nœud client est validée.

Brève description des dessins

[0026] D'autres caractéristiques et avantages de l'invention apparaîtront à la lecture d'un mode de réalisation préférentiel de l'invention, fait en référence aux figures jointes parmi lesquelles :

[0027] [Fig.1] représente de manière schématique un réseau de nœuds asynchrones illustrant le contexte d'application de la présente invention ;

[0028] [Fig.2] représente de manière schématique un système embarqué dans un nœud asynchrone permettant de mettre en œuvre la méthode de génération de preuve de temps écoulé entre deux évènements selon un mode de réalisation de la présente invention ;

[0029] [Fig.3] représente une vérification de la cohérence des temps mesurés par différentes horloges dans le système embarqué de la [Fig.2] ;

[0030] [Fig.4] représente un organigramme d'une méthode de génération de preuve de temps écoulé entre évènements successifs au sein d'un nœud, selon un mode général de réalisation de la présente invention ;

[0031] [Fig.5] représente une méthode de vérification du temps écoulé entre transactions successives émises par un nœud dans un registre distribué ;

[0032] [Fig.6A] représente de manière schématique une étape d'enregistrement d'un nœud client dans la mise en œuvre de la méthode de vérification de la [Fig.5] ;

[0033] [Fig.6B] représente de manière schématique une étape de validation de temps écoulé entre deux transactions successives émises par un nœud client.

[0034] **EXPOSÉ DÉTAILLÉ DE MODES DE RÉALISATION PARTICULIERS**

[0035] Nous considérons dans la suite un réseau de nœuds asynchrones, chaque nœud pouvant émettre des messages sur le réseau. Par réseau de nœuds asynchrones, nous entendons que les différents nœuds du réseau ne sont pas synchronisés par une horloge commune.

[0036] Sans perte de généralité, nous supposerons dans la suite que ces messages sont des transactions destinées à être enregistrées dans un registre distribué (*distributed ledger*), pouvant être un graphe de transactions ou une chaine de blocs. La chaine de blocs est formée par la concaténation de transactions ordonnées temporellement et estampillées par les nœuds qui les ont émises.

[0037] Un réseau de nœuds asynchrones, 100, a été représenté de manière schématique en [Fig.1].

[0038] Il comprend au moins un nœud émetteur, 110, capable de générer et émettre une transaction, dite transaction client, fournissant une preuve de temps écoulé, et au moins un nœud validateur, 120, destiner à vérifier l'exactitude de la preuve ainsi fournie, le nœud validateur stockant la transaction dans le registre distribué lorsque la preuve est considérée exacte. Chaque nœud du réseau, 130, peut ensuite lire la transaction dans le registre et vérifier que cette preuve est bien exacte.

[0039] La génération de preuve de temps écoulé dans le nœud émetteur fait appel à un système embarqué comportant une zone de confiance, dont l'architecture est schématiquement représentée en [Fig.2].

[0040] Le système embarqué ou SOM (*System On Module*), 200, peut être réalisé sous la forme d'une carte fille destinée à être enfichée sur la carte mère du nœud en question. Le système embarqué comporte une zone de confiance 210, comprenant elle-même un environnement d'exécution sécurisée, 213, encore dénommé TEE (*Trusted Execution Environment*) et un composant matériel de confiance, 215, encore dénommé TPM (*Trusted Platform Module*). La zone de confiance, 210, est isolée de la zone d'environnement non sécurisé, dit « normal », 220.

[0041] Le code d'exécution et les données qui résident dans l'environnement TEE sont protégées contre des attaques logicielles. Plus précisément, la confidentialité et l'intégrité du code d'exécution et des données y sont garanties. Le TEE dispose d'un processeur et d'une zone mémoire permettant à des applications qui y résident de pouvoir être exécutées sans risque de corruption. Un exemple de TEE est SGX™ (*Software Guard eXtensions*) de la société Intel.

[0042] Le module TPM, 215, est un équipement passif dépourvu de système d'exploitation mais pouvant contenir des éléments matériels sécurisés tels qu'un coffre-fort pour stocker des clés cryptographiques, un générateur de nombre aléatoire, ou, comme dans le cas présent, un compteur jouant le rôle de chronomètre (*timer*) sécurisé. Plusieurs horloges sont disponibles dans le système embarqué :

- une première horloge, , dite horloge de contrôle, 217, présentant le plus haut niveau de sécurité, est fournie par le module TPM. Elle donne le temps écoulé à partir de sa dernière initialisation. Cette horloge peut être réinitialisée au moyen d'une commande d'initialisation du TEE. A chaque fois que l'horloge de contrôle est initialisée, un compteur monotone permanent, dit compteur permanent de RAZ, est incrémenté. Il est important de noter que ce compteur permanent de RAZ ne peut jamais être réinitialisé y compris par le TEE. La dérive de l'horloge de contrôle par rapport à une horloge de référence peut en outre être corrigée par une commande d'ajustement du TEE ;
- une seconde horloge, , est fournie par un compteur permanent (*persistent TA timer*), 221, du système d'exploitation de l'environnement TEE. La mise à jour de cette seconde horloge ne peut se faire qu'en accédant au niveau de privilège le plus élevé du système d'exploitation ;
- une troisième horloge est constituée par une horloge externe de précision supérieure à celle des deux premières, Clk_GPS, dite horloge de référence, par exemple un temps GPS fourni par un récepteur GPS, 230, à partir des messages satellitaires de navigation qu'il reçoit, est transmis à l'environnement TEE du système embarqué. Il convient cependant de noter que cette horloge de référence peut être corrompue, soit parce que des messages de navigation factices (GPS spoofing) sont reçus par le récepteur et conduisent à un temps erroné, soit parce que l'horloge externe fait l'objet d'une attaque lors de son acheminement entre le récepteur GPS et l'environnement

TEE ;

- une quatrième horloge optionnelle peut être fournie par un module 221 du système d'exploitation de l'environnement non sécurisé, 220, du système embarqué (par exemple une horloge de système Linux). Cette horloge, Clk_REE, non protégée et modifiable par l'utilisateur, est transmise à l'environnement TEE.

[0043] Une application de confiance (*trusted application*) destinée à être exécutée au sein de l'environnement TEE vise à vérifier la cohérence des différentes horloges et à concilier les mesures de temps écoulé entre deux événements, à partir de ces mêmes horloges. Plus précisément, comme illustré en Fig. 3, l'application de confiance, 219, notée ci-après ETC_app, s'assure que les intervalles de temps respectivement mesurés par l'horloge de référence, la seconde et le cas échéant la troisième horloge sont bien cohérents avec l'intervalle de temps mesuré par l'horloge de contrôle, autrement dit que :

[0044] [Math.1]

$$|\Delta_{GPS} - \Delta_{TPM}| < \varepsilon_{max}$$

(1-1)

[0045] [Math.2]

$$|\Delta_{TEE} - \Delta_{TPM}| < \varepsilon_{max}$$

(1-2)

[0046] [Math.3]

$$|\Delta_{REE} - \Delta_{TPM}| < \varepsilon_{max}$$

(1-3)

[0047] avec :

[0048] [Math.4]

$$\Delta_{TPM} = time_TPM(tx_{i+1}) - time_TPM(tx_i), \qquad (2\text{-}1)$$

[0049] [Math.5]

$$\Delta_{TEE} = time_TEE(tx_{i+1}) - time_TEE(tx_i)$$

, (2-2)

[0050] [Math.6]

$$\Delta_{GPS} = time_GPS(tx_{i+1}) - time_GPS(tx_i)$$

, (2-3)

[0051] [Math.7]

$$\Delta_{REE} = time_REE(tx_{i+1}) - time_REE(tx_i)$$

, (2-4)

[0052] et où ε_{max} est une marge de tolérance, tx_i et tx_{i+1} sont deux évènements successifs

observés par le système embarqué, par exemple les instants où sont émises deux transactions successives.

[0053] Lorsque les conditions (1-1), (1-2) et, optionnellement (1-3), sont vérifiées, l'application ETC_app conclut à la cohérence des mesures de temps écoulé entre les évènements successifs et transmet sur le réseau un message de preuve de temps écoulé. Ce message comporte des données relatives à l'évènement tx_{i+1}, le rang de cet évènement, $i+1$, le temps auquel cet évènement est intervenu, mesuré par les différentes horloges du système embarqué, les dérives des horloges extérieures au module TPM par rapport à l'horloge de contrôle, à savoir $\varepsilon_{GPS} = \Delta_{GPS} - \Delta_{TPM}$, $\varepsilon_{TEE} = \Delta_{TEE} - \Delta_{TPM}$ et, le cas échéant $\varepsilon_{REE} = \Delta_{REE} - \Delta_{TPM}$. Il est important de noter que les dérives sont estimées ici par rapport à l'horloge de contrôle, qui est la plus sûre, et non par rapport à l'horloge de référence qui est la plus précise et la plus stable. En revanche, au sein du système embarqué, dès lors que les conditions (1-1) à (1-3) sont remplies, autrement dit, dès lors que les mesures de temps écoulé sont cohérentes entre elles, les horloges du module TPM, de l'environnement TEE et de l'environnement REE sont corrigées de leur dérive par rapport à l'horloge de référence.

[0054] Un exemple de message de preuve de temps écoulé contient typiquement un premier champ de données contenant les données, D, relatives à l'évènement observé (tx_{i+1}) ; un compteur incrémenté à chaque nouvel évènement donnant l'indice ($i+1$) de l'évènement observé ; un premier champ temporel, donnant le temps auquel est intervenu cet évènement, mesuré par l'horloge de contrôle ainsi que le contenu du compteur permanent de RAZ, soit $time_TPM(tx_{i+1}) \parallel CTRAZ$; un second champ temporel résultant de la concaténation du temps mesuré par l'horloge Clk_TEE, et de la dérive $\varepsilon_{TEE} = \Delta_{TEE} - \Delta_{TPM}$ sur l'intervalle séparant l'évènement observé et l'évènement précédent, soit $time_TEE(tx_{i+1}) \parallel \varepsilon_{TEE}$; un troisième champ temporel résultant de la concaténation du temps écoulé, Δ_{GPS}, mesuré par l'horloge de référence et de la dérive $\varepsilon_{GPS} = \Delta_{GPS} - \Delta_{TPM}$ par rapport à l'horloge de contrôle sur le même intervalle, soit $time_GPS(tx_{i+1}) \parallel \varepsilon_{GPS}$; et enfin un quatrième champ temporel optionnel, résultant de la concaténation du temps mesuré par l'horloge Clk_REE, et de la dérive $\varepsilon_{REE} = \Delta_{REE} - \Delta_{TPM}$, sur le même intervalle, soit $time_REE(tx_{i+1}) \parallel \varepsilon_{REE}$.

[0055] Le message peut également avantageusement comporter un champ supplémentaire contenant une valeur hachée du code de l'application de confiance, ETC_app, au moyen d'une fonction de hachage. Le code en question est celui qui est stocké dans l'environnement TEE au moment où l'application ETC_app effectue la vérification de cohérence.

[0056] Il peut enfin comprendre le temps auquel est intervenu l'évènement précédent mesuré

par l'horloge de contrôle, soit $time_TPM(tx_i)$, stockée dans une mémoire de l'environnement TEE. Cette valeur permet de s'assurer du chainage temporel des évènements intervenus au sein du nœud émetteur.

[0057] Le message est ensuite signé au moyen de la clé privée du nœud émetteur et avant d'être transmis sur le réseau.

[0058] Un nœud tiers, dit nœud validateur, recevant un message de preuve de temps écoulé peut tout d'abord vérifier au moyen de la clé publique du nœud que le temps écoulé a bien été mesuré par le nœud émetteur. Il peut s'assurer ensuite, de la cohérence des horloges du nœud émetteur lors de la mesure. Pour ce faire, il s'assure tout d'abord que la valeur du compteur $CTRAZ$ n'a pas changé depuis l'évènement précédent, vérifie que les conditions (1-1), (1-2) et, le cas échéant, (1-3) sont bien respectées en comparant les valeurs absolues $|\varepsilon_{GPS}|$, $|\varepsilon_{TEE}|$ et, le cas échéant, $|\varepsilon_{REE}|$, avec la marge de tolérance, ε_{max}. Si ces valeurs absolues sont inférieures à la marge de tolérance, le nœud validateur conclut à la cohérence des horloges.

[0059] Il peut enfin reconstruire l'historique des évènements au sein du nœud émetteur avec les intervalles de temps exacts entre évènements successifs. Les intervalles de temps sont donnés avec la précision de l'horloge de référence, Clk_GPS, et la fiabilité de l'horloge de contrôle, Clk_TPM, du nœud émetteur.

[0060] La [Fig.4] récapitule sous forme d'organigramme une méthode de génération de preuve de temps écoulé entre évènements successifs au sein d'un nœud, selon un mode général de réalisation de la présente invention.

[0061] On se place dans les mêmes conditions que précédemment, le système embarqué du nœud dispose au moins d'une première horloge, dite horloge de contrôle, au sein d'un module TPM, d'une seconde horloge fournie par le système d'exploitation de l'environnement TEE, dite horloge système, et d'une troisième horloge, externe et non protégée, de précision sensiblement supérieure à la première et la seconde horloges et, pour cette raison dénommée horloge de précision.

[0062] Dans une première étape 410, le système embarqué du nœud mesure le temps auquel intervient l'évènement observé avec les première, seconde et troisième horloges.

[0063] A l'étape 420, une application de confiance dans la zone TEE vérifie la cohérence des horloges à partir des mesures obtenues $time_TPM(tx_{i+1})$, $time_TEE(tx_{i+1})$, et $time_GPS(tx_{i+1})$. A cette fin, il mesure l'intervalle de temps écoulé entre l'évènement observé et l'évènement précédent, par différence avec les mesures précédentes, $time_TPM(tx_i)$, $time_TEE(tx_i)$, et $time_GPS(tx_i)$, également stockées dans la zone TEE, soit Δ_{TPM}, Δ_{TEE}, Δ_{GPS}. Plus précisément, l'application de confiance vérifie que les mesures d'intervalle de temps écoulé ne diffèrent pas plus d'une marge de tolérance prédéterminée, autrement dit que les dérives de la seconde et troisième

horloges par rapport à la première sur l'intervalle séparant deux évènement successifs, sont inférieures à une marge de tolérance.

[0064] Dans l'affirmative, il est conclu à la cohérence des horloges et la méthode se poursuit à l'étape 440.

[0065] Dans la négative, la mesure n'est pas fiable ou une attaque a eu lieu et l'on retourne à l'étape 410 pour l'observation du prochain évènement. Alternativement, ou bien au terme de plusieurs échecs, le système embarqué se déclare en défaut.

[0066] A l'étape 440, si le cumul des dérives dans le temps excède une valeur maximale prédéterminée, les dérives sont corrigées en 445 sur la base de l'horloge de précision.

[0067] A l'étape 450, le système embarqué forme un message de preuve de temps écoulé comme décrit précédemment. En particulier, le message comprend un compteur d'évènements, le compteur permanent de RAZ, le temps auquel est intervenu ledit évènement, tel que mesuré au moyen des différentes horloges, soit $time_TPM(tx_{i+1})$, $time_TPM(tx_{i+1})$, $time_TEE(tx_{i+1})$, et $time_GPS(tx_{i+1})$ et les dérives des seconde et troisième horloges par rapport à l'horloge de contrôle, ε_{TEE}, ε_{GPS}. Le message peut en outre contenir le haché du code de l'application de confiance, ETC_app et/ou le haché du temps auquel est intervenu l'évènement précédent.

[0068] A l'étape 460, le système embarqué signe le message au moyen de la clé secrète du nœud et transmet le message ainsi signé sur le réseau.

[0069] Cette méthode de génération de preuve permet à un nœud tiers du réseau de vérifier que le temps écoulé entre deux évènements successifs du nœud émetteur est bien correct. Elle peut être appliquée par exemple pour prouver les instants auxquels un capteur effectue des mesures dans un réseau IoT ou, comme décrit ci-après dans un mécanisme de consensus pour valider une ou plusieurs transaction(s) à ajouter au registre distribué (*ledger*).

[0070] La [Fig.5] représente une méthode de vérification de temps écoulé entre transactions successives émises par un nœud à destination d'un registre distribué.

[0071] On supposera que la technologie de registre distribué employée, permet de déployer des contrats intelligents, encore dénommés contrats autonomes ou « *smart contracts* ». Un exemple d'une telle technologie est Ethereum. Pour rappel, un contrat intelligent est un programme qui est déployé (c'est-à-dire stocké) dans un registre distribué (*ledger*) et peut être exécuté par n'importe quel nœud du réseau.

[0072] La méthode de vérification décrite ci-après permet à un nœud vérificateur de vérifier le temps écoulé entre deux transactions successives émises par un nœud émetteur, encore dit client, vers un registre distribué. Le nœud émetteur comprend un système embarqué (SOM) tel que décrit précédemment en relation avec la [Fig.2].

[0073] Elle comprend une phase préalable d'enregistrement (*enrolment*), 510, au cours de

laquelle le nœud émetteur transmet une transaction à un premier contrat intelligent, SC_REC. Cette phase d'enregistrement, décrite en détail plus loin, permet au nœud de s'authentifier vis-à-vis des pairs du réseau et de stocker dans la chaine l'heure initiale de chacune des première, seconde et troisième horloges, $time_TPM(tx_0)$, $time_TEE(tx_0)$, $time_GPS(tx_0)$ lues dans l'environnement TEE de son système embarqué.

[0074] Suite à cet enregistrement, le nœud émetteur peut émettre, en 520, des transactions, dites transactions « client », comportant, d'une part, une charge utile à enregistrer dans le registre distribué (ledger) et, d'autre part, une preuve de temps écoulé depuis l'émission de la dernière transaction. Cette transaction peut être considérée comme un cas particulier de message généré selon la méthode de génération de preuve précédemment décrite en relation avec la [Fig.5].

[0075] Cette transaction client comprend par conséquent, le compteur permanent de RAZ, les temps fournis par les différentes horloges internes, Clk_TPM,Clk_TEE, Clk_GPS, le cas échéant, Clk_REE, ainsi que leurs dérives respectives par rapport à l'horloge de contrôle.

[0076] La transaction client peut en outre contenir la valeur hachée du code de l'application de confiance, $-^{app}$, et/ou celle du temps, $time_TPM(tx_i)$, auquel le nœud client a émis la transaction précédente.

[0077] La cohérence des temps fournis par chacune des horloges est vérifiée par le nœud émetteur avant qu'il n'émette une transaction client à destination du registre distribué (ledger).

[0078] Tout nœud tiers du réseau peut ensuite vérifier que le temps écoulé entre deux transactions successives du nœud client est bien correct. Pour ce faire, il vérifie tout d'abord que le compteur permanent de RAZ n'a pas varié, puis vérifie que les mesures de temps écoulé entre la transaction courante et la transaction précédente, au moyen des différentes horloges internes, soit Δ_{TPM}, Δ_{TEE}, Δ_{GPS} ne diffèrent pas plus qu'une marge de tolérance ε_{max}. Alternativement ou cumulativement, il peut extraire les temps $time_TPM(tx_{i+1})$, $time_TEE(tx_{i+1})$, $time_GPS(tx_{i+1})$, (le cas échéant $time_REE(tx_{i+1})$) de la transaction en cours, et lire dans le registre distribué (ledger) les temps $time_TPM(tx_i)$,$time_TEE(tx_i)$, $time_GPS(tx_i)$,($time_REE(tx_i)$, de la transaction précédente et en déduit les mesures de temps écoulé entre les transactions, par les différentes horloges, soit Δ_{TEE},Δ_{GPS},(et , le cas échéant, Δ_{REE}). Les différences entre les mesures de temps écoulés ne doivent pas être supérieures à la marge de tolérance ε_{max}.

[0079] Certains nœuds dits validateurs (jouant le rôle de mineurs) peuvent attester qu'ils ont effectué une telle vérification avec succès et enregistrer cette attestation dans le registre

distribué (ledger) au moyen d'une transaction, dite transaction de validation, transmis au contrat intelligent SC_REC.

[0080] Avant d'émettre une telle transaction en 520, un nœud validateur doit s'être préalablement enregistré selon la procédure d'enregistrement décrite plus haut et disposer d'une cotation de réputation suffisante, comme détaillé plus loin La transaction de validation atteste que le nœud validateur a effectué une vérification de la preuve de temps écoulé entre deux transactions successives d'un nœud client. Avant de procéder à la vérification de la cohérence des mesures de temps écoulé entre transactions clients successives, le nœud validateur s'assure lui-même de la cohérence de ses propres horloges.

[0081] La transaction de validation est spécifique en ce qu'elle comprend dans sa charge utile la valeur hachée de la transaction client qui a été vérifiée au moyen d'une fonction de hachage ainsi que le temps de référence auquel cette transaction client a été vérifiée.

[0082] Elle comprend en outre, selon le même format que la transaction client : un compteur incrémenté à chaque nouvelle transaction émise par le nœud validateur donnant ici l'indice $(j+1)$ de la transaction de validation ; un premier champ temporel, donnant le temps auquel est intervenu la validation, mesuré par l'horloge de contrôle du nœud validateur ainsi que le contenu du compteur permanent de RAZ de ce nœud, soit $time_TPM^{val}(tx_{j+1}) \parallel CTRAZ^{val}$ (l'indice supérieur « val » indique que la valeur est relative au nœud validateur); un second champ temporel résultant de la concaténation du temps écoulé entre les transactions j (non nécessairement une transaction de validation) et $j+1$,tel que mesuré par l'horloge Clk_TEE^{val}, avec la dérive $\varepsilon_{TEE}^{val} = \Delta_{TEE}^{val} - \Delta_{TPM}^{val}$ sur l'intervalle séparant la transaction de validation et la transaction précédente du nœud validateur, soit $time_TEE^{val}(tx_{j+1}) \parallel \varepsilon_{TEE}^{val}$; un troisième champ temporel résultant de la concaténation du temps écoulé, Δ_{GPS}^{val}, tel que mesuré par l'horloge de référence du nœud validateur et de la dérive $\varepsilon_{GPS}^{val} = \Delta_{GPS}^{val} - \Delta_{TPM}^{val}$par rapport à l'horloge de contrôle du nœud validateur, soit $time_GPS^{val}(tx_{j+1}) \parallel \varepsilon_{GPS}^{val}$; et enfin un quatrième champ temporel optionnel, résultant de la concaténation du temps écoulé, mesuré par l'horloge Clk_REE^{val}, et de la dérive $\varepsilon_{REE}^{val} = \Delta_{REE}^{val} - \Delta_{TPM}^{val}$, sur le même intervalle, soit $time_REE^{val}(tx_{j+1}) \parallel \varepsilon_{REE}^{val}$.

[0083] La transaction de validation peut en outre contenir la valeur hachée du code de l'application de confiance, ETC^{val}_app et/ou celle du temps, $time_TPM^{val}(tx_{j+1})$, auquel a été émise la transaction précédente.

[0084] Avant de valider une transaction client, le nœud validateur peut s'assurer que le temps de référence auquel il procède à la validation est supérieur au temps de référence contenu dans la transaction client mais n'excède pas ce denier d'un temps de propagation maximal dans le réseau, autrement dit que :

[0085] [Math.8]

$$0 < \left(time_GPS^{val}(tx_{j+1}) - time_GPS(tx_{i+1}) \right) \leq \tau$$

(3)où est un temps de propagation maximal dans le réseau.

[0086] Un nœud du réseau peut se voir attribuer une cotation de confiance/ réputation en fonction du nombre Nde transactions validées qu'il a émises et du temps écoulé Δ depuis son enregistrement dans la chaine. Plus le nombre N est grand et plus la durée Δ est importante, plus le nœud sera considéré comme fiable. Lorsque la cotation de confiance/réputation d'un nœud est suffisamment élevée, il peut être autorisé à devenir nœud validateur. Le seuil de cotation requis pour être nœud validateur peut croître dans le temps, de manière à contrôler le nombre total de nœuds vérificateurs dans la chaine.

[0087] La [Fig.6A] illustre l'étape d'enregistrement d'un nœud client, 510, dans la méthode de vérification de temps écoulé entre transactions successives de la [Fig.5].

[0088] La transaction d'enregistrement (désignée par $x_enrolment$ sur la figure) du nœud est émise à un instant noté « $registration\ start\ time$» et enregistrée dans le registre distribué (ledger) après avoir été validée. Elle comprend les valeurs initiales des première, seconde et troisième horloges, $time_TPM(tx_0)$, $time_TEE(tx_0)$, $time_GPS(tx_0)$, la valeur du compteur de RAZ (ce compteur peut ne pas avoir une valeur nulle lors de l'enregistrement). Elle peut comprendre en outre le haché de l'application de confiance, ETC_app, ainsi que le temps (mesuré par l'horloge TPM) auquel a été émise la précédente transaction par le nœud client. La transaction d'enregistrement est signée au moyen de la clé privée du nœud en question.

[0089] Cette transaction est transmise au contrat intelligent d'enregistrement, SC_REC. Elle peut ensuite être validée par un nœud validateur, la validation étant ainsi simplifiée car la vérification de cohérence des mesures de temps écoulé depuis la transaction précédente est ici omise.

[0090] Une fois le nœud enregistré dans le registre distribué (ledger), celui-ci peut émettre des transactions client (notées tx_client) au contrat intelligent SC_REC, dès lors qu'il a vérifié la cohérence de ses horloges comme décrit précédemment. Chaque transaction comprend une preuve de temps écoulé entre la transaction émise et la transaction précédente.

[0091] La [Fig.6B] représente de manière schématique une étape de validation de temps écoulé entre deux transactions successives émises par un nœud client. Les transactions successives d'un même nœud sont authentifiées au moyen de leur signature.

[0092] Un nœud validateur peut vérifier la cohérence des mesures de temps écoulé entre transactions successives. Le nœud validateur s'assure tout d'abord que la valeur du compteur de RAZ n'a pas varié entre les deux transactions (pas de réinitialisation de l'horloge de contrôle), lit dans la transaction les valeurs des différentes horloges internes présentes dans la transaction courante, lit dans le registre distribué (ledger) les valeurs des horloges internes présentes dans la transaction précédente enregistrée dans le registre distribué (ledger), et en déduit les mesures de temps écoulé entre ces deux transactions au moyen des différentes horloges, à, savoir Δ_{TPM},Δ_{GPS}, Δ_{TEE}, et optionnellement Δ_{REE}, et s'assure de leur cohérence entre elles à une marge de tolérance près. Si ces mesures sont cohérentes, le nœud validateur transmet au contrat intelligent une transaction de validation, après s'être assuré de la cohérence de ses propres horloges. La transaction client est enregistrée dans le registre distribué (*ledger*) lorsqu'elle est validée par un nombre prédéterminé de nœuds validateurs. L'enregistrement peut être effectué par bloc, après qu'un nombre prédéterminé de transactions client ont été validées.

Revendications

[Revendication 1] Méthode de preuve de temps écoulé entre évènements successifs intervenant dans un nœud d'un réseau asynchrone, ledit nœud comprenant un système embarqué (200) partitionné en une zone de confiance (210) et une zone non sécurisée (220), la zone de confiance comprenant un environnement d'exécution sécurisé ou TEE (213) dans lequel l'intégrité et la confidentialité du code et des données sont assurées, et un composant matériel de confiance ou TPM (215) comportant une première horloge, dit horloge de contrôle donnant le temps écoulé à partir de sa dernière initialisation et un compteur monotone indiquant le nombre d'initialisations de l'horloge de contrôle, l'environnement TEE disposant d'une seconde horloge, dite horloge système, et recevant d'une part l'horloge de contrôle du TPM et d'autre part une troisième horloge, dite horloge de précision , d'un périphérique extérieur au système embarqué, l'horloge de contrôle présentant un degré de sécurité supérieur à l'horloge système et à l'horloge de précision, l'horloge de précision présentant une précision et une stabilité supérieures à celles de l'horloge de contrôle et de l'horloge système, dans laquelle :
une application résidant dans l'environnement TEE mesure (410) le temps auquel intervient un évènement observé par le nœud, la mesure étant effectuée à l'aide de la première, de la seconde et de la troisième horloges, pour fournir une première, une seconde et une troisième mesures de temps relatives à l'évènement observé ;
ladite application détermine (420), à partir des mesures ainsi obtenues pour l'évènement observé et de celles obtenues pour un évènement précédent, une première, une seconde et une troisième mesures de temps écoulé entre ledit évènement observé et l'évènement précédent ;
ladite application vérifie la cohérence (430) des première, seconde et troisième horloges en comparant les mesures de temps écoulé ainsi obtenues et si elles ne diffèrent pas plus d'une marge de tolérance prédéterminée, forme un message de preuve (440) de temps écoulé entre l'évènement observé et l'évènement précédent, le message de temps écoulé comprenant la valeur du compteur monotone, les mesures de temps relatives à l'évènement observé et les dérives des seconde (ε_{TEE}) et troisième horloge (ε_{GPS}) par rapport à la première horloge sur l'intervalle de temps écoulé ;
ladite application signe (460) ledit message avec une clé privée du nœud

avant de le transmettre sur le réseau asynchrone.

[Revendication 2] Méthode de preuve de temps écoulé entre évènements successifs selon la revendication 1, caractérisée en ce que les dérives de l'horloge de contrôle et de l'horloge système par rapport à l'horloge de précision sont intégrées dans le temps par ladite application et que, lorsque l'une d'elles excède une valeur maximale prédéterminée, celle-ci est corrigée.

[Revendication 3] Méthode de preuve de temps écoulé entre transactions successives émises par un nœud, dit nœud client, à destination d'un registre distribué, caractérisée en ce qu'elle met en œuvre une méthode de preuve de temps écoulé entre évènements successifs selon la revendication 1 ou 2, l'évènement observé étant la formation d'une transaction par ledit nœud client.

[Revendication 4] Méthode de preuve de temps écoulé entre transactions successives selon la revendication 3, caractérisée en ce que la transaction émise par le nœud client comprend en outre un haché du code de l'application au moment où la transaction est formée.

[Revendication 5] Méthode de preuve de temps écoulé entre transactions successives selon la revendication 3 ou 4, caractérisée en ce que la transaction émise par le nœud client comprend en outre la valeur d'un compteur de transactions, ledit compteur étant incrémenté à chaque transaction émise.

[Revendication 6] Méthode de preuve de temps écoulé entre transactions successives selon l'une des revendications 3 à 5, caractérisée en ce que la transaction émise par le nœud client comprend en outre une mesure du temps auquel la transaction précédente a été émise, la mesure ayant été effectuée au moyen de l'horloge de contrôle et étant signée par une clé privée du module TPM.

[Revendication 7] Méthode de preuve de temps écoulé entre transactions successives selon l'une des revendications 3 à 6, caractérisée en ce que, préalablement à l'émission des transactions successives, le nœud client émet une transaction d'enregistrement (510) à destination du registre distribué, ladite transaction d'enregistrement comprenant une valeur initiale du compteur monotone, et indique les valeurs initiales de l'horloge de contrôle, de l'horloge système et de l'horloge de précision.

[Revendication 8] Méthode de vérification du temps écoulé entre transactions successives émises par un nœud client mettant en œuvre la méthode de preuve de temps écoulé selon l'une des revendications 3 à 7, la vérification étant effectuée par un nœud tiers du réseau asynchrone, dit nœud validateur, pour une nouvelle transaction émise par le nœud client, ladite méthode

étant caractérisée en ce que le nœud validateur extrait de la transaction courante la valeur du compte monotone et vérifie dans le registre distribué qu'elle est identique à celle de la transaction précédente, puis extrait les dérives des seconde et troisième horloges et les compare à une marge de tolérance prédéterminée pour vérifier la cohérence des horloges du nœud client.

[Revendication 9] Méthode de vérification du temps écoulé entre transactions successives émises par un nœud client selon la revendication 8, caractérisée en ce que, préalablement à la vérification de la cohérence des horloges du nœud client, le nœud vérificateur s'assure à l'aide de la signature de la transaction que celle-ci a bien été émise par le nœud client.

[Revendication 10] Méthode de vérification du temps écoulé entre transactions successives émises par un nœud client selon la revendication 8 ou 9, caractérisée en ce que le nœud vérificateur extrait de la transaction du nœud client la valeur de l'horloge de précision et la compare à la valeur de sa propre horloge de précision à la réception de cette transaction, et vérifie que la différence de la seconde avec la première est supérieure à zéro et inférieure un temps de propagation maximal prédéterminé.

[Revendication 11] Méthode de vérification du temps écoulé entre transactions successives émises par un nœud client selon l'une des revendications 8 à 10, caractérisée en ce que le nœud vérificateur est un nœud présentant un degré de fiabilité supérieur à une valeur prédéterminée, le degré de fiabilité d'un nœud étant une fonction croissante du nombre de transactions validées avec succès et du temps écoulé depuis son enregistrement dans le registre distribué.

[Revendication 12] Méthode de vérification du temps écoulé entre transactions successives émises par un nœud client selon l'une des revendications 8 à 11, caractérisée en ce que, si la vérification a été effectuée avec succès, le nœud validateur émet une transaction, dite de transaction de validation, comprenant un haché de la transaction validée et le temps auquel elle a été reçue, mesurée par l'horloge de précision du nœud validateur.

[Revendication 13] Méthode de vérification du temps écoulé entre transactions successives émises par un nœud client selon la revendication 12, caractérisée en ce que le nœud validateur vérifie préalablement la cohérence de ses première, seconde et troisième horloges avant d'émettre la transaction de validation vers le registre distribué.

[Revendication 14] Méthode de vérification du temps écoulé entre transactions successives émises par un nœud client selon la revendication 13, caractérisée en ce

que la transaction de validation comprend les mesures du temps auquel est effectuée la validation au moyen des première, seconde et troisième horloges, la valeur du compteur monotone du nœud validateur ainsi que les dérives des seconde (ε_{TEE}^{val}) et troisième horloge (ε_{GPS}^{val}) du nœud validateur par rapport à la première horloge du nœud validateur sur l'intervalle de temps écoulé depuis la dernière transaction émise par le nœud validateur.

[Revendication 15] Méthode de vérification du temps écoulé entre transactions successives émises par un nœud client selon la revendication 14, caractérisée en ce que la transaction de validation comprend en outre un haché du code exécutant la vérification dans l'environnement TEE du nœud validateur, au moment où la transaction du nœud client est validée.

[Fig. 1]

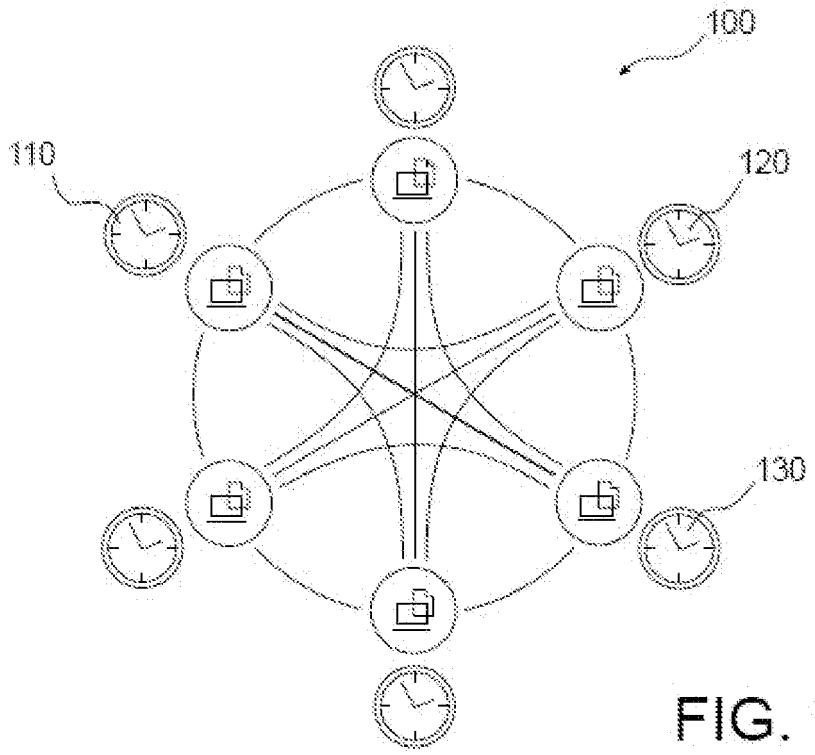


FIG. 1

[Fig. 2]

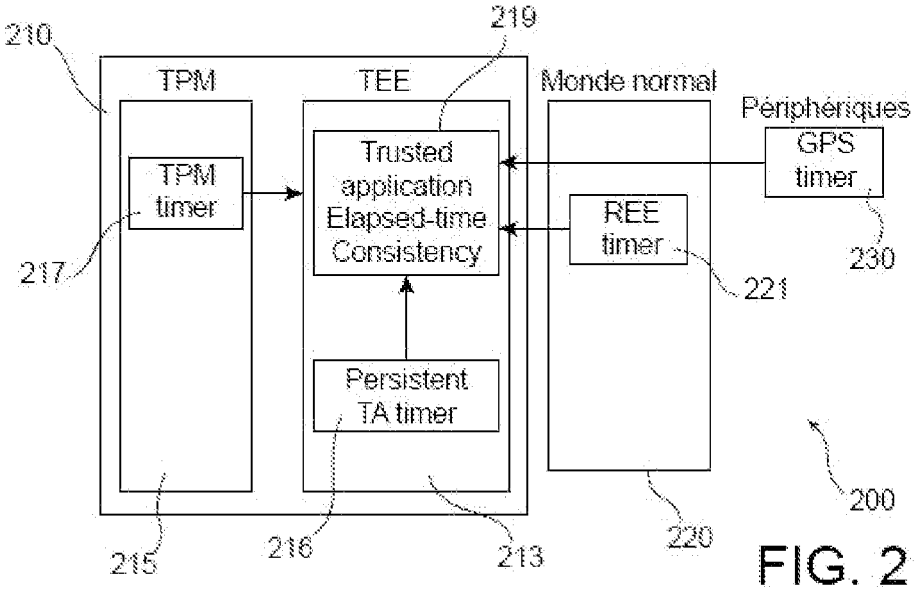


FIG. 2

[Fig. 3]

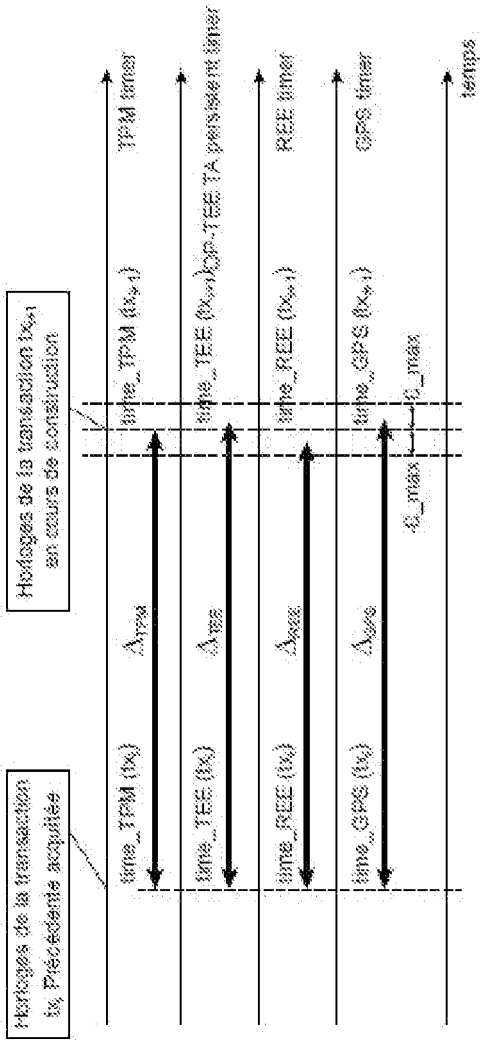

Horloges de la transaction tx précédente acquises
Horloges de la transaction tx en cours de construction
time_TPM (tx)
time_TEE (tx)
time_REE (tx)
time_GPS (tx)
Δtpm
Δtee
Δree
Δgps
time_TPM (tx+1)
time_TEE (tx+1)OP-TEE TA persistent time
time_REE (tx+1)
time_GPS (tx+1)
Δ_max
TPM timer
REE timer
GPS timer
temps

FIG. 3

[Fig. 4]

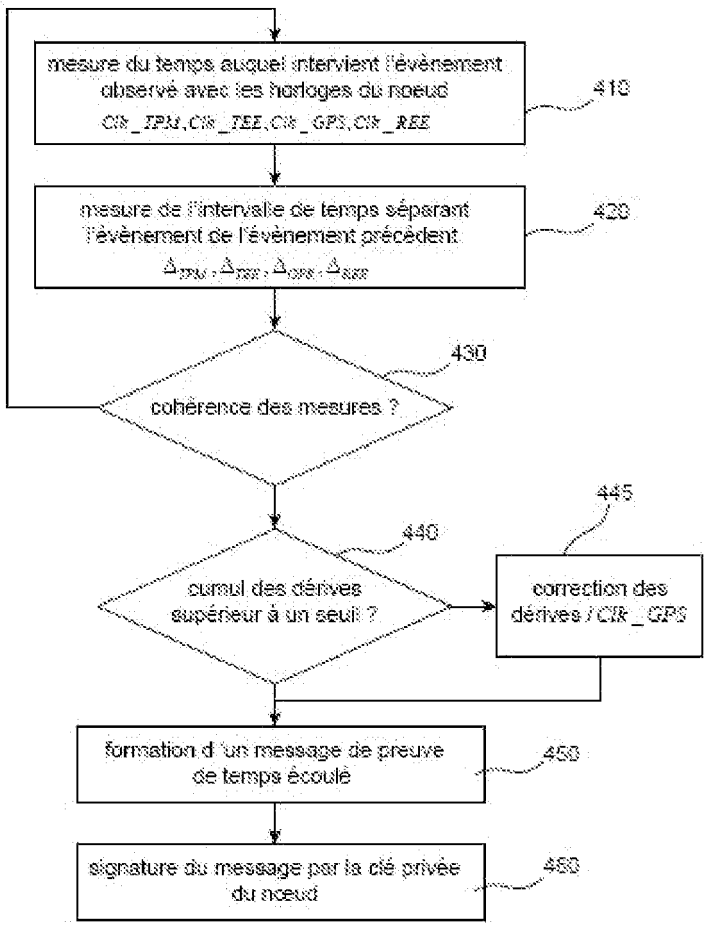

mesure du temps auquel intervient l'évènement observé avec les horloges du noeud
Clk_TPM, Clk_TEE, Clk_GPS, Clk_REE
410
mesure de l'intervalle de temps séparant l'évènement de l'évènement précédent
$\Delta_{TPM}, \Delta_{TEE}, \Delta_{GPS}, \Delta_{REE}$
420
cohérence des mesures ?
430
cumul des dérives supérieur à un seuil ?
440
correction des dérives / Clk_GPS
445
formation d'un message de preuve de temps écoulé
450
signature du message par la clé privée du noeud
460

FIG. 4

[Fig. 5]

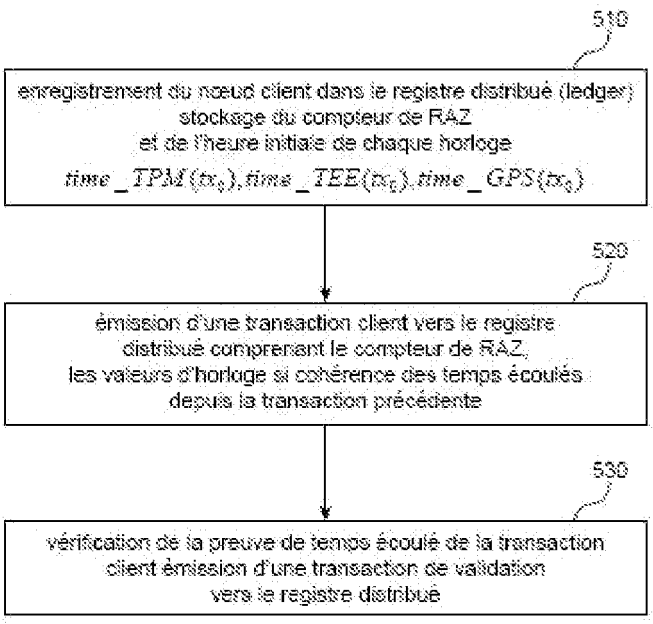


FIG. 5

[Fig. 6A]

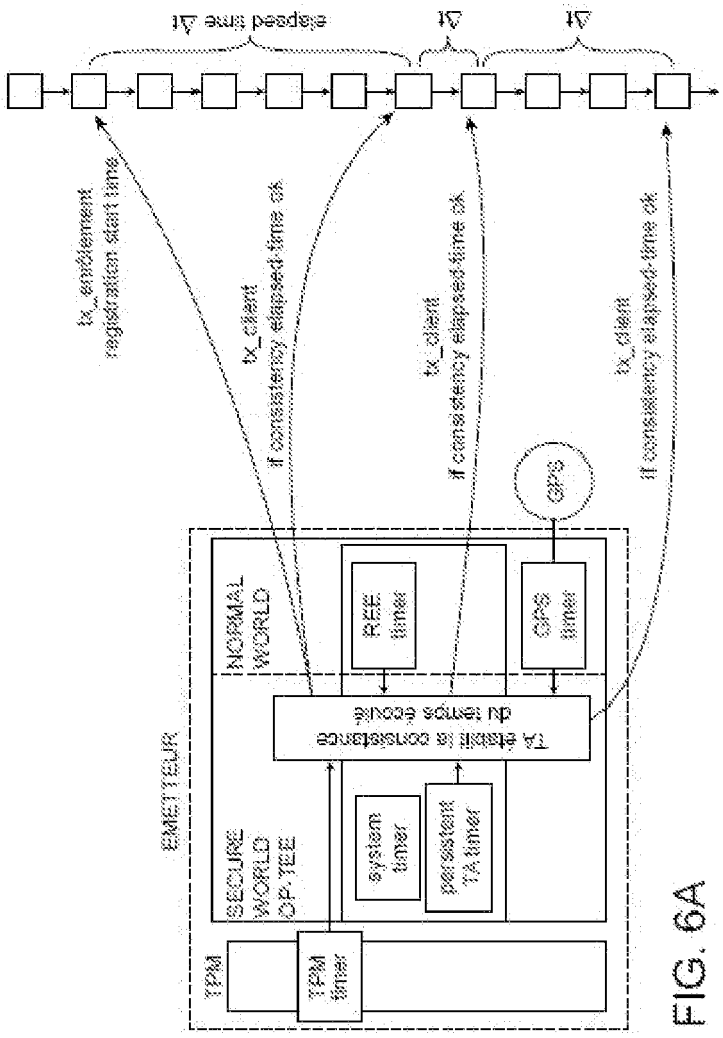


FIG. 6A

[Fig. 6B]

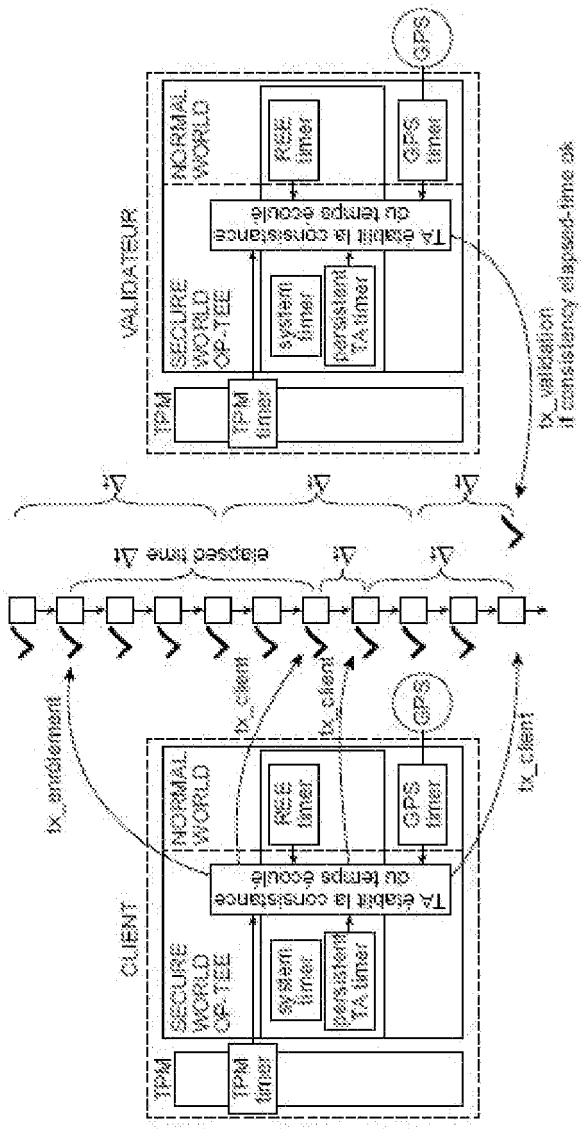

GPS
NORMAL WORLD
REE timer
GPS timer
VALIDATEUR
TA établit la consistance du temps écoulé
SECURE WORLD OP-TEE
system timer
persistent TA timer
TPM
TPM timer
tx_validation
si consistency elapsed-time ok
Δt
elapsed time Δt
tx_enrolement
tx_client
tx_client
tx_client
GPS
NORMAL WORLD
REE timer
GPS timer
CLIENT
TA établit la consistance du temps écoulé
SECURE WORLD OP-TEE
system timer
persistent TA timer
TPM
TPM timer

FIG. 6B

RAPPORT DE RECHERCHE

articles L.612-14, L.612-53 à 69 du code de la propriété intellectuelle

OBJET DU RAPPORT DE RECHERCHE

L'I.N.P.I. annexe à chaque brevet un "RAPPORT DE RECHERCHE" citant les éléments de l'état de la technique qui peuvent être pris en considération pour apprécier la brevetabilité de l'invention, au sens des articles L. 611-11 (nouveauté) et L. 611-14 (activité inventive) du code de la propriété intellectuelle. Ce rapport porte sur les revendications du brevet qui définissent l'objet de l'invention et délimitent l'étendue de la protection.
Après délivrance, l'I.N.P.I. peut, à la requête de toute personne intéressée, formuler un "AVIS DOCUMENTAIRE" sur la base des documents cités dans ce rapport de recherche et de tout autre document que le requérant souhaite voir prendre en considération.

CONDITIONS D'ETABLISSEMENT DU PRESENT RAPPORT DE RECHERCHE

[x] Le demandeur a présenté des observations en réponse au rapport de recherche préliminaire.

[x] Le demandeur a maintenu les revendications.

☐ Le demandeur a modifié les revendications.

☐ Le demandeur a modifié la description pour en éliminer les éléments qui n'étaient plus en concordance avec les nouvelles revendications.

☐ Les tiers ont présenté des observations après publication du rapport de recherche préliminaire.

☐ Un rapport de recherche préliminaire complémentaire a été établi.

DOCUMENTS CITES DANS LE PRESENT RAPPORT DE RECHERCHE

La répartition des documents entre les rubriques 1, 2 et 3 tient compte, le cas échéant, des revendications déposées en dernier lieu et/ou des observations présentées.

☐ Les documents énumérés à la rubrique 1 ci-après sont susceptibles d'être pris en considération pour apprécier la brevetabilité de l'invention.

[x] Les documents énumérés à la rubrique 2 ci-après illustrent l'arrière-plan technologique général.

☐ Les documents énumérés à la rubrique 3 ci-après ont été cités en cours de procédure, mais leur pertinence dépend de la validité des priorités revendiquées.

☐ Aucun document n'a été cité en cours de procédure.

1. ELEMENTS DE L'ETAT DE LA TECHNIQUE SUSCEPTIBLES D'ETRE PRIS EN CONSIDERATION POUR APPRECIER LA BREVETABILITE DE L'INVENTION

NEANT

2. ELEMENTS DE L'ETAT DE LA TECHNIQUE ILLUSTRANT L'ARRIERE-PLAN TECHNOLOGIQUE GENERAL

US 2021/176058 A1 (CHENG QIANG [CN])
10 juin 2021 (2021-06-10)

MIC BOWMAN ET AL: "On Elapsed Time Consensus Protocols",
IACR, INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH
,
vol. 20210127:132313
23 janvier 2021 (2021-01-23), pages 1-47,
XP061051923,
Extrait de l'Internet:
URL:https://eprint.iacr.org/2021/086.pdf
[extrait le 2021-01-23]

US 2021/314143 A1 (CONNER JASON [US])
7 octobre 2021 (2021-10-07)

3. ELEMENTS DE L'ETAT DE LA TECHNIQUE DONT LA PERTINENCE DEPEND DE LA VALIDITE DES PRIORITES

NEANT