(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 943 198**

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

(21) **N° d'enregistrement national :** **09 51646**

(51) Int Cl[8] : ***H 04 L 29/06*** *(2006.01), G 06 K 19/07*

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 16.03.09.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 17.09.10 Bulletin 10/37.

(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 :**

(71) **Demandeur(s) :** *GROUPE DES ECOLES DE TELECOMMUNICATIONS(GET)-ECOLE NATIONALE SUPERIEURE DES TELECOMMUNICATIONS(ENST)* — FR.

(72) **Inventeur(s) :** URIEN PASCAL.

(73) **Titulaire(s) :** GROUPE DES ECOLES DE TELECOMMUNICATIONS(GET)-ECOLE NATIONALE SUPERIEURE DES TELECOMMUNICATIONS(ENST).

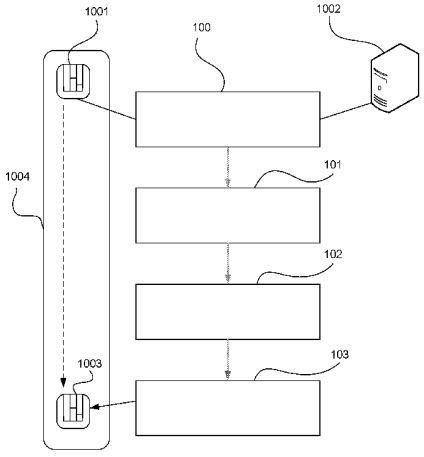
(74) **Mandataire(s) :** CABINET PATRICE VIDON.

(54) PROCEDE DE PRODUCTION DE DONNEES DE SECURISATION, DISPOSITIF ET PROGRAMME D'ORDINATEUR CORRESPONDANT.

(57) L'invention concerne un procédé de production de données de sécurisation, permettant la mise en oeuvre d'une session sécurisée entre une première et au moins une deuxième entité, selon un protocole d'établissement de sessions sécurisées.

Selon l'invention, un tel procédé comprend:

- une étape d'initialisation d'une troisième entité sécurisée liée à ladite première entité;
- une étape de génération d'au moins une partie desdites données de sécurisation au sein de ladite troisième entité;
- une première étape de transmission desdites données de sécurisation générées de ladite troisième entité sécurisée vers ladite première entité;
- une deuxième étape de transmission d'au moins une partie desdites données de sécurisation générées au sein de ladite troisième entité sécurisée à destination d'au moins une quatrième entité sécurisée préalablement initialisée et liée à ladite troisième entité sécurisée.



FR 2 943 198 - A1

Procédé de production de données de sécurisation, dispositif et programme d'ordinateur correspondant.

1 DOMAINE DE L'INVENTION

La présente invention se rapporte au domaine de la gestion des échanges d'informations réalisées entre deux entités d'un réseau de communication.

Plus particulièrement, la présente invention se rapporte à la sécurisation de tels échanges. De nombreuses applications, notamment de commerce ou d'accès à des informations confidentielles, utilisent les protocoles SSL (de l'anglais « Secure Socket Layer » pour « Couche d'encapsulation sécurisée ») ou TLS (de l'anglais « Transport Layer Security » pour « couche de transport sécurisée ») pour échanger des données de manière sécurisée. Bien que des preuves mathématiques existent pour ces protocoles, leur réalisation intégrale par des systèmes informatiques peu surs peut permettre des attaques qui diminuent la confiance que l'on doit légitimement apporter aux procédures d'échange mettant en œuvre ces protocoles.

2 SOLUTIONS DE L'ART ANTERIEUR

La mise en œuvre d'une connexion sécurisée entre deux entités d'un réseau de communication passe par l'initiation d'une session sécurisée basée soit sur le protocole SSL, soit sur le protocole TLS.

Ainsi, pour l'établissement d'une telle session, les deux entités utilisent des mécanismes qui sont censés assurer que la session créée ne pourra pas faire l'objet d'un piratage ou d'un espionnage. Or, les entités en question sont bien souvent vulnérables et non sécurisées de sorte que même si elles produisent des données de sécurisation (par exemple des certificats, des clés de cryptographie ou des secrets partagés) conformément aux protocoles de sécurisation (SSL ou TLS par exemple), rien n'assure que ces entités n'ont pas préalablement fait l'objet d'une attaque et que ces données de sécurisation ne sont pas récupérées directement lors de leurs calculs.

La demande de brevet publiée WO 2008/145558 décrit une méthode de sécurisation des échanges dans laquelle une production de données de sécurisation

est réalisée pour la mise en œuvre d'une session sécurisée entre une première et une deuxième entité, selon un protocole d'établissement de sessions sécurisées tel que SSL ou TLS. Cette méthode permet de résoudre en partie les inconvénients posés par la mise en œuvre des protocoles SSL et TLS par des entités non sécurisées.

Cette méthode comprend une initialisation d'une entité sécurisée tièrce, liée à la première entité, une génération d'une partie des données de sécurisation au sein de la troisième entité et une transmission des données de sécurisation de la troisième entité sécurisée vers ladite première entité. Typiquement, la troisième entité est par exemple une carte à puce de type JavaCard qui réalise une partie des calculs nécessaires à l'établissement de la session sécurisée.

Ainsi, la méthode de WO 2008/145558 permet d'initier un échange de données entre deux entités tout en ayant l'assurance que le matériel cryptographique nécessaire à l'établissement de la session aura été conçu de manière sécurisée.

Cependant, dans le cas où plusieurs échanges de fichiers différents sont obligatoires, il est nécessaire de recourir à la méthode de WO 2008/145558 autant de fois qu'il y a de fichiers à échanger.

Or les capacités de traitement et d'entrée/sortie des entités sécurisées, telles que les cartes à puces ou les java cards, sont souvent faibles : il n'est pas réaliste, en termes de performances, d'utiliser une telle entité de sécurisation pour réaliser des traitements cryptographiques intensifs. Ainsi, la méthode décrite dans WO 2008/145558 ne peut pas être employée seule lorsque des performances importantes sont requises et que de nombreuses sessions sécurisées doivent se dérouler en parallèle.

3 RESUME DE L'INVENTION

L'invention ne présente pas ces inconvénients de l'art antérieur. En effet, l'invention concerne un procédé de production de données de sécurisation, permettant la mise en œuvre d'une session sécurisée entre une première et au

moins une deuxième entité, selon un protocole d'établissement de sessions sécurisées.

Selon l'invention, un tel procédé comprend :

- une étape d'initialisation d'une troisième entité sécurisée liée à ladite première entité ;
- une étape de génération d'au moins une partie desdites données de sécurisation au sein de ladite troisième entité ;
- une première étape de transmission desdites données de sécurisation générées de ladite troisième entité sécurisée vers ladite première entité ;
- une deuxième étape de transmission d'au moins une partie desdites données de sécurisation générées au sein de ladite troisième entité sécurisée à destination d'au moins une quatrième entité sécurisée préalablement initialisée et liée à ladite troisième entité sécurisée.

Ainsi, l'invention permet à différentes entités sécurisées, telles que par exemple des puces, des cartes à puce, des « dongles » de disposer de données de sécurisation, telles que des données de chiffrement tout en n'ayant pas le besoin de générer elles mêmes ces données. Ces données sont générées par l'intermédiaire d'une autre entité sécurisée et transmises après leur création pour être réutilisées par la suite.

Selon un mode de réalisation particulier de l'invention, ladite troisième entité, dite entité maître, génère au moins une partie d'un secret partagé entre ladite première et ladite deuxième entité.

Ainsi, le secret est partagé également avec toutes les entités sécurisées. Elles peuvent donc le réutiliser par la suite pour, par exemple, entamer une nouvelle session sécurisée si le besoin s'en fait sentir.

Plus particulièrement, ladite au moins une partie desdites données de sécurisation générées transmise à ladite au moins une quatrième entité, dite entité esclave, comprend ledit secret partagé sous une forme chiffrée et au moins un identifiant de session de communication sécurisée.

Ainsi, le fait de transmettre les données de sécurisation sous une forme chiffrée au module esclave permet de se prémunir d'éventuels vols ou tentatives de vol de ces données de sécurisation.

Selon un mode de réalisation particulier de l'invention, ledit protocole d'établissement de sessions sécurisées est le protocole SSL.

Selon un mode de réalisation particulier de l'invention, ledit protocole d'établissement de sessions sécurisées est le protocole TLS.

Selon une caractéristique particulière de l'invention, ledit procédé de production comprend en outre :

- une étape de transmission, par ladite première entité, d'au moins un message à destination d'une unité fonctionnelle « RECORD » mise en œuvre au sein de ladite troisième entité ;
- une étape de réception, par ladite première entité, d'au moins un message en provenance de ladite unité fonctionnelle « RECORD » ;
- une étape de calcul d'un ensemble de clés par ladite troisième entité ;
- une étape de collecte dudit ensemble de clés disponibles par ladite première entité auprès de ladite troisième entité.

Ainsi, l'invention est à même de générer des secrets partagés par plusieurs entités sécurisées, comme par exemple plusieurs cartes à puce, car l'ensemble de clés est calculé par la troisième entité.

Plus particulièrement, ladite deuxième étape de transmission est mise en œuvre par un gestionnaire de modules de sécurités qui obtient lesdites données de sécurisation à partir de ladite troisième entité.

Selon une caractéristique particulière de l'invention, ladite deuxième étape de transmission est mise en œuvre lors d'une phase de reprise de ladite session sécurisée.

Ainsi, l'invention permet de gérer, de manière centralisée, le partage des clés entre les entités sécurisées et augmente ainsi le niveau de sécurité de l'ensemble du système.

Selon un autre aspect, l'invention concerne également un procédé

d'établissement d'une session de communication sécurisée entre une première et au moins une deuxième entité, selon un protocole d'établissement de sessions sécurisées. Selon l'invention, un tel procédé comprend :

- une étape d'obtention d'un identifiant de session et d'un secret éphémère calculé lors d'une précédente session de communication sécurisée par une troisième entité sécurisée liée à ladite première entité ;
- une étape de transmission dudit identifiant de session et dudit secret éphémère à une quatrième entité sécurisée, préalablement initialisée et liée à ladite troisième entité sécurisée ;
- une étape d'établissement de ladite session de communication sécurisée en utilisant ladite quatrième entité sécurisée.

Ainsi, l'invention permet d'utiliser d'autres entités sécurisées, telles que des cartes à puce ou des javacard pour l'établissement de sessions de communication sécurisées qui on été précédemment initialisées par une autre entité sécurisée. En conséquence, l'invention permet le traitement en parallèle de plusieurs transactions, comme par exemple des téléchargements de fichiers, en utilisant les services de plusieurs entités sécurisées, tout en minimisant le temps nécessaire à l'établissement de session, tout en offrant un excellent niveau de sécurisation.

L'invention concerne également un dispositif de production de données de sécurisation, permettant la mise en œuvre d'une session sécurisée entre une première et au moins une deuxième entité, selon un protocole d'établissement de sessions sécurisées. Selon l'invention, un tel dispositif comprend :

- des moyens d'initialisation d'une troisième entité sécurisée, attachée à ladite première entité ;
- des moyens de génération d'au moins une partie desdites données de sécurisation ;
- des moyens de transmission desdites données de sécurisation vers ladite première entité ;
- des moyens de transmission d'au moins une partie desdites données de

sécurisation générées au sein de ladite troisième entité sécurisée à destination d'au moins une quatrième entité sécurisée préalablement initialisée et liée à ladite troisième entité sécurisée.

Selon un mode de réalisation particulier de l'invention, lesdits moyens de génération et lesdits moyens de transmission sont regroupés dans une carte à puce.

Selon un mode de réalisation particulier, l'invention concerne également un dispositif portable, tel qu'un jeton USB, comprenant un moyen de stockage d'un gestionnaire de modules de sécurité et au moins deux lecteurs de cartes au format SIM et un dispositif de production de donnée de sécurisation tel que décrit précédemment.

Selon un autre aspect, l'invention concerne également un produit programme d'ordinateur téléchargeable depuis un réseau de communication et/ou stocké sur un support lisible par ordinateur et/ou exécutable par un microprocesseur, et comprenant des instructions de code de programme pour l'exécution du procédé de production tel que décrit précédemment.

Selon un autre aspect, l'invention concerne également un produit programme d'ordinateur téléchargeable depuis un réseau de communication et/ou stocké sur un support lisible par ordinateur et/ou exécutable par un microprocesseur, et comprenant des instructions de code de programme pour l'exécution du procédé d'établissement de session tel que décrit précédemment.

4 LISTE DES FIGURES

D'autres caractéristiques et avantages de l'invention apparaîtront plus clairement à la lecture de la description suivante d'un mode de réalisation préférentiel, donné à titre de simple exemple illustratif et non limitatif, et des dessins annexés, parmi lesquels :

- la figure 1 présente un synoptique du procédé de production de données sécurisées selon l'invention ;
- la figure 2 illustre un exemple de mise en œuvre du procédé de sécurisation à l'aide d'une grille de modules de sécurité selon l'invention ;
- La figure 3 présente l'architecture logique d'une grille de modules de

sécurité selon l'invention ;

- la figure 4 décrit une architecture d'un dispositif de production de données de sécurisation, également appelé module de sécurité.

5 DESCRIPTION DETAILLEE DE L'INVENTION

5.1 Rappel du principe de l'invention

Le principe général de l'invention repose sur une mise en œuvre conjointe d'un ensemble comprenant plusieurs modules de sécurité, et appelé grille de modules de sécurité. Cette grille de modules de sécurité comprend ainsi plusieurs entités sécurisées qui interviennent dans l'établissement de la session de communication sécurisée. Ainsi, à la différence de la méthode décrite dans le document WO 2008/145558, l'invention permet de résoudre les problèmes de performance inhérents à l'utilisation d'entités sécurisées externes.

De cette façon, l'invention élève le niveau de sécurité lors de l'établissement de la session sécurisée tout en assurant un maintien des performances générales du système d'authentification constitué des entités souhaitant établir une session sécurisée (par exemple un client et un serveur) et de la grille de modules de sécurité (comprenant les troisième et quatrième entités) qui est par exemple liée au client.

De manière générale, la grille de modules de sécurité peut être matérialisée sous la forme d'une ou plusieurs cartes à puce à insérer dans un lecteur de carte spécifique, d'un module de type « dongle », à insérer par exemple dans un emplacement de type « USB » (de l'anglais « Universal Serial Bus ») d'un ordinateur ou toute autre forme permettant une communication entre l'entité qui souhaite établir la session sécurisée et la grille de module de sécurité.

La grille de modules de sécurité peut être dédiée à la mise en œuvre d'un protocole particulier tel que SSL et/ou TLS. Ce n'est cependant pas le seul mode de réalisation possible de l'invention. Il est en effet tout à fait envisageable que la grille de modules de sécurité puisse mettre en œuvre plusieurs protocoles afin de garantir une plus grande interopérabilité.

Il est rappelé qu'un module de sécurité désigne, dans le cadre de l'invention, une puce électronique qualifiée usuellement de *"Tamper Resistant Device"*, littéralement un *"composant qui résiste aux attaques"*, qui est à même de gérer des contre mesures physiques et logiques.

Ce module de sécurité comprend notamment une pile logicielle SSL/TLS comportant les unités fonctionnelles HANDSHAKE, ALERT, CCS et RECORD qui sont bien connues de l'homme du métier. Ce module de sécurité communique avec une entité utilisatrice (client ou serveur) à l'aide d'une interface fonctionnelle permettant d'échanger des messages protocolaires SSL/TLS et d'obtenir au moins quatre types de paramètres : *« keys_bloc »*, *« cipher_suite »*, *« SessionID »* et la valeur chiffrée du *« master_secret »*.

La valeur chiffrée du *« master-secret »* (*Master_secret**) est obtenue à l'aide d'une clé secrète partagée entre les différents modules de sécurité et une valeur publique *salt*, selon la relation,

Master_secret = F(Key_Module,salt, MasterSecret)*

L'entité utilisatrice du module de sécurité gère une couche de communication et intègre les unités fonctionnelles ALERT et RECORD et de manière optionnelle les unités fonctionnelles HANDSHAKE et CCS. Les informations issues de la couche APPLICATION sont sécurisées par la couche RECORD.

L'invention propose de mettre en œuvre conjointement l'entité utilisatrice et la grille de modules de sécurités pour établir une session sécurisée avec un serveur. Astucieusement, une partie des étapes nécessaires à l'établissement de la session est réalisée par l'intermédiaire de la grille de module de sécurité tandis que l'autre partie est réalisée par l'entité utilisatrice. Il se peut, dans un mode de réalisation particulier de l'invention, que les étapes mises en œuvre par l'entité utilisatrice et par la grille de module de sécurité diffèrent à chaque création d'une nouvelle session sécurisée. De cette manière, il est plus difficile de prévoir le fonctionnement général du système et de tenter de forcer le mécanisme de sécurisation offert par l'invention.

Dans un mode de réalisation particulier de l'invention, au sein de la grille de modules de sécurité, on distingue deux types de modules de sécurité différents : les modules maîtres et les modules esclaves. Un module de sécurité esclave dépend d'un module de sécurité maître sans lequel il ne peut pas fonctionner. Plus particulièrement, selon l'invention, un module maître est le seul capable de calculer un secret particulier éphémère (le « mastersecret », ce dernier terme étant utilisé par la suite).

On rappelle que le « mastersecret », dans le cadre par exemple de la mise en œuvre du protocole TLS, est calculé lors de la phase dite « full mode ». Lors de cette phase, les échanges entre le client et le serveur permettent de calculer un secret commun, partagé entre le client et le serveur et qui sert de base à la création de toutes les autres données de chiffrement nécessaire à la session sécurisée.

Dans une grille de modules de sécurité selon l'invention seul un module maître peut participer, avec l'entité utilisatrice, au calcul de ce « mastersecret ».

Par la suite, afin de permettre un mécanisme de reprise de session plus rapide, par exemple dans le cadre d'une phase dite de « Resumed mode», un module esclave peut utiliser le « mastersecret » calculé par son module maître pour poursuivre une session sécurisée ou pour réaliser d'autres opérations lors de la session sécurisée.

Selon l'invention, un module esclave entre en possession du « mastersecret » par l'intermédiaire du module maître auquel il est associé. Pour ce faire, le module maître distribue, selon l'invention, ce « mastersecret » aux modules esclaves, mais de manière sécurisée.

Cela signifie que, selon l'invention, la distribution du « mastersecret » est réalisée selon un protocole particulier régissant les échanges de données entre le module maître et le module esclave auquel il est associé. Un tel protocole peut prendre la forme de commandes qui sont transmises à ces modules pour permettre l'échange.

Toujours selon l'invention, d'un point de vue logique l'identité de l'entité utilisatrice est liée uniquement au module maître. Cela signifie que dans le

processus d'établissement de la session sécurisée, l'entité utilisatrice n'a pas connaissance de la présence des modules esclaves.

On présente, en relation avec la figure 1, le procédé de production de données sécurisées selon l'invention. Il comprend :

- une étape d'initialisation (100) d'un module de sécurité 1001 (par exemple une carte à puce), attachée à une première entité 1002 (par exemple un ordinateur personnel) ;
- une étape de génération (101) d'une partie des données de sécurisation au sein du module de sécurité ;
- une étape de transmission (102) des données de sécurisation du module de sécurité vers la première entité.
- une étape de transmission (103) d'au moins une partie desdites données de sécurisation générées au sein du module de sécurité 1001 à destination d'un deuxième module de sécurité 1003 préalablement initialisée et lié au premier module de sécurité 1001, formant ainsi une grille de modules de sécurité 1004 dans laquelle au moins certaines données de sécurisation sont partagées.

En d'autres termes, l'invention propose une grille de modules de sécurité, servant à établir des sessions sécurisées de transmission de données et qui partagent des données pour établir ces sessions sécurisées de transmission.

Par la suite, on présente un mode de réalisation de l'invention dans lequel un gestionnaire de modules de sécurité, intégrée à la grille de modules de sécurité, gère le fonctionnement général de celle-ci. Il est clair cependant que l'invention ne se limite pas à cette mise en oeuvre particulière.

5.2 Description d'un mode de réalisation

On présente dans ce mode de réalisation, la mise en œuvre d'une grille de modules de sécurité selon l'invention.

On décrit les fonctionnalités originales d'une grille de modules de sécurité qui selon l'invention réalise la phase d'authentification du protocole TLS, puis permet à une application d'utiliser le tunnel sécurisé préalablement établi.

Comme cela a déjà été évoqué, un module de sécurité réalise les fonctions de client ou de serveur TLS, son logiciel embarqué comporte donc les unités fonctionnelles HANDSHAKE, ALERT, CCS et RECORD.

La figure 2 présente le module de sécurité TLS et son utilisateur, c'est-à-dire une application munie d'un sous ensemble de la pile TLS, soit de manière obligatoire les couches RECORD et ALERT et de manière optionnelle les couches CCS et HANDSHAKE. Cette entité utilisatrice peut être un client (par exemple une application cliente de type navigateur web) ou un serveur (par exemple un serveur web gérant les sessions sécurisées).

Un module de sécurité offre une interface fonctionnelle qui comprend neuf commandes, SET-Credentials, Start, Process-TLS, GET-Keys_bloc, Compute-Keys_bloc, GET-Cipher_suite, GET-SessionID, GET-Master_secret, SET-Master-Secret.

De telles commandes peuvent être réalisées en suivant la norme ISO 7816 selon un codage couramment dénommé « APDUs » (de l'anglais « Application Protocol Data Unit » pour « Unité de données (PDU) de la couche Application »).

Le module de sécurité (210) qui met en œuvre le procédé de production selon l'invention comprend les unités fonctionnelles nécessaires à la mise en œuvre du procédé de sécurisation à savoir les couches « RECORD » (2104) et « ALERT » (2102) et de manière optionnelle les couches « CCS » (2103) et « HANDSHAKE » (2101).

L'interface fonctionnelle (220) permet à l'entité utilisatrice (200) de faire appel au module de sécurité (210) pour la production de données de sécurisation.

5.3 Description des commandes

5.3.1 Commande SET-Credentials

Le rôle du module, c'est-à-dire son comportement client ou entité serveur ainsi que les différents paramètres nécessaires à son fonctionnement, qualifiés usuellement de lettres de crédits ou « credentials » en langue anglaise (certificats *X509, clé privée RSA*) est activé par la commande SET-Credentials() :

```
SET-Credentials(Credentials, rôle)
```

5.3.2 Start(Unix-Time)

Dans ce mode de réalisation, une commande « Start » initialise une session TLS; puisque les modules de sécurité ne comportent pas en règle générale d'horloge elle renseigne également l'heure GMT au format dit UNIX, c'est-à-dire un nombre de 32 bits qui mesure le nombre de secondes écoulées depuis le 1ier janvier 1970 :

```
Start(Unix-Time)
```

Une telle commande permet, en quelque sorte, de préparer le module de sécurité à effectuer les calculs nécessaires dans le cadre de l'invention.

5.3.3 Process-TLS

Les paquets TLS, c'est-à-dire les messages produits par une l'unité fonctionnelle RECORD, sont transmis au module de sécurité à l'aide de la commande Process-TLS(Record-Packets) qui retourne un ou plusieurs messages RECORD.

```
Record-Packets = Process-TLS(Record-Packets)
```

5.3.4 GET-Keys_bloc

Lorsque un module de sécurité TLS a conduit avec succès l'authentification de son interlocuteur il calcule le keys_bloc, la couche RECORD bascule en mode chiffré, et délivre les messages CCS et FINISHED. La commande GET-Keys_bloc collecte alors l'ensemble des clés disponibles,

```
keys_bloc = GET-Keys_bloc()
```

L'utilisateur des services du module de sécurité peut alors gérer de manière autonome (sans l'aide du module de sécurité) sa propre couche RECORD. En effet il connaît les clés du canal sécurisé (keys_bloc) et la valeur courante du paramètre seq_num égale à 1 (la valeur 0 a été utilisée pour le calcul d'intégrité HMAC du message FINISHED).

5.3.5 Compute-Keys_bloc

La commande Compute-Keys_bloc() associée aux nombres aléatoires générés par l'entité cliente et l'entité serveur (Client-Random et Server-Random) permet de calculer le paramètre « keys_bloc » . Elle est utile lors d'une session de

type « Session Resumption », ou l'utilisateur du module de sécurité emploie ce dernier, uniquement pour l'obtention du keys_bloc.

```
keys_bloc = Compute-Keys_bloc(Client-Random, Server-Random)
```

Il est important de remarquer que dans ce cas le module de sécurité n'exporte pas la valeur du « master_secret ». Il est donc impossible de conduire une session de type « Session Resumption » en l'absence du module de sécurité, qui garantit donc la bonne foi de l'usager du service.

5.3.6 GET-Cipher_suite

Une commande GET-Cipher_suite permet de connaître les paramètres de sécurité, indexés par le nombre cipher_suite, associé à l'unité fonctionnelle RECORD.

```
cipher_suite = Get-Cipher_suite()
```

5.3.7 GET-SessionID

La commande GET-SessionID retourne le paramètre « SessionID » associé à la session précédente associée à un mastersecret particulier. C'est une information utile pour la grille de modules sécurité qui permet à des modules esclaves de réaliser une phase de « Session Resumption ».

```
SessionID = GET-SessionID()
```

5.3.8 GET-Master_secret

La commande GET-Master_secret() collecte une valeur chiffrée du *master_secret (master_secret*)* ainsi qu'un ensemble de paramètres (*salt*) permettant de réaliser le déchiffrement de cette information.

```
master_secret* | salt = GET-Master_secret()
```

Le master_secret est chiffré à l'aide d'une clé secrète symétrique ou asymétrique *(Key_Module)*, partagée par un ensemble de modules de sécurité, et associée à un algorithme de chiffrement (tel que AES, Triple DES, RSA) et d'un nombre aléatoire *salt* généré par le module de sécurité.

```
Master_secret* = F(Key_Module,salt, MasterSecret)
```

5.3.9 Set-Master_Secret

La commande Set-Master_Secret(Master_Secret* | Salt, SessionID) réalise la mise jour d'un master_secret, associé à un index SessionID dans un module de sécurité de type esclave par exemple.

L'invention concerne ainsi également toute carte à puce ou entité sécurisée de ce type qui comprend les commandes précédentes permettant la lecture, le transfert et l'initialisation d'une session sécurisée à partir d'un secret éphémère (le « mastersecret ») calculé par une autre entité sécurisée.

En d'autres termes, l'invention concerne également une méthode d'établissement d'une session de communication à l'aide d'une entité sécurisée qui récupère le secret éphémère et l'identifiant d'une session qui a été précédemment initialisée par une autre entité sécurisée. Ces deux entités sécurisées sont de préférence liées entre elles de sorte qu'elles sont soit présente au sein d'une même carte à puce, soit qu'elles communique par l'intermédiaire d'un module spécifique qui va gérer des interactions (par exemple l'exécution de certaines des commandes précédemment décrites) entre les entités sécurisées.

Ainsi, les objectifs de sécurisation des données sont atteints, selon l'invention, à l'aide d'un module de gestion, également appelé gestionnaire de modules de sécurité, du type hébergeant et exécutant un logiciel assurant notamment des fonctions de gestion et de mémorisation de données de sécurisation, ledit logiciel comprenant des moyens d'exécution de commandes de récupération, de mémorisation et de transmission, par exemple envoyées au logiciel par au moins un logiciel client et appartenant à un jeu de commandes de récupération, de mémorisation et de transmission prédéterminé (GET-Session_ID, GET-Master_Secret, Set-Master_Secret, etc.).

5.4 Mise en œuvre du protocole

A l'aide des neuf commandes décrites précédemment il est possible de mettre en œuvre une grille de modules de sécurité.

La figure 3 présente l'architecture logique d'une grille de modules de sécurité selon l'invention. Une unité fonctionnelle dite *Gestionnaire de Modules de Sécurité* contrôle une pluralité de modules de sécurité.

Selon l'invention, il existe deux classes de modules de sécurité les modules dit maître et les modules dit esclaves.

Les modules maîtres sont identifiés par des index variant de 1 à p. Les modules esclaves sont identifiés par des index strictement supérieurs à p.

Un module maître stocke un certificat *X509* mais également la clé privée RSA nécessaire à l'authentification du client. Les modules maîtres partagent une clé *KeyModule, utilisée pour les opérations de chiffrement et de déchiffrement du mastersecret.*

Un module esclave partage avec les modules maîtres une clé cryptographique commune *KeyModule*, mais ne stocke pas la clé privée du client.

Le *Gestionnaire de Modules de Sécurité* est associé à au moins un module maître, les configurations à n modules comportent en conséquence p modules maître (p étant supérieur ou égal à 1) et $k=n-p$ modules esclaves (k pouvant être égal à zéro). Par exemple, une configuration de grille comprenant $n=16$ modules, dont $p=4$ modules maîtres, comprendra $k=12$ modules esclaves.

Lors de l'ouverture d'une session TCP, le *Gestionnaire de Module de Sécurité* sélectionne de manière prioritaire un module de sécurité maître. Si cette opération est impossible, c'est-à-dire que tous les modules maîtres sont affectés à des sessions en cours d'ouverture, un module esclave est choisi. Si aucun module n'est libre le *Gestionnaire de Module de Sécurité* se met alors en attente de la disponibilité d'un module.

Selon l'invention, au début de chaque session le *Gestionnaire de Modules de Sécurité* met à jour les paramètres (*SessionID*, *MasterSecret*) utilisées par une précédente session à l'aide, selon l'invention, de la commande *Set-MasterSecret*. Grâce à cette procédure il permet à un module (maître ou esclave) de gérer une session en mode *Resumption*.

Si un module esclave échoue dans une tentative d'ouverture d'une session en mode *Resumption* à l'aide des données transmises par le gestionnaire de modules de sécurité, c'est-à-dire si le serveur impose une session en mode *Full* (par exemple parce que la durée de vie de la session « resume » a expiré), il termine la session courante.

A la fin de chaque ouverture de session (quand la procédure de HANDSHAKE est terminée), le *Gestionnaire de Modules de Sécurité* collecte les paramètres SessionID et MasterSecret) grâce aux commandes Get-SessionID et Get-MasterSecret introduites par l'invention. Ainsi, le Gestionnaire de Modules de Sécurité est à même de fournir, lors d'une prochaine session, les données collectées, aussi bien aux modules maîtres qu'aux modules esclaves.

On présente, en relation avec la figure 3, une vue schématique d'une grille de modules de sécurité selon l'invention. Cette grille de modules de sécurité 300, comprend un composant hébergeant un gestionnaire de module de sécurité (GMS) 301, chargé de mémoriser d'une part et de distribuer d'autre part les données générées par les modules maîtres.

La grille de module de sécurité 300 comprend également des modules maîtres 302 à 305 qui génèrent au moins une partie des données de sécurisation en lien avec l'entité à laquelle la grille de modules de sécurité est connectée. Dans les modes de réalisation de l'invention présentés précédemment, les modules maîtres calculent la valeur du *MasterSecret* pour une session en mode « Full ». La grille comprend également des modules de sécurité esclaves 307 à 318.

Les modules maîtres peuvent être associés à un nombre prédéterminé de modules esclaves (par exemple trois dans l'exemple de la figure 3) formant ainsi un groupe de modules de sécurité 306. Cette préassociation n'est pas obligatoire. Le gestionnaire de modules de sécurité 301 peut dynamiquement, en fonction des besoins, associer les modules de sécurité esclave en fonction du nombre de sessions de sécurisation requises à l'aide d'une unité fonctionnelle comprenant des moyens d'obtention d'un nombre de connexion ou d'un nombre d'éléments à télécharger, s'il s'agit par exemple d'une session de communication « http »

requérant le téléchargement d'images ou d'autres éléments en provenance d'un serveur Web.

Une telle mise en place des sessions sécurisées grâce au procédé et à l'architecture de grille de modules de sécurité de l'invention présente de nombreux avantages.

Si l'on note respectivement *TF* et *TR* les temps nécessaires pour réaliser une session *FULL* et *RESUMPTION* dans un module de sécurité. Pour des raisons théoriques évoquées dans plusieurs publications scientifiques *TR* est inférieur à *TF (TR<TF)*, par exemple *TR* est de l'ordre de la moitié de *TF*. Cette propriété est détaillée par exemple dans l'article intitulé «*The OpenEapSmartcard Platform*», écrit par Pascal Urien et Mesmin Dandjinou, qui est disponible sous la référence «*Network Control and Engineering for QoS, Security and Mobility, IV: Fourth IFIP International Conference on Network Control and Engineering for QoS, Security, and Mobility, Lannion, France, November 14-18, 2005, par Dominique Gaïti, Edition: illustrated, Springer, 2007, ISBN 0387496890, 9780387496894*».

Dans l'état de l'art actuel les serveurs WEB utilisent largement le mode RESUMPTION afin de limiter la charge des calculs asymétriques (RSA, etc). Typiquement un navigateur télécharge, via une requête HTTPS, un premier fichier (une page HTML) en mode FULL, puis il conserve le même *MasterSecret* (et donc autorise le mode RESUMPTION) durant une période de temps prédéfinie, par exemple 10 minutes.

La norme HTTP 1.1 (RFC 2616) recommande l'usage de deux connexions TCP au plus entre un navigateur et un serveur WEB. Cependant des navigateurs commerciaux, tels que Internet Explorer utilisent jusqu'à quatre connexions TCP simultanées.

L'utilisation d'un seul module de sécurité permet le téléchargement d'au plus *1/TF* fichiers par seconde en mode FULL et d'au plus *1/TR* fichier par seconde en mode RESUMPTION.

En l'absence de procédure assurant le transfert du « *MasterSecret* » entre modules de sécurité, tel que le propose l'invention, la mise en œuvre de *N*

modules de sécurité ne permet pas de dépasser la limite de *N/TF* fichiers par seconde.

En effet, comme les modules de sécurité ne partagent pas de données, ils sont obligés d'initialiser, de manière indépendante, les sessions sécurisées. Or, une telle initialisation doit être réalisée en mode FULL, et non pas en mode RESUMPTION. Donc, le nombre maximum de fichiers transmis par seconde ne peut pas dépasser la limite *N/TF*.

Une des caractéristiques avantageuse de l'invention est de mettre en place l'échange sécurisé du « *MasterSecret* » entre modules de sécurité. Dans ce cas la mise en œuvre de *N* modules de sécurité permet le téléchargement d'au plus *N/TR* fichiers par seconde. En effet, comme les modules de la grille de modules de sécurité selon l'invention partagent le « MasterSecret » (dont il convient de rappeler qu'il sert à la composition de tout autre matériel cryptographique ou de chiffrement ultérieur au HANDSHAKE), les modules esclaves peuvent être autorisés à initier une session en mode RESUMPTION.

Ainsi, si le nombre de connexions TCP utilisées par le navigateur est limité à *NS*, le nombre optimal de module de sécurité *N* est égal à cette valeur (*N* = *NS*). On en déduit la limite de téléchargement de fichiers, en utilisant l'architecture de grille de sécurité selon l'invention : *NS/TF*.

5.5 Description d'une grille de modules de sécurité selon l'invention

On présente, en relation avec la figure 4 un module de sécurité sous la forme d'un circuit intégré de silicium (400), qualifié usuellement de "Tamper Resistant Device", littéralement un "composant qui résiste aux attaques", tel que par exemple le composant ST22 (produit par la société ST Microelectronics) et disponible sous différents formats tels que des cartes en PVC, (cartes à puce, carte SIM,…) intégrées dans des jetons USB, ou dans des mémoires MMC (MultiMedia Card).

Un tel module de sécurité incorpore tous les moyens sécurisés de stockage de données, et permet également l'exécution de logiciels dans un environnement sûr et protégé.

Plus précisément il comporte une unité centrale (CPU, 401), une mémoire ROM stockant le code du système d'exploitation (402), de la mémoire RAM (403), et une mémoire non volatile (NVR, 404), utilisée comme dispositif de stockage analogue à un disque dur, et qui contient par exemple un logiciel embarqué TLS. Un bus système (410) relie les différents organes du module sécurisé. L'interface avec le monde extérieur (420) est assurée par un port IO d'entrée/sortie (405), conforme à des standards tels que ISO 7816, USB, USB-OTG, ISO 7816-12, MMC, IEEE 802.3, IEEE 802.11, etc.

Les cartes à puces JAVA, communément désignées par le terme JAVACARD, sont une classe particulière de module de sécurité.

Dans au moins un mode de réalisation, un dispositif mettant en œuvre le procédé de l'invention se présente sous la forme dispositif portable, tel qu'un jeton ou une clé USB. Ce dispositif comprend, d'une part un moyen de stockage notamment d'un logiciel de type « Gestionnaire de Modules de Sécurité » selon l'invention et au moins deux lecteurs de cartes au format SIM. Le stockage du gestionnaire de modules de sécurité selon l'invention peut être réalisé sur un composant électronique spécifique du type FPGA (« *field-programmable gate array* » pour « réseau de portes programmables »)

Les lecteurs de carte à puce peuvent accueillir respectivement des modules de sécurité maîtres et des modules de sécurité esclaves pour composer une grille de modules de sécurité. Lorsqu'il est branché, par exemple à un ordinateur personnel le dispositif joue le rôle de fournisseur de ressources de sécurisation.

Le Gestionnaire de Modules de Sécurité réalise une interface entre l'ordinateur personnel et les modules de sécurité. Il est notamment capable de transmettre les commandes de création de clé secrète au module maître et les commandes de transmission de clé secrète préalablement calculée au module esclave.

Ainsi, dans ce mode de réalisation, l'invention permet de fournir, de manière très simple, une solution de sécurisation forte sans qu'il soit nécessaire de réaliser de nombreuses modifications dans les architectures de communication

existantes : au pire, il est nécessaire d'installer un driver spécifique au dispositif sur l'ordinateur sur lequel ce dispositif doit être branché : ceci sera valable par exemple pour les ordinateurs disposant de système d'exploitation plutôt ancien. Au mieux, le dispositif de l'invention est reconnu comme étant un lecteur de carte à puce standard, ne nécessitant aucune installation supplémentaire.

Dans ce mode de réalisation et dans tous les cas, le composant « Gestionnaire de Modules de Sécurité » est chargé de faire l'interface entre la grille de module de sécurité et le terminal dans lequel le dispositif est enfiché.

REVENDICATIONS

1. Procédé de production de données de sécurisation, permettant la mise en œuvre d'une session sécurisée entre une première et au moins une deuxième entité, selon un protocole d'établissement de sessions sécurisées, caractérisé en ce qu'il comprend :
   - une étape d'initialisation d'une troisième entité sécurisée liée à ladite première entité ;
   - une étape de génération d'au moins une partie desdites données de sécurisation au sein de ladite troisième entité ;
   - une première étape de transmission desdites données de sécurisation générées de ladite troisième entité sécurisée vers ladite première entité ;
   - une deuxième étape de transmission d'au moins une partie desdites données de sécurisation générées au sein de ladite troisième entité sécurisée à destination d'au moins une quatrième entité sécurisée préalablement initialisée et liée à ladite troisième entité sécurisée.
2. Procédé de production selon la revendication 1, caractérisé en ce que ladite troisième entité, dite entité maître, génère au moins une partie d'un secret partagé entre ladite première et ladite deuxième entité.
3. Procédé de production selon la revendication 2, caractérisé en ce que ladite au moins une partie desdites données de sécurisation générées transmise à ladite au moins une quatrième entité, dite entité esclave, comprend ledit secret partagé sous une forme chiffrée et au moins un identifiant de session de communication sécurisée.
4. Procédé de production selon l'une quelconque des revendications 1 et 2, caractérisé en ce que ledit protocole d'établissement de sessions sécurisées est le protocole SSL.
5. Procédé de production selon l'une quelconque des revendications 1 et 2, caractérisé en ce que ledit protocole d'établissement de sessions sécurisées est le protocole TLS.
6. Procédé de production selon la revendication 5, caractérisé en ce qu'il

comprend en outre :

- une étape de transmission, par ladite première entité, d'au moins un message à destination d'une unité fonctionnelle « RECORD » mise en œuvre au sein de ladite troisième entité ;
- une étape de réception, par ladite première entité, d'au moins un message en provenance de ladite unité fonctionnelle « RECORD » ;
- une étape de calcul d'un ensemble de clés par ladite troisième entité ;
- une étape de collecte dudit ensemble de clés disponibles par ladite première entité auprès de ladite troisième entité.

7. Procédé de production selon la revendication 1, caractérisé en ce que ladite deuxième étape de transmission est mise en œuvre par un gestionnaire de modules de sécurités qui obtient lesdites données de sécurisation à partir de ladite troisième entité.

8. Procédé de production selon la revendication 1, caractérisé en ce que ladite deuxième étape de transmission est mise en œuvre lors d'une phase de reprise de ladite session sécurisée.

9. Procédé d'établissement d'une session de communication sécurisée entre une première et au moins une deuxième entité, selon un protocole d'établissement de sessions sécurisées, caractérisé en ce qu'il comprend :

- une étape d'obtention d'un identifiant de session et d'un secret éphémère calculé lors d'une précédente session de communication sécurisée par une troisième entité sécurisée liée à ladite première entité à l'aide dudit procédé de production de données de sécurisation selon la revendication 1 ;
- une étape de transmission dudit identifiant de session et dudit secret éphémère à une quatrième entité sécurisée, préalablement initialisée et liée à ladite troisième entité sécurisée ;
- une étape d'établissement de ladite session de communication sécurisée en utilisant ladite quatrième entité sécurisée.

10. Dispositif de production de données de sécurisation, permettant la mise en

œuvre d'une session sécurisée entre une première et au moins une deuxième entité, selon un protocole d'établissement de sessions sécurisées, caractérisé en ce qu'il comprend :

- des moyens d'initialisation, d'une troisième entité sécurisée attachée à ladite première entité ;
- des moyens de génération d'au moins une partie desdites données de sécurisation ;
- des moyens de transmission desdites données de sécurisation vers ladite première entité ;
- des moyens de transmission d'au moins une partie desdites données de sécurisation générées au sein de ladite troisième entité sécurisée à destination d'au moins une quatrième entité sécurisée préalablement initialisée et liée à ladite troisième entité sécurisée.

11. Dispositif de production de données de sécurisation selon la revendication 10, caractérisé en ce que lesdits moyens de génération et lesdits moyens de transmission sont regroupés dans une carte à puce.

12. Dispositif portable, tel qu'un jeton USB, caractérisé en ce qu'il comprend un moyen de stockage d'un gestionnaire de modules de sécurité et au moins deux lecteurs de cartes au format SIM et un dispositif de production de donnée de sécurisation selon la revendication 10.

13. Produit programme d'ordinateur téléchargeable depuis un réseau de communication et/ou stocké sur un support lisible par ordinateur et/ou exécutable par un microprocesseur, caractérisé en ce qu'il comprend des instructions de code de programme pour l'exécution du procédé de production selon l'une au moins des revendications 1 à 8, lorsqu'il est exécuté sur un ordinateur.

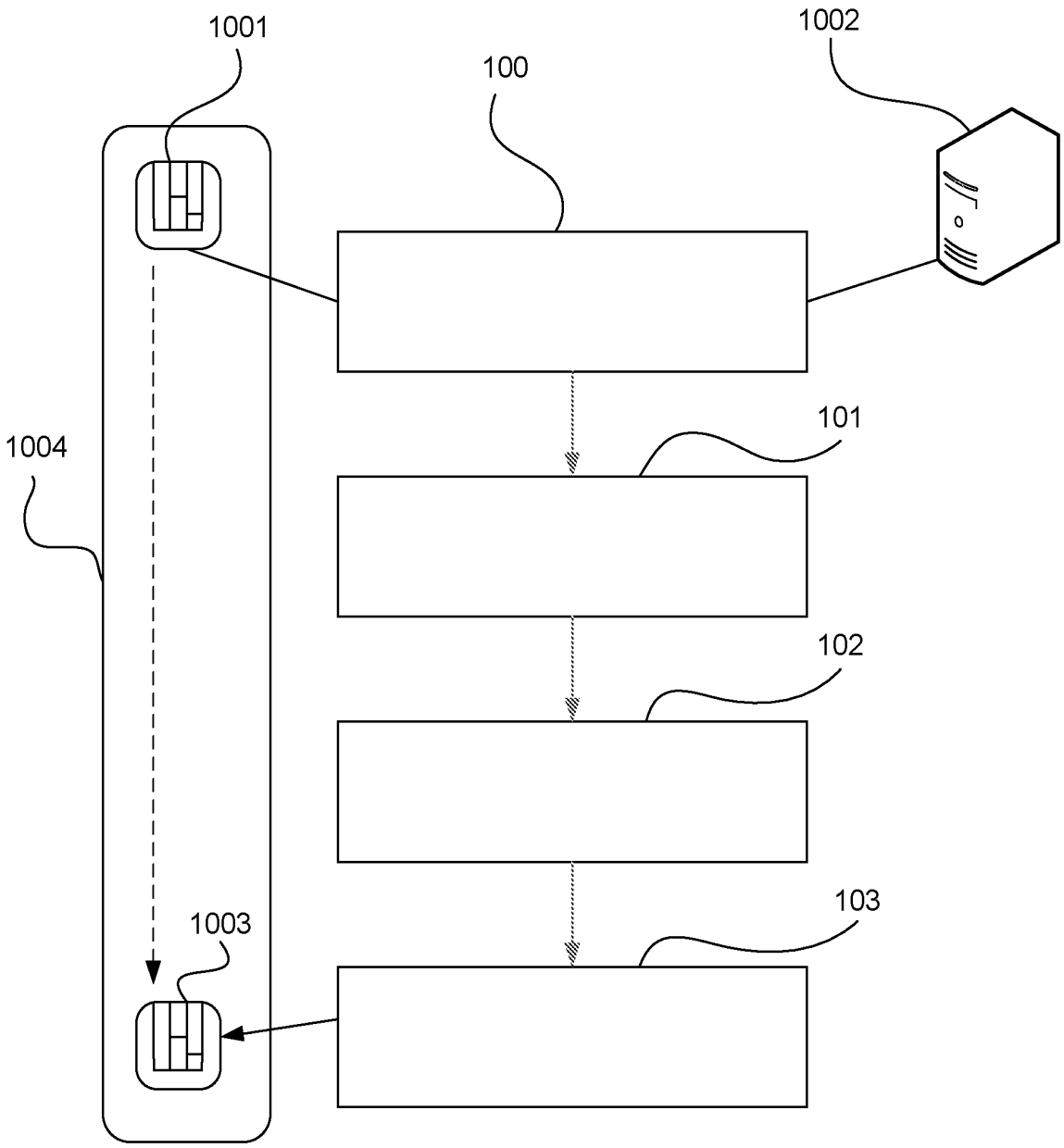


Figure 1

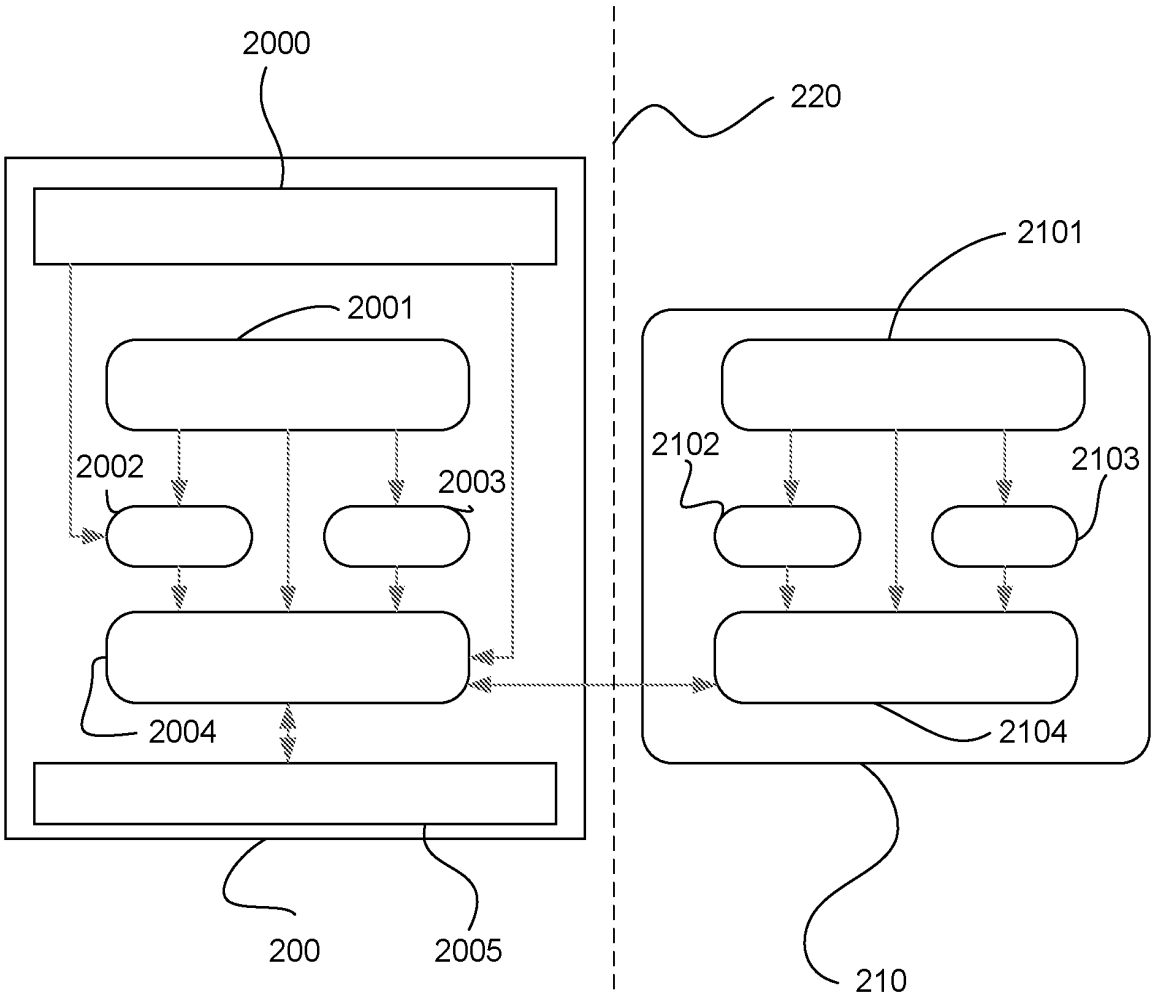


Figure 2

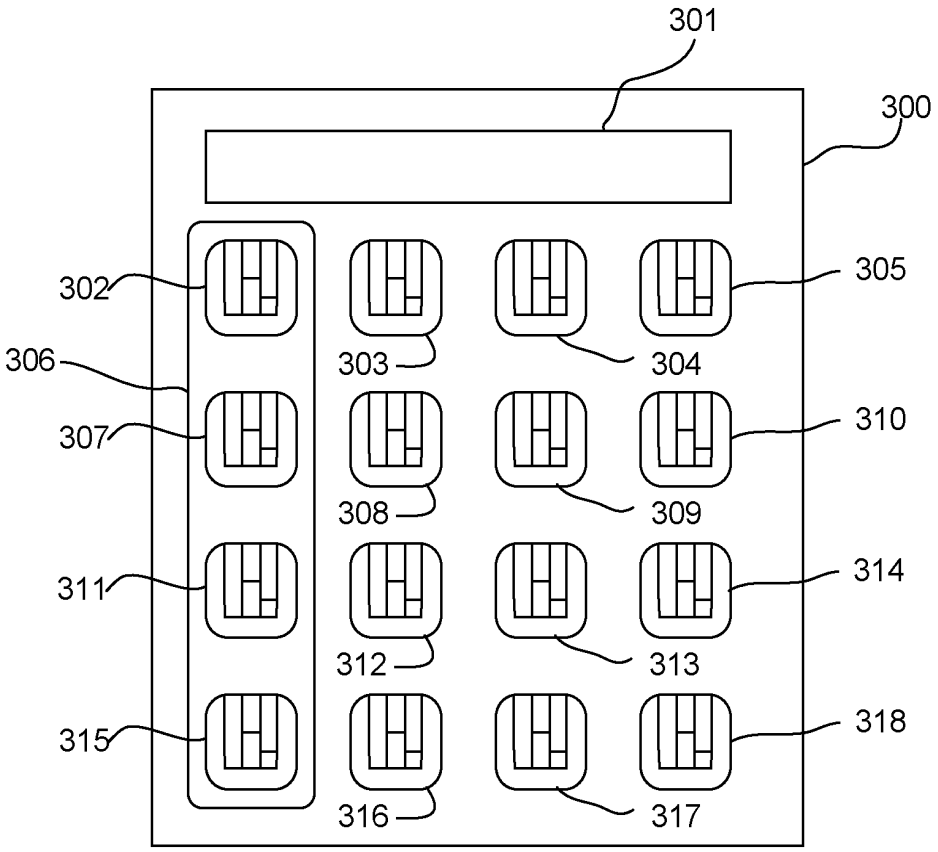

301
300
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318

Figure 3

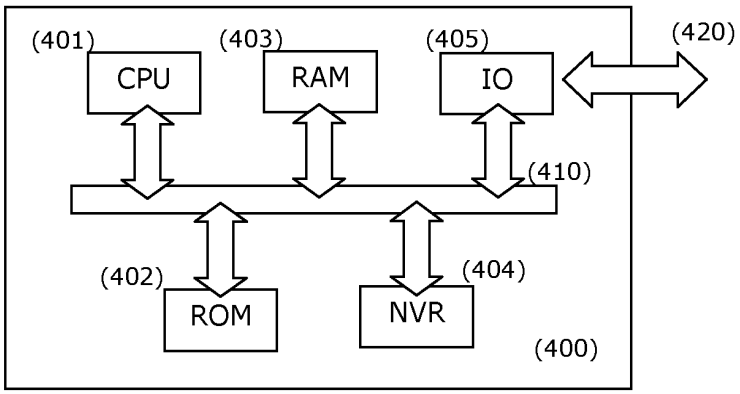


Figure 4

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 722240
FR 0951646

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS

Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
X	WO 2008/145558 A (GROUPE ECOLES TELECOMM [FR]; URIEN PASCAL [FR]) 4 décembre 2008 (2008-12-04) * le document en entier *	1-13	H04L29/06 G06K19/07
A	WO 01/60040 A (BULL CP8 [FR]; URIEN PASCAL [FR]) 16 août 2001 (2001-08-16) * abrégé * * page 6, ligne 22 - page 7, ligne 20 * * page 28, ligne 19 - page 30, ligne 14 *	1-13	
A	WO 2006/021865 A (AXALTO SA [FR]; SMADJA PHILIPPE [FR]; AUSSEL JEAN-DANIEL [FR]) 2 mars 2006 (2006-03-02) * abrégé * * page 2, ligne 9 - page 3, ligne 10 *	1-13	
A	EP 1 349 032 A (UBS AG [CH]) 1 octobre 2003 (2003-10-01) * abrégé *	1-13	
A	WO 99/39475 A (TANDEM COMPUTERS INC [US]) 5 août 1999 (1999-08-05) * abrégé * * page 7, ligne 7 - ligne 31 *	1-13	DOMAINES TECHNIQUES RECHERCHÉS (IPC) H04L G06F

Date d'achèvement de la recherche	Examinateur
29 décembre 2009	Bertolissi, Edy

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 0951646 FA 722240

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 29-12-2009
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
WO 2008145558 A	04-12-2008	FR 2916592 A1	28-11-2008
WO 0160040 A	16-08-2001	AT 345003 T	15-11-2006
		AU 3564801 A	20-08-2001
		CA 2366568 A1	16-08-2001
		CN 1363171 A	07-08-2002
		DE 60124367 T2	30-08-2007
		EP 1208684 A2	29-05-2002
		FR 2805062 A1	17-08-2001
		HK 1048906 A1	18-03-2005
		JP 3845018 B2	15-11-2006
		JP 2003522361 T	22-07-2003
		TW 509847 B	11-11-2002
		US 2002138549 A1	26-09-2002
WO 2006021865 A	02-03-2006	AT 445195 T	15-10-2009
		CN 101027676 A	29-08-2007
		JP 2008511232 T	10-04-2008
		US 2008263649 A1	23-10-2008
EP 1349032 A	01-10-2003	AT 254773 T	15-12-2003
		DE 10212619 A1	09-10-2003
		DE 60200093 D1	24-12-2003
		DE 60200093 T2	22-04-2004
		US 2003177392 A1	18-09-2003
WO 9939475 A	05-08-1999	TW 413988 B	01-12-2000
		US 6378072 B1	23-04-2002
		US 2002073316 A1	13-06-2002

EPO FORM P0465