(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
30 juin 2016 (30.06.2016)

WIPO | PCT

(10) Numéro de publication internationale
WO 2016/102832 A1

(51) Classification internationale des brevets :
G06F 21/44 (2013.01) *G06F 21/12* (2013.01)

(21) Numéro de la demande internationale :
PCT/FR2015/053586

(22) Date de dépôt international :
17 décembre 2015 (17.12.2015)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
1463169 22 décembre 2014 (22.12.2014) FR

(71) Déposant : **OBERTHUR TECHNOLOGIES** [FR/FR]; 420 rue d'Estienne d'Orves, 92700 Colombes (FR).

(72) Inventeurs : **DOTTAX, Emmanuelle**; 420 rue d'Estienne d'Orves, 92700 Colombes (FR). **MURESIANU, Philippe**; 420 rue d'Estienne d'Orves, 92700 Colombes (FR). **SARTORI, Michele**; 420 rue d'Estienne d'Orves, 92700 Colombes (FR).

(74) Mandataires : **ORSINI, Fabienne** et al.; 14-16 rue Ballu, 75009 Paris (FR).

(81) États désignés *(sauf indication contraire, pour tout titre de protection nationale disponible)* : AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) États désignés *(sauf indication contraire, pour tout titre de protection régionale disponible)* : ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), eurasien (AM, AZ, BY, KG, KZ, RU, TJ, TM), européen (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Publiée :

— *avec rapport de recherche internationale (Art. 21(3))*

(54) **Title** : METHOD FOR AUTHENTICATING AN APPLICATION, ASSOCIATED ELECTRONIC APPARATUS AND COMPUTER PROGRAM

(54) **Titre** : PROCÉDÉ D'AUTHENTIFICATION D'UNE APPLICATION, APPAREIL ÉLECTRONIQUE ET PROGRAMME D'ORDINATEUR ASSOCIÉS

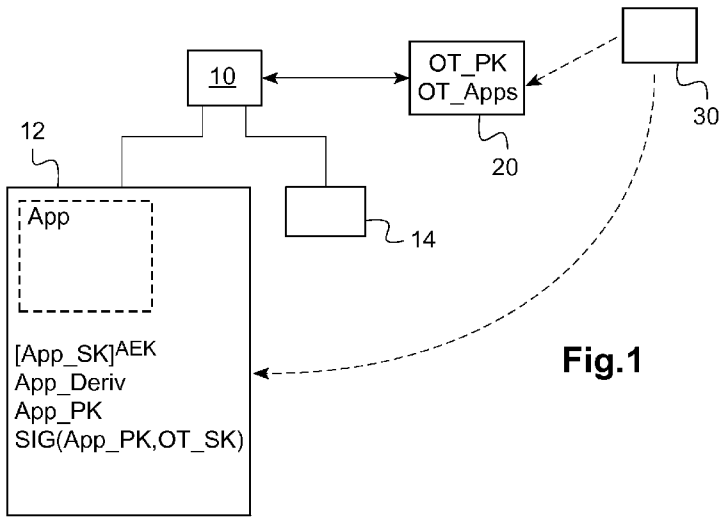


Fig.1

(57) **Abstract** : The invention relates to a method for authenticating an application (App) executed by a processor (10) of an electronic apparatus, which comprises the following steps: deriving a cryptographic key (AEK) from a piece of data (OT_Apps) received from an external electronic entity (20); decrypting, by means of the derived cryptographic key (AEK), an encrypted stored secret key ($[App\_SK]^{AEK}$); transmitting authentication data prepared by means of the decrypted secret key (App_SK). The invention also proposes an associated electronic apparatus and computer program.

(57) **Abrégé** :

[Suite sur la page suivante]

WO 2016/102832 A1

L'invention concerne un procédé d'authentification d'une application (App) exécutée par un processeur (10) d'un appareil électronique, comprenant les étapes suivantes : - dérivation d'une clé cryptographique (AEK) à partir d'une donnée (OT_Apps) reçue d'une entité électronique extérieure (20); - déchiffrement, au moyen de la clé cryptographique dérivée (AEK), d'une clé secrète mémorisée chiffrée ($[App\_SK]^{AEK}$); - émission de données d'authentification préparées au moyen de la clé secrète déchiffrée (App_SK). Un appareil électronique et un programme d'ordinateur associés sont également proposés.

Procédé d'authentification d'une application, appareil électronique et programme d'ordinateur associés

DOMAINE TECHNIQUE AUQUEL SE RAPPORTE L'INVENTION

La présente invention concerne l'authentification des applications exécutables sur un processeur.

Elle concerne plus particulièrement un procédé d'authentification d'une application, ainsi qu'un appareil électronique et un programme d'ordinateur associés.

L'invention s'applique particulièrement avantageusement dans le cas où le processeur sur lequel s'exécute l'application échange des données avec un module sécurisé.

ARRIERE-PLAN TECHNOLOGIQUE

Dans les procédés de traitement et d'échanges de données impliquant un processeur, sur lequel s'exécute une application, et un dispositif électronique tiers, il arrive que l'application doive s'authentifier auprès du dispositif électronique tiers, afin de garantir à celui-ci qu'il échange des données avec l'application prévue à cet effet, et non avec un programme malveillant (ou "*malware*").

Afin de s'authentifier auprès du dispositif électronique tiers, il a été proposé que l'application transmette à celui-ci des données d'authentification (par exemple une signature) préparées au moyen d'une clé secrète (par exemple une clé privée) connue seulement de l'application.

Dans cette configuration toutefois, la diffusion d'une telle application dans des environnements non-sécurisés, tels que les appareils électroniques sur lesquels une telle application peut être installée, risque de rendre la clé secrète accessible à des personnes malintentionnées.

OBJET DE L'INVENTION

Dans ce contexte, la présente invention propose un procédé d'authentification d'une application exécutée par un processeur d'un appareil électronique, caractérisé en ce qu'il comprend les étapes suivantes :

- dérivation d'une clé cryptographique à partir d'une donnée reçue d'une entité électronique extérieure ;

- déchiffrement, au moyen de la clé cryptographique dérivée, d'une clé secrète mémorisée chiffrée ;

- émission de données d'authentification préparées au moyen de la clé secrète déchiffrée.

La clé secrète peut être ainsi utilisée pour la préparation des données d'authentification après réception des données d'une entité électronique extérieure, mais n'est pas accessible sans ces données et ne pourra donc en revanche pas être utilisée par une personne malintentionnée.

Selon des caractéristiques optionnelles et donc non limitatives :

- la clé secrète chiffrée est mémorisée dans une mémoire non-volatile réinscriptible ;

- la clé secrète déchiffrée est mémorisée dans une mémoire vive ;

- le procédé d'authentification comprend des étapes de réception d'un défi et de préparation desdites données d'authentification par application au défi reçu d'un algorithme cryptographique utilisant la clé secrète déchiffrée ;

- ladite donnée reçue est préparée par l'entité électronique extérieure sur la base d'une donnée reçue de l'application et d'une clé mémorisée dans l'entité électronique extérieure ;

- les données d'authentification sont émises à destination d'un module sécurisé ;

- le défi est généré, par exemple par tirage aléatoire, par le module sécurisé et émis par le module sécurisé à destination de l'application ;

- l'entité électronique extérieure est le module sécurisé ;

- l'entité électronique extérieure est un serveur (et est donc distincte du module sécurisé) ;

- la clé secrète est une clé privée ;

- le procédé comprend une étape d'émission par le module sécurisé d'un message signé au moyen d'une clé privée mémorisée dans le module sécurisé à destination d'un serveur ;

- le processus d'authentification du module sécurisé auprès de l'application utilise une clé publique mémorisée au sein de l'appareil électronique en association avec un code d'authentification de message produit par une autorité extérieure ;

- l'appareil électronique est un téléphone mobile ou une tablette numérique ;

- l'appareil électronique comprend un élément de maintien sur une partie

du corps de l'utilisateur ;

- le procédé comprend une étape d'émission par l'application d'un message signé au moyen de la clé secrète déchiffrée à destination d'un serveur ;

- le procédé comprend une étape de déchiffrement d'une partie d'un code exécutable de l'application au moyen de la clé cryptographique dérivée ;

- le module sécurisé est un élément sécurisé.

Le procédé d'authentification peut comprendre en outre les étapes suivantes :

- vérification, par le module sécurisé et au moyen d'une première clé publique mémorisée dans le module sécurisé, d'une signature annexée à une seconde clé publique associée à ladite clé privée (dans une infrastructure à clé publique ou PKI, pour "*Public Key Infrastructure*") ;

- vérification, par le module sécurisé, desdites données d'authentification au moyen de la seconde clé publique.

Le procédé d'authentification peut également comprendre un processus d'authentification du module sécurisé auprès de l'application.

L'invention propose également un appareil électronique comprenant un processeur et au moins une mémoire mémorisant une application exécutable par le processeur, l'application étant conçue de sorte que le processeur mette en œuvre un procédé tel que proposé ci-dessus lorsque l'application est exécutée par le processeur.

De même, l'invention propose que l'application susmentionnée soit un programme d'ordinateur comprenant des instructions exécutables par un processeur d'un appareil électronique (notamment par le processeur de l'appareil électronique susmentionné) et conçues de sorte que le processeur mette en œuvre un procédé tel que proposé ci-dessus lorsque ces instructions sont exécutées par le processeur.

DESCRIPTION DETAILLEE D'UN EXEMPLE DE REALISATION

La description qui va suivre en regard des dessins annexés, donnés à titre d'exemples non limitatifs, fera bien comprendre en quoi consiste l'invention et comment elle peut être réalisée.

Sur les dessins annexés :

- la figure 1 représente schématiquement les éléments principaux d'un premier exemple de système dans lequel peut être mise en œuvre l'invention ;

- la figure 2 représente les étapes principales d'un premier exemple de procédé d'authentification d'une application ;

- la figure 3 représente les étapes principales d'un second exemple de procédé d'authentification d'une application ;

- la figure 4 représente les étapes principales d'un troisième exemple de procédé d'authentification d'une application ;

- la figure 5 représente schématiquement les éléments principaux d'un second exemple de système dans lequel peut être mise en œuvre l'invention ; et

- la figure 6 représente les étapes principales d'un quatrième exemple de procédé d'authentification d'une application.

La figure 1 représente schématiquement les éléments principaux d'un premier exemple de système, ici un appareil électronique, dans lequel peut être mise en œuvre l'invention. Cet appareil électronique est par exemple un téléphone mobile, une tablette numérique, une montre connectée ou des lunettes connectées. Dans ces derniers cas, l'appareil électronique comprend un élément de maintien sur une partie du corps de l'utilisateur (par exemple un bracelet dans le cas d'une montre connectée, ou des branches dans le cas de lunettes connectées).

Ces éléments comprennent un processeur 10 (par exemple un microprocesseur), une mémoire non-volatile réinscriptible 12, une mémoire vive 14 et un module sécurisé 20.

Ces éléments font ici partie de l'appareil électronique susmentionné. En variante, ces éléments pourraient être répartis dans plusieurs appareils électroniques formant un système, le processeur 10, la mémoire non-volatile 12 et la mémoire vive 14 étant par exemple compris dans un même appareil électronique, tandis que le module sécurisé est compris dans un autre appareil électronique.

La mémoire non-volatile réinscriptible 12 mémorise des applications dont l'exécution par le processeur 10 permet la mise en œuvre de procédés de traitement de données et d'échanges de données, ici notamment avec le module sécurisé 20. La mémoire non-volatile réinscriptible 12 mémorise ainsi notamment une application App qui cherche, lors de son exécution par le processeur 10, à s'authentifier auprès du module sécurisé 20 et met notamment en œuvre pour ce faire les étapes décrites ci-dessous en référence à la figure 2.

La mémoire non-volatile réinscriptible 12 mémorise également des données utilisées lors des procédés de traitement mis en œuvre par le processeur 10.

En particulier, la mémoire non-volatile réinscriptible 12 mémorise les données cryptographiques suivantes, utilisées comme décrit ci-dessous dans le cadre du procédé de la figure 2 :

- une clé privée App_SK attribuée à l'application App, mémorisée dans la mémoire non-volatile réinscriptible 12 sous forme chiffrée par une clé cryptographique AEK (d'où la notation $[App\_SK]^{AEK}$ adoptée en figure 1) ;
- des données de dérivation App_Deriv ;
- une clé publique App_PK associée à la clé privée App_SK (dans une infrastructure à clé publique ou PKI, pour "*Public Key Infrastructure*") et la signature SIG(App_PK,OT_SK) de cette clé publique App_PK.

Ces données sont spécifiques à l'application concernée (ici l'application App) et ont été calculées par une autorité extérieure 30, puis associées à l'application App et par exemple inscrites dans la mémoire non-volatile réinscriptible 12 lors de l'installation de l'application App dans cette même mémoire non-volatile réinscriptible 12 (comme schématiquement représenté en pointillés en figure 1).

La clé cryptographique AEK (avec laquelle est chiffrée la clé privée App_SK) est obtenue (notamment au préalable, au niveau de l'autorité extérieure 30, en vue du chiffrement de la clé privée App_SK) par application d'une fonction de dérivation F à une clé racine OT_Apps (commune à toutes les applications) et aux données de dérivation App_Deriv susmentionnées (spécifiques à l'application concernée) : AEK = F(OT_Apps,App_Deriv).

On remarque qu'une partie du code exécutable (c'est-à-dire des instructions exécutables par le processeur 10) de l'application App peut également être chiffrée au moyen de la clé cryptographique AEK. Un tel chiffrement peut par exemple être réalisé par le fournisseur de l'application. (Une autre partie du code est dans ce cas mémorisée dans la mémoire non-volatile réinscriptible 12 sans chiffrement afin de pouvoir effectuer les étapes décrites ci-dessous, au moins jusqu'au déchiffrement de la partie chiffrée du code.)

La signature SIG(App_PK,OT_SK) est quant à elle obtenue (également au niveau de l'autorité extérieure 30) par application à la clé publique App_PK

d'un algorithme cryptographique de signature utilisant une clé privée de l'autorité extérieure OT_SK (cette clé privée étant utilisée quelle que soit l'application concernée).

Au cours des procédés mis en œuvre par le processeur 10, et notamment le procédé de la figure 2, la mémoire vive 14 peut mémoriser les données manipulées par ces procédés, notamment comme décrit ci-après.

Le module sécurisé 20 est par exemple un élément sécurisé, tel qu'un élément sécurisé intégré à l'appareil électronique (ou eSE pour "*embedded Secure Element*"). En variante, il pourrait s'agir d'une carte à microcircuit, par exemple de type UICC (pour "*Universal Integrated Circuit Card*") ou eUICC (pour "*embedded Universal Integrated Circuit Card*"), ou d'une carte de type micro-SD sécurisée.

Le module sécurisé 20 comprend un processeur et une mémoire non-volatile réinscriptible. Comme représenté en figure 1, cette mémoire non-volatile réinscriptible mémorise notamment la clé racine OT_Apps déjà mentionnée et la clé publique OT_PK associée à la clé privée OT_SK susmentionnée (dans une infrastructure à clé publique). Ces clés ont par exemple été inscrites dans le module sécurisé lors d'une étape de personnalisation de ce module sécurisé 20 (comme schématiquement représenté en pointillés en figure 1). Selon une possibilité de réalisation, potentiellement utilisable lorsque l'appareil électronique précité comprend des moyens de télécommunication, ces clés pourraient être mises à jour à distance (selon une technique dite OTA pour "*Over-The-Air*") depuis un serveur de l'autorité extérieure 30.

Les procédés mis en œuvre par le module sécurisé 20 comme décrit ci-dessous résultent de l'exécution, par le processeur du module sécurisé 20, d'instructions de programmes mémorisés dans la mémoire non-volatile réinscriptible du module sécurisé 20, ou d'une autre mémoire du module sécurisé 20.

En variante, le module sécurisé 20 pourrait être réalisé sous forme d'un circuit intégré à application spécifique (ou ASIC pour "Application Specific Integrated Circuit"), programmé pour mettre en œuvre les procédés décrits ci-dessous.

Le processeur 10 est relié au module sécurisé 20, par exemple au moyen d'un bus (ou en variante via une liaison de type ISO7816 ou SPI ou

HCI/SWP), et peut ainsi, du fait de l'exécution des instructions des applications mémorisées en mémoire non-volatile 12 ou éventuellement en mémoire vive 14, échanger des données avec le module sécurisé 20. Le processeur 10 et le module sécurisé 20 pourraient en variante échanger des données par l'intermédiaire d'un réseau téléphonique et/ou informatique (par exemple en utilisant des protocoles de type GSM et/ou UMTS et/ou TCP/IP et/ou Bluetooth), notamment dans les cas mentionnés ci-dessus où le processeur 10 et le module sécurisé 20 ne font pas partie du même appareil électronique.

La figure 2 représente les étapes principales d'un premier exemple de procédé d'authentification d'une application. Un tel procédé est ici mis en œuvre dans le système de la figure 1.

Comme indiqué ci-dessous, ce procédé comprend des échanges entre le processeur 10 et le module sécurisé 20. Ces échanges peuvent être réalisés directement entre ces éléments, sans mise en place d'un canal sécurisé. Toutefois, selon une variante envisageable, on peut au préalable établir un canal sécurisé entre le processeur 10 et le module sécurisé 20 et réaliser les échanges décrits ci-dessous via ce canal sécurisé, ce qui permet d'empêcher un attaquant passif d'avoir accès (par simple observation) aux données échangées.

Pour ce faire, le module sécurisé 20 et le processeur 10 utilisent par exemple un schéma de type Diffie-Hellman, en générant chacun de leur côté une clé éphémère, en signant cette clé éphémère avec leur clé secrète, en vérifiant la signature effectuée par l'autre module et, en cas de vérification, en déduisant de ces éléments un secret commun et des clés de session à utiliser pour chiffrer les échanges ultérieurs.

Quoiqu'il en soit, le procédé de la figure 2 débute à l'étape E2 par l'émission par le processeur 10 (du fait de l'exécution de l'application App) de la clé publique App_PK et de la signature SIG(App_PK,OT_SK) associée.

Le module sécurisé 20 reçoit la clé publique App_PK et la signature SIG(App_PK,OT_SK) associée à l'étape E4.

Le module sécurisé 20 peut ainsi vérifier à l'étape E6 la signature reçue SIG(App_PK,OT_SK) à l'aide de la clé publique OT_PK (associée à la clé privée OT_SK et mémorisée dans le module sécurisé comme déjà indiqué), par exemple en appliquant la signature SIG(App_PK,OT_SK) un algorithme cryptographique de vérification de signature utilisant la clé publique OT_PK et en comparant le résultat

obtenu à la clé publique App_PK (éventuellement transformée par une fonction de hachage utilisée pour former la signature). Selon une variante envisageable, la signature SIG(App_PK,OT_SK) peut être vérifiée en appliquant, à la clé publique App_PK et à la signature reçue SIG(App_PK,OT_SK), un algorithme cryptographique de vérification de signature utilisant la clé publique OT_ PK et produisant (directement) un résultat positif ou négatif.

Si la signature n'est pas correctement vérifiée, le module sécurisé 20 met fin au processus d'authentification, en renvoyant par exemple un message d'erreur au processeur 10.

Si la signature est correctement vérifiée, le procédé de la figure 2 se poursuit à l'étape E8 à laquelle le module sécurisé 20 génère un défi (ou "*challenge*" selon l'appellation anglo-saxonne) SE_Ch, par exemple par tirage aléatoire.

Le module sécurisé 20 émet ensuite à l'étape E10 le défi SE_Ch et la clé racine OT_Apps à destination du processeur 10 sur lequel s'exécute l'application App.

Le processeur 10 reçoit le défi SE_Ch et la clé racine OT_Apps à l'étape E12 et met alors en œuvre les étapes suivantes du fait de l'exécution de l'application App.

Le processeur 10 détermine à l'étape E14 la clé cryptographique AEK (déjà mentionnée plus haut) en appliquant la fonction de dérivation F à la clé racine OT_Apps reçue à l'étape E12 et aux données de dérivation App_Deriv mémorisées dans la mémoire non-volatile 12.

Le processeur 10 peut alors déchiffrer à l'étape E16 la clé privée App_SK par application à la version chiffrée $[App\_SK]^{AEK}$ (mémorisée dans la mémoire non-volatile 12) d'un algorithme cryptographique de déchiffrement utilisant la clé cryptographique AEK déterminée à l'étape E14. La clé privée App_SK ainsi obtenue (c'est-à-dire déchiffrée) peut être mémorisée dans la mémoire vive 14 jusqu'à la mise en œuvre de l'étape E18 décrite ci-dessous (puis éventuellement supprimée ensuite).

Dans les cas où une partie du code de l'application App est chiffrée au moyen de la clé cryptographique AEK comme décrit ci-dessus, le processeur 10 peut alors également procéder au déchiffrement de la partie chiffrée du code de l'application App.

Le processeur 10 génère à l'étape E18 une signature SIG' par application au défi SE_Ch reçu à l'étape E12 d'un algorithme cryptographique de signature utilisant la clé privée App_SK (déchiffrée à l'étape E16).

Le processeur 10 envoie la signature SIG' ainsi obtenue au module sécurisé 20 à l'étape E20.

Le module sécurisé 20 reçoit la signature SIG' à l'étape E22 et peut ainsi vérifier la signature SIG' à l'étape E24 en utilisant la clé publique App_PK (dont la signature par l'autorité extérieure a été vérifiée à l'étape E6), par exemple en appliquant à la signature SIG' reçue un algorithme cryptographique de vérification de signature utilisant la clé publique App_PK, puis en comparant le résultat obtenu au défi SE_Ch généré à l'étape E8. Selon une variante envisageable, la signature SIG' peut être vérifiée en appliquant, au défi SE_Ch et à la signature reçue SIG', un algorithme cryptographique de vérification de signature utilisant la clé publique App_PK et produisant (directement) un résultat positif ou négatif.

Si la signature SIG' n'est pas correctement vérifiée, le procédé d'authentification a échoué.

Si la signature SIG' est correctement vérifiée à l'étape E24, le processeur 10 sur lequel s'exécute l'application App a ainsi démontré qu'il détenait la clé privée App_SK et s'est donc authentifié.

La figure 3 représente les étapes principales d'un second exemple de procédé d'authentification d'une application.

Ce procédé est mis en œuvre dans un système du type de celui de la figure 1, dans lequel la mémoire non-volatile réinscriptible 12 mémorise en outre des données de dérivation additionnelles App_AEK_Deriv. Ces données sont également inscrites dans la mémoire non-volatile 12 lors de l'installation de l'application App (préparée par ou en collaboration avec l'autorité extérieure).

Par ailleurs, dans ce mode de réalisation, la clé cryptographique AEK utilisée pour le chiffrement de la clé privée App_SK est obtenue en utilisant non seulement les données de dérivation App_Deriv, mais aussi les données de dérivation App_AEK_Deriv, par exemple comme suit :

- une clé dérivée OT_App_K est déterminée par application d'une fonction de dérivation G à la clé racine OT_Apps et aux données de dérivation App_Deriv, soit OT_App_K = G(OT_App,App_Deriv) ;

- la clé cryptographique AEK est déterminée par application d'une

fonction de dérivation F à la clé dérivée OT_App_K et aux données de dérivation additionnelles App_AEK_Deriv, soit AEK = F(OT_App_K,App_AEK_Deriv).

Comme pour le procédé de la figure 2, il est envisageable de manière optionnelle d'établir au préalable un canal sécurisé entre le processeur 10 et le module sécurisé 20 et de réaliser alors les échanges décrits ci-dessous via ce canal sécurisé.

Le procédé d'authentification de la figure 3 débute à l'étape E30 par l'émission par le processeur 10 (du fait de l'exécution de l'application App) de la clé publique App_PK et de la signature SIG(App_PK,OT_SK) associée, ainsi que des données de dérivation App_Deriv.

Le module sécurisé 20 reçoit la clé publique App_PK, la signature SIG(App_PK,OT_SK) associée et les données de dérivation App_Deriv à l'étape E32.

Le module sécurisé 20 peut ainsi vérifier à l'étape E34 la signature reçue SIG(App_PK,OT_SK) à l'aide de la clé publique OT_PK, comme expliqué ci-dessus pour l'étape E6 dans le cadre de la figure 2.

Si la signature n'est pas correctement vérifiée, le module sécurisé 20 met fin au processus d'authentification, en renvoyant par exemple un message d'erreur au processeur 10.

Si la signature est correctement vérifiée, le procédé de la figure 2 se poursuit à l'étape E36 à laquelle le module sécurisé 20 génère un défi SE_Ch, par exemple par tirage aléatoire.

Le module sécurisé 20 détermine alors à l'étape E38 la clé dérivée OT_App_K par application de la fonction de dérivation G à la clé racine OT_Apps (mémorisée comme indiqué plus haut et visible en figure 1 dans le module sécurisé 20) et aux données de dérivation App_Deriv :

OT_App_K = G (OT_Apps,App_Deriv).

Le module sécurisé 20 émet ensuite à l'étape E40 le défi SE_Ch et la clé dérivée OT_App_K à destination du processeur 10 sur lequel s'exécute l'application App.

On remarque que la clé transmise (ici OT_App_K) est ainsi différente selon l'application (ici App) avec laquelle le module sécurisé est en communication (puisque les données de dérivation App_Deriv sont variables selon l'application), ce qui évite d'émettre une clé toujours identique.

Le processeur 10 reçoit le défi SE_Ch et la clé dérivée OT_App_K à l'étape E42 et met alors en œuvre les étapes suivantes du fait de l'exécution de l'application App.

Le processeur 10 détermine à l'étape E44 la clé cryptographique AEK en appliquant la fonction de dérivation F à la clé dérivée OT_App_K reçue à l'étape E12 et aux données de dérivation additionnelles App_AEK_Deriv mémorisées comme indiqué plus haut dans la mémoire non-volatile 12.

Le procédé se poursuit ensuite de manière identique à ce qui a été décrit ci-dessus en référence à la figure 2, à partir de l'étape E16.

La figure 4 représente les étapes principales d'un troisième exemple de procédé d'authentification d'une application.

Ce procédé est mis en œuvre dans un système du type de celui de la figure 1, dans lequel la mémoire non-volatile réinscriptible 12 mémorise en outre des données de dérivation App_AEK_Deriv (comme pour le second exemple de procédé d'authentification décrit ci-dessus en référence à la figure 3) et des données de dérivation App_AIK_Deriv. Ces données sont spécifiques à l'application concernée (ici l'application App) et inscrites dans la mémoire non-volatile 12 lors de l'installation de l'application App (préparée par ou en collaboration avec l'autorité extérieure).

Par ailleurs, dans ce mode de réalisation, la clé cryptographique AEK utilisée pour le chiffrement de la clé privée App_SK est obtenue conformément à ce qui a été décrit ci-dessus dans le cadre de la description de la figure 3.

Le module sécurisé 20 mémorise en outre une clé privée d'authentification OT_Auth_SK et la mémoire non-volatile 12 mémorise (en association avec l'application App) la clé publique OT_Auth_PK associée à la clé privée d'authentification OT_Auth_SK (dans une infrastructure à clé publique) et un code d'authentification MAC de cette clé publique OT_Auth_PK.

Un tel code d'authentification MAC (pour "*Message Authentication Code*") est produit au préalable par l'autorité extérieure 30 par application à la clé publique OT_Auth_PK d'un algorithme cryptographique de génération de code d'authentification de message utilisant une clé AIK obtenue par dérivation à partir de la clé dérivée OT_App_K et des données de dérivation App_AIK_Deriv, ici en utilisant la fonction de dérivation F déjà mentionnée :

AIK = F(OT_App_K,App_AIK_Deriv).

Il est également possible d'utiliser la clé AIK afin d'obtenir un code d'authentification de l'application App (qui pourrait alors être mémorisé annexé à l'application App et utilisé pour vérifier l'intégrité de l'application App).

Le procédé de la figure 4 débute de manière identique à celui de la figure 3 par les étapes E30 à E44 décrites ci-dessus.

Le procédé de la figure 4 se poursuit ensuite à l'étape E46 à laquelle le processeur 10 déchiffre la clé privée App_SK par application à la version chiffrée $[App\_SK]^{AEK}$ (mémorisée dans la mémoire non-volatile 12) d'un algorithme cryptographique de déchiffrement utilisant la clé cryptographique AEK déterminée à l'étape E44. La clé privée App_SK ainsi obtenue est mémorisée dans la mémoire vive 14 au moins jusqu'à la mise en œuvre de l'étape E48 décrite ci-dessous.

Le processeur 10 génère alors à l'étape E48 une signature SIG' par application au défi SE_Ch reçu à l'étape E42 d'un algorithme cryptographique de signature utilisant la clé privée App_SK (déchiffrée à l'étape E46).

Le processeur 10 génère un défi App_Ch, par exemple par tirage aléatoire, à l'étape E50.

Le processeur 10 envoie la signature SIG' et le défi App_Ch au module sécurisé 20 à l'étape E52.

Comme déjà indiqué pour les autres modes de réalisation, les étapes E46 à E52 sont mises en œuvre du fait de l'exécution par le processeur de l'application App utilisée dans le présent mode de réalisation.

Le module sécurisé 20 reçoit la signature SIG' à l'étape E54 et peut ainsi vérifier la signature SIG' à l'étape E56 en utilisant la clé publique App_PK (dont la signature par l'autorité extérieure a été vérifiée à l'étape E34), par exemple comme décrit ci-dessus à propos de l'étape E24.

Si la signature SIG' n'est pas correctement vérifiée, le procédé d'authentification a échoué.

Si la signature SIG' est correctement vérifiée à l'étape E56, le processeur 10 sur lequel s'exécute l'application App s'est authentifié.

Le procédé décrit ici se poursuit alors en vue de l'authentification du module sécurisé 20 par le processeur 10 sur lequel s'exécute l'application App.

Pour ce faire, le module sécurisé 20 génère à l'étape E58 une signature SIG'' par application au défi App_Ch reçu à l'étape E54 d'un algorithme

cryptographique de signature utilisant la clé privée d'authentification OT_Auth_SK (mémorisée dans le module sécurisé comme indiqué plus haut).

Le module sécurisé 20 envoie la signature SIG'' ainsi obtenue au processeur 10 à l'étape E60.

Le processeur 10 reçoit la signature SIG'' à l'étape E62 et met en œuvre les étapes qui suivent du fait de l'exécution de l'application App utilisée dans ce mode de réalisation.

Le processeur 10 détermine à l'étape E64 la clé AIK par dérivation à partir de la clé dérivée OT_App_K (reçue à l'étape E42) et des données de dérivation App_AIK_Deriv (mémorisées dans la mémoire non-volatile 12) en utilisant la fonction de dérivation F : AIK = F(OT_App_K,App_AIK_Deriv).

Le processeur 10 peut ainsi vérifier à l'étape E66 que le code d'authentification MAC mémorisé en association avec la clé publique OT_Auth_PK est bien égal au résultat de l'application, à la clé publique OT_Auth_PK mémorisée, de l'algorithme de génération de code d'authentification de message utilisant la clé AIK obtenue à l'étape E64, ce qui permet de s'assurer de l'intégrité de la clé publique OT_Auth_PK mémorisée.

Si cette vérification est un échec, le processeur 10 met fin au procédé d'authentification, en envoyant par exemple un message d'erreur au module sécurisé 20.

Si la vérification de l'étape E66 est correctement effectuée, le processeur 10 vérifie la signature SIG'' à l'étape E68 en utilisant la clé publique OT_Auth_PK, par exemple en appliquant à la signature SIG'' reçue un algorithme cryptographique de vérification de signature utilisant la clé publique OT_Auth_PK, puis en comparant le résultat obtenu au défi App_Ch généré à l'étape E50.

Si la signature SIG'' n'est pas correctement vérifiée, le procédé d'authentification a échoué.

Si la signature SIG'' est correctement vérifiée à l'étape E68, le module sécurisé 20 s'est authentifié auprès du processeur 10 (sur lequel s'exécute l'application App). On obtient ainsi une authentification mutuelle du processeur 10 sur lequel s'exécute l'application App et du module sécurisé 20.

Dans certains modes de réalisation dans lesquels une partie du code de l'application App est mémorisée chiffrée dans la mémoire non-volatile 12, on peut procéder, après une telle authentification mutuelle, au déchiffrement de cette

partie mémorisée chiffrée, au moyen de la clé cryptographique AEK dans l'exemple donnée plus haut.

La figure 5 représente schématiquement les éléments principaux d'un second exemple de système dans lequel peut être mise en œuvre l'invention.

Ce système comprend un appareil électronique 100, par exemple un téléphone cellulaire, un serveur de fournisseur d'application 200 et un serveur d'une autorité extérieure 300.

L'appareil électronique comprend un processeur 110 (par exemple un microprocesseur), une mémoire non-volatile réinscriptible 112, une mémoire vive 114, un module de communication 116 et un module sécurisé 120.

La mémoire non-volatile réinscriptible 112 mémorise des applications dont l'exécution par le processeur 110 permet la mise en œuvre de procédés de traitement de données et d'échanges de données, ici notamment avec le module sécurisé 120 et le serveur 200. La mémoire non-volatile réinscriptible 12 mémorise ainsi notamment une application App qui cherche, lors de son exécution par le processeur 10, à s'authentifier auprès du module sécurisé 120 et met notamment en œuvre pour ce faire les étapes décrites ci-dessous en référence à la figure 6.

La mémoire non-volatile réinscriptible 112 mémorise également des données utilisées lors des procédés de traitement mis en œuvre par le processeur 10.

En particulier, la mémoire non-volatile réinscriptible 112 mémorise les données cryptographiques suivantes, utilisées comme décrit ci-dessous dans le cadre du procédé de la figure 6 :

- une clé privée App_SK attribuée à l'application App, mémorisée dans la mémoire non-volatile réinscriptible 112 sous forme chiffrée par une clé cryptographique AEK ;

- des données de dérivation App_Deriv, App_AEK_Deriv et App_AIK_Deriv ;

- une clé publique App_PK associée à la clé privée App_SK (dans une infrastructure à clé publique) et la signature SIG de cette clé publique App_PK au moyen d'une clé privée d'une autorité extérieure OT_SK ;

- une clé publique OT_Auth_PK associée (dans une infrastructure à clé publique) à une clé privée OT_Auth_SK mémorisée dans le module sécurisé 120, comme mentionné plus bas, ainsi qu'un code d'authentification MAC de la clé

publique OT_Auth_PK.

Mise à part la clé publique OT_Auth_PK (associée au module sécurisé 120), ces données sont spécifiques à l'application concernée (ici l'application App).

La paire clé privée App_SK – clé publique App_PK, ainsi que les données de dérivation App_Deriv, App_AEK_Deriv, App_AIK_Deriv, sont préparées et fournies par le fournisseur de l'application App (la signature SIG et la clé publique OT_Auth_PK étant fournies par l'autorité extérieure 300 au fournisseur de l'application App comme schématiquement représenté en pointillés en figure 5).

Au préalable (c'est-à-dire avant l'installation de l'application App au sein de l'appareil électronique 100), le fournisseur de l'application calcule :

- une clé dérivée Prov_App_K par application d'une fonction de dérivation G à une clé racine Prov_Apps (commune à toutes les applications) et à la donnée de dérivation App_Deriv (spécifique à l'application concernée) : Prov_App_K = G(Prov_Apps,App_Deriv) ;

- la clé cryptographique AEK par application d'une fonction de dérivation F à la clé dérivée Prov_App_K et à la donnée de dérivation App_AEK_Deriv (spécifique à l'application concernée) : AEK = F(Prov_App_K,App_AEK_Deriv) ;

- une clé AIK par application de la fonction de dérivation F à la clé dérivée Prov_App_K et à la donnée de dérivation App_AIK_Deriv (spécifique à l'application concernée) : AIK = F(Prov_App_K,App_AIK_Deriv).

Le fournisseur de l'application peut ainsi chiffrer la clé privée App_SK par application d'une fonction cryptographique de chiffrement utilisant la clé cryptographique AEK.

Le fournisseur de l'application obtient par ailleurs de l'autorité extérieure précitée la signature SIG de la clé publique App_PK (la clé publique App_PK étant par exemple transmise à un serveur de l'autorité extérieure, qui calcule la signature SIG au moyen de sa clé privée OT_SK et retourne la signature SIG).

Le fournisseur de l'application obtient également de l'autorité extérieure 300 (comme schématiquement représenté en pointillés en figure 5) la clé publique OT_Auth_PK (associée à la clé privée OT_Auth_SK mémorisée dans le module sécurisé 120) et peut ainsi générer le code d'authentification MAC en appliquant, à la clé publique OT_Auth_PK, un algorithme de génération de code

d'authentification utilisant la clé AIK préalablement calculée comme indiqué ci-dessus.

Toutes ces données peuvent ainsi être associées à l'application App par le fournisseur d'application et inscrites dans la mémoire non-volatile réinscriptible 112 lors de l'installation de l'application App dans cette même mémoire non-volatile réinscriptible 12.

Au cours des procédés mis en œuvre par le processeur 110, et notamment le procédé de la figure 6, la mémoire vive 114 peut mémoriser les données manipulées par ces procédés, notamment comme décrit ci-après.

Le module sécurisé 120 est du même type que le module sécurisé 20 décrit ci-dessus en référence à la figure 1 et ne sera donc pas décrit à nouveau.

Le module sécurisé 120 mémorise, par exemple dans sa mémoire non-volatile réinscriptible, la clé publique OT_PK associée (dans une infrastructure à clé publique) à la clé privée OT_SK de l'autorité extérieure et la clé privée OT_Auth_SK déjà mentionnée, fournies par l'autorité extérieure 300 (comme représenté en pointillés en figure 5).

Le processeur 110 est relié au module sécurisé 120 et au module de communication 116, par exemple au moyen d'un bus, et peut ainsi échanger des données avec ces éléments.

Le module de communication 116 permet la connexion du processeur 110 à un réseau de télécommunication, comprenant par exemple un réseau de téléphonie mobile (par exemple de type GSM) et un réseau public (tel que le réseau Internet, utilisant par exemple un protocole de type IP), sur lequel est également connecté le serveur du fournisseur d'application 200 et le serveur de l'autorité extérieure 300.

Le processeur 110, le serveur du fournisseur d'application 200 et le serveur de l'autorité extérieure 300 peuvent ainsi échanger des données via le module de communication 116, comme schématiquement représenté en figure 5, notamment des données utilisées au cours d'un procédé d'authentification tel que celui présenté ci-dessous en référence à la figure 6. Les messages échangés comme décrits ci-dessous peuvent par exemple être placés au sein de requêtes http ou https ou dans des réponses à de telles requêtes.

Comme visible en figure 5, le serveur 200 mémorise la clé racine Prov_Apps déjà mentionnée (générée par le fournisseur d'application et valable

pour toutes les applications).

La figure 6 représente les étapes principales d'un quatrième exemple de procédé d'authentification d'une application, mise en œuvre ici dans le cadre du système de la figure 5. En particulier, les étapes de ce procédé mises en œuvre par le processeur 110 sont effectuées du fait de l'exécution par le processeur 110 de l'application App.

Le procédé de la figure 6 débute à l'étape E100, à laquelle le processeur 100 envoie la donnée de dérivation App_Deriv à destination du serveur 200.

Le serveur 200 reçoit la donnée de dérivation App_Deriv à l'étape E102 et peut ainsi déterminer à l'étape E108 la clé dérivée Prov_App_K par application de la fonction de dérivation G à la clé racine Prov_Apps (mémorisée dans le serveur 200) et à la donnée de dérivation App_Deriv reçue :

Prov_App_K = G(Prov_Apps,App_Deriv).

Le serveur 200 envoie la clé dérivée Prov_App_K au processeur 110 à l'étape E106.

Le processeur 110 reçoit la clé dérivée Prov_App_K à l'étape E108 et la mémorise (par exemple dans la mémoire vive 114) pour utilisation ultérieure (voir ci-dessous les étapes E122 et E142).

Le processeur 110 émet alors à l'étape E110 la clé publique App_PK et la signature SIG associée, à destination du module sécurisé 120. Comme indiqué pour les procédés des figures 2 et 3, il est envisageable de manière optionnelle d'établir au préalable un canal sécurisé entre le processeur 110 et le module sécurisé 120 et de réaliser alors l'émission de l'étape E110 et les échanges décrits ci-dessous via ce canal sécurisé.

Le module sécurisé 120 reçoit la clé publique App_PK et la signature SIG à l'étape E112.

Le module sécurisé 120 peut ainsi vérifier à l'étape E114 la signature reçue SIG à l'aide de la clé publique OT_PK, comme expliqué ci-dessus pour l'étape E6 dans le cadre de la figure 2.

Si la signature n'est pas correctement vérifiée, le module sécurisé 120 met fin au processus d'authentification, en renvoyant par exemple un message d'erreur au processeur 110.

Si la signature est correctement vérifiée, le procédé de la figure 6 se poursuit à l'étape E116 à laquelle le module sécurisé 120 génère un défi SE_Ch,

par exemple par tirage aléatoire.

Le module sécurisé 120 peut ainsi émettre le défi SE_Ch à destination du processeur 110 à l'étape E118.

Le processeur 110 reçoit le défi SE_Ch à l'étape E120 et détermine à l'étape E122 la clé cryptographique AEK en appliquant la fonction de dérivation F à la clé dérivée Prov_App_K reçue à l'étape E108 et à la donnée de dérivation App_AEK_Deriv mémorisée comme indiqué plus haut dans la mémoire non-volatile 112 : AEK = F(Prov_App_K,App_AEK_Deriv).

Le processeur 110 peut ainsi déchiffrer à l'étape E124 la clé privée App_SK par application à la version chiffrée $[App\_SK]^{AEK}$ (mémorisée dans la mémoire non-volatile 112) d'un algorithme cryptographique de déchiffrement utilisant la clé cryptographique AEK déterminée à l'étape E122. La clé privée App_SK ainsi obtenue est mémorisée dans la mémoire vive 114 au moins jusqu'à la mise en œuvre de l'étape E126 décrite ci-dessous.

Le processeur 110 génère alors à l'étape E126 une signature SIG' par application au défi SE_Ch reçu à l'étape E120 d'un algorithme cryptographique de signature utilisant la clé privée App_SK (déchiffrée à l'étape E124).

Le processeur 110 génère un défi App_Ch, par exemple par tirage aléatoire, à l'étape E128.

Le processeur 110 envoie la signature SIG' et le défi App_Ch au module sécurisé 120 à l'étape E130.

Le module sécurisé 120 reçoit la signature SIG' à l'étape E132 et peut ainsi vérifier la signature SIG' à l'étape E134 en utilisant la clé publique App_PK (dont la signature par l'autorité extérieure a été vérifiée à l'étape E114), par exemple comme décrit ci-dessus à propos de l'étape E24.

Si la signature SIG' n'est pas correctement vérifiée, le procédé d'authentification a échoué. Le module sécurisé 120 peut alors émettre un message d'erreur à destination du processeur 110.

Si la signature SIG' est correctement vérifiée à l'étape E134, le processeur 110 sur lequel s'exécute l'application App s'est authentifié auprès du module sécurisé 120.

Le procédé se poursuit alors en vue de l'authentification du module sécurisé 120 par le processeur 110 sur lequel s'exécute l'application App.

Pour ce faire, le module sécurisé 120 génère à l'étape E136 une

signature SIG'' par application au défi App_Ch reçu à l'étape E132 d'un algorithme cryptographique de signature utilisant la clé privée OT_Auth_SK (mémorisée dans le module sécurisé 120 comme indiqué plus haut).

Le module sécurisé 120 envoie la signature SIG'' ainsi obtenue au processeur 110 à l'étape E138.

Le processeur 110 reçoit la signature SIG'' à l'étape E140 et détermine à l'étape E142 la clé AIK par dérivation à partir de la clé dérivée Prov_App_K (reçue à l'étape E108) et de la donnée de dérivation App_AIK_Deriv (mémorisées dans la mémoire non-volatile 112) en utilisant la fonction de dérivation F :

AIK = F(Prov_App_K,App_AIK_Deriv).

Le processeur 10 peut ainsi vérifier à l'étape E144 que le code d'authentification MAC mémorisé en association avec la clé publique OT_Auth_PK est bien égal au résultat de l'application, à la clé publique OT_Auth_PK mémorisée, de l'algorithme de génération de code d'authentification de message utilisant la clé AIK obtenue à l'étape E142, ce qui permet de s'assurer de l'intégrité de la clé publique OT_Auth_PK mémorisée.

Si cette vérification est un échec, le processeur 110 met fin au procédé d'authentification, en envoyant par exemple un message d'erreur au module sécurisé 120.

Si la vérification de l'étape E144 est correctement effectuée, le processeur 110 vérifie la signature SIG'' à l'étape E146 en utilisant la clé publique OT_Auth_PK, par exemple en appliquant à la signature SIG'' reçue un algorithme cryptographique de vérification de signature utilisant la clé publique OT_Auth_PK, puis en comparant le résultat obtenu au défi App_Ch généré à l'étape E128.

Si la signature SIG'' n'est pas correctement vérifiée, le procédé d'authentification a échoué.

Si la signature SIG'' est correctement vérifiée à l'étape E146, le module sécurisé 120 s'est authentifié auprès du processeur 110 (sur lequel s'exécute l'application App). On obtient ainsi une authentification mutuelle du processeur 110 sur lequel s'exécute l'application App et du module sécurisé 120.

Dans certains modes de réalisation dans lesquels une partie du code de l'application App est mémorisée chiffrée dans la mémoire non-volatile 12, on peut procéder, après une telle authentification mutuelle, au déchiffrement de cette partie mémorisée chiffrée (étape E148), au moyen de la clé cryptographique AEK

dans l'exemple donnée plus haut.

Le processeur 110 peut alors émettre à l'étape E150 un message de confirmation de l'authentification à destination du module sécurisé 120.

Le module sécurisé 120 reçoit ce message de confirmation à l'étape E152 et peut alors émettre un message signé, par exemple au moyen de la clé privée OT_Auth_SK, à destination du serveur de l'autorité extérieure 300 (E154), où ce message peut être reçu et sa signature vérifiée au moyen de la clé publique OT_Auth_PK (étape E156).

On pourrait prévoir en variante (ou éventuellement en complément de ce qui vient d'être décrit) l'émission par le processeur 110 (du fait de l'exécution de l'application App) d'un message signé (par exemple au moyen de la clé privée App_SK) à destination du serveur 200.

REVENDICATIONS

1. Procédé d'authentification d'une application (App) exécutée par un processeur (10 ; 110) d'un appareil électronique (100), caractérisé en ce qu'il comprend les étapes suivantes :

- dérivation (E14 ; E44 ; E122) d'une clé cryptographique (AEK) à partir d'une donnée (OT_Apps ; OT_App_K ; Prov_App_K) reçue d'une entité électronique extérieure (20 ; 200) ;

- déchiffrement (E16 ; E46 ; E124), au moyen de la clé cryptographique dérivée (AEK), d'une clé secrète mémorisée chiffrée ($[App\_SK]^{AEK}$) ;

- émission (E20 ; E52 ; E140) de données d'authentification (SIG') préparées au moyen de la clé secrète déchiffrée (App_SK).

2. Procédé d'authentification selon la revendication 1, dans lequel la clé secrète chiffrée ($[App\_SK]^{AEK}$) est mémorisée dans une mémoire non-volatile réinscriptible (12 ; 112) et dans lequel la clé secrète déchiffrée (App_SK) est mémorisée dans une mémoire vive (14 ; 114).

3. Procédé d'authentification selon la revendication 1 ou 2, comprenant les étapes suivantes :

- réception (E12 ; E42 ; E120) d'un défi (SE_Ch) ;

- préparation (E18 ; E48 ; E126) desdites données d'authentification (SIG') par application au défi reçu (SE_Ch) d'un algorithme cryptographique utilisant la clé secrète déchiffrée (App_SK).

4. Procédé d'authentification selon l'une des revendications 1 à 3, dans lequel ladite donnée reçue (OT_App_K ; Prov_App_K) est préparée par l'entité électronique extérieure (20 ; 200) sur la base d'une donnée (App_Deriv) reçue de l'application (App) et d'une clé (OT_Apps ; Prov_Apps) mémorisée dans l'entité électronique extérieure (20 ; 200).

5. Procédé d'authentification selon l'une des revendications 1 à 4, dans lequel les données d'authentification (SIG') sont émises à destination d'un module sécurisé (20).

6. Procédé d'authentification selon la revendication 5, comprenant une étape d'émission par le module sécurisé (20) d'un message signé au moyen d'une clé privée (OT_Auth_SK) mémorisée dans le module sécurisé (20) à destination d'un serveur (300).

7. Procédé d'authentification selon la revendication 5 ou 6, dans lequel l'entité électronique extérieure est le module sécurisé (20).

8. Procédé d'authentification selon l'une des revendications 1 à 6, dans lequel l'entité électronique extérieure est un serveur (200).

9. Procédé d'authentification selon l'une des revendications 5 à 8, la revendication 8 étant prise dans la dépendance de la revendication 5 ou 6, dans lequel la clé secrète est une clé privée (App_SK), le procédé d'authentification comprenant les étapes suivantes :

- vérification (E6 ; E34 ; E114), par le module sécurisé (20) et au moyen d'une première clé publique (OT_PK) mémorisée dans le module sécurisé (20), d'une signature (SIG) annexée à une seconde clé publique (App_PK) associée à ladite clé privée (App_SK) ;

- vérification (E24 ; E56 ; E134), par le module sécurisé (20), desdites données d'authentification (SIG') au moyen de la seconde clé publique (App_PK).

10. Procédé d'authentification selon l'une des revendications 5 à 9, la revendication 8 étant prise dans la dépendance de la revendication 5 ou 6, comprenant en outre un processus d'authentification (E58, E68 ; E136 ; E146) du module sécurisé (20) auprès de l'application (App).

11. Procédé d'authentification selon la revendication 10, dans lequel le processus d'authentification du module sécurisé auprès de l'application utilise une clé publique (OT_Auth_PK) mémorisée au sein de l'appareil électronique en association avec un code d'authentification de message (MAC) produit par une autorité extérieure (30 ; 300).

12. Procédé d'authentification selon l'une des revendications 5 à 11, la revendication 8 étant prise dans la dépendance de la revendication 5 ou 6, dans lequel le module sécurisé (20) est un élément sécurisé.

13. Procédé d'authentification selon l'une des revendications 1 à 12, dans lequel l'appareil électronique est un téléphone mobile.

14. Procédé d'authentification selon l'une des revendications 1 à 12, dans lequel l'appareil électronique est une tablette numérique.

15. Procédé d'authentification selon l'une des revendications 1 à 12, dans lequel l'appareil électronique comprend un élément de maintien sur une partie du corps de l'utilisateur.

16. Procédé d'authentification selon l'une des revendications 1 à 15,

comprenant une étape d'émission par l'application (App) d'un message signé au moyen de la clé secrète déchiffrée (App_SK) à destination d'un serveur (200).

17. Procédé d'authentification selon l'une des revendications 1 à 16, comprenant une étape de déchiffrement d'une partie d'un code exécutable de l'application au moyen de la clé cryptographique dérivée (AEK).

18. Appareil électronique comprenant un processeur (10 ; 110) et au moins une mémoire (12 ; 112) mémorisant une application (App) exécutable par le processeur (10 ; 110), l'application (App) étant conçue de sorte que le processeur (10 ; 110) mette en œuvre un procédé selon l'une des revendications 1 à 17 lorsque l'application (App) est exécutée par le processeur (10 ; 110).

19. Programme d'ordinateur (App) comprenant des instructions exécutables par un processeur (10 ; 110) d'un appareil électronique et conçues de sorte que le processeur (10 ; 110) mette en œuvre un procédé selon l'une des revendications 1 à 17 lorsque ces instructions sont exécutées par le processeur (10 ; 110).

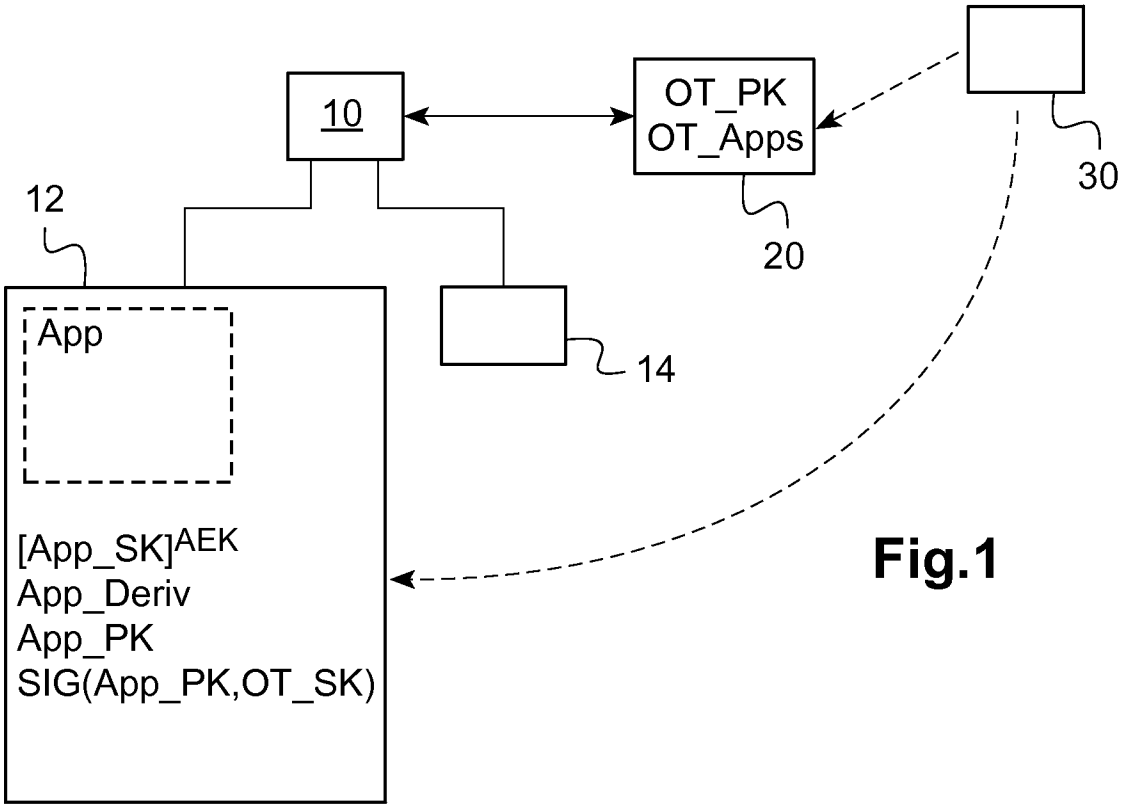

10
OT_PK
OT_Apps
30
20
12
App
14
[App_SK]AEK
App_Deriv
App_PK
SIG(App_PK,OT_SK)

Fig.1

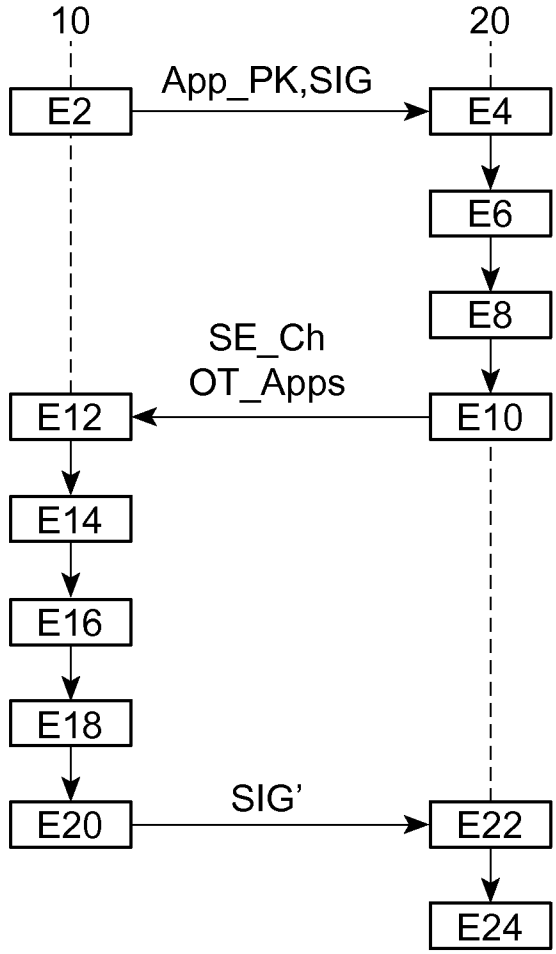

10
20
App_PK,SIG
E2
E4
E6
E8
SE_Ch
OT_Apps
E12
E10
E14
E16
E18
SIG'
E20
E22
E24

Fig.2

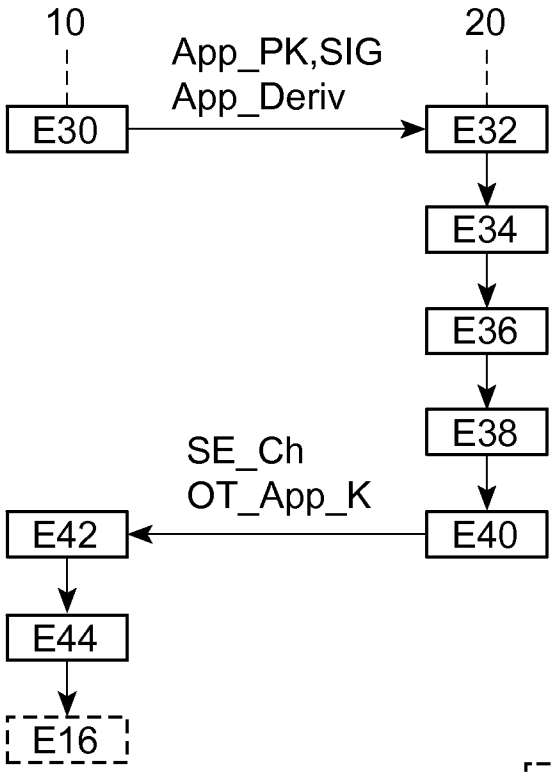

10
20
App_PK,SIG
App_Deriv
E30
E32
E34
E36
E38
SE_Ch
OT_App_K
E42
E40
E44
E16

Fig.3

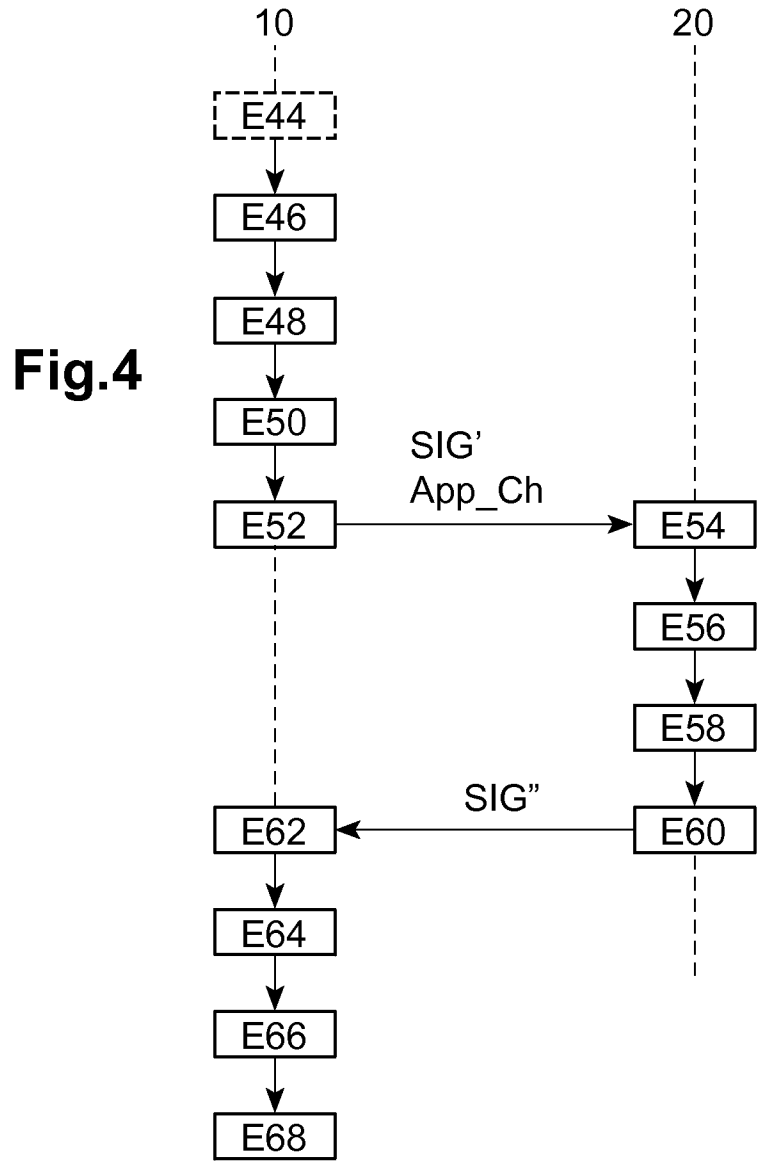

10
20
E44
E46
E48
E50
SIG'
App_Ch
E52
E54
E56
E58
SIG"
E62
E60
E64
E66
E68

Fig.4

Fig.5

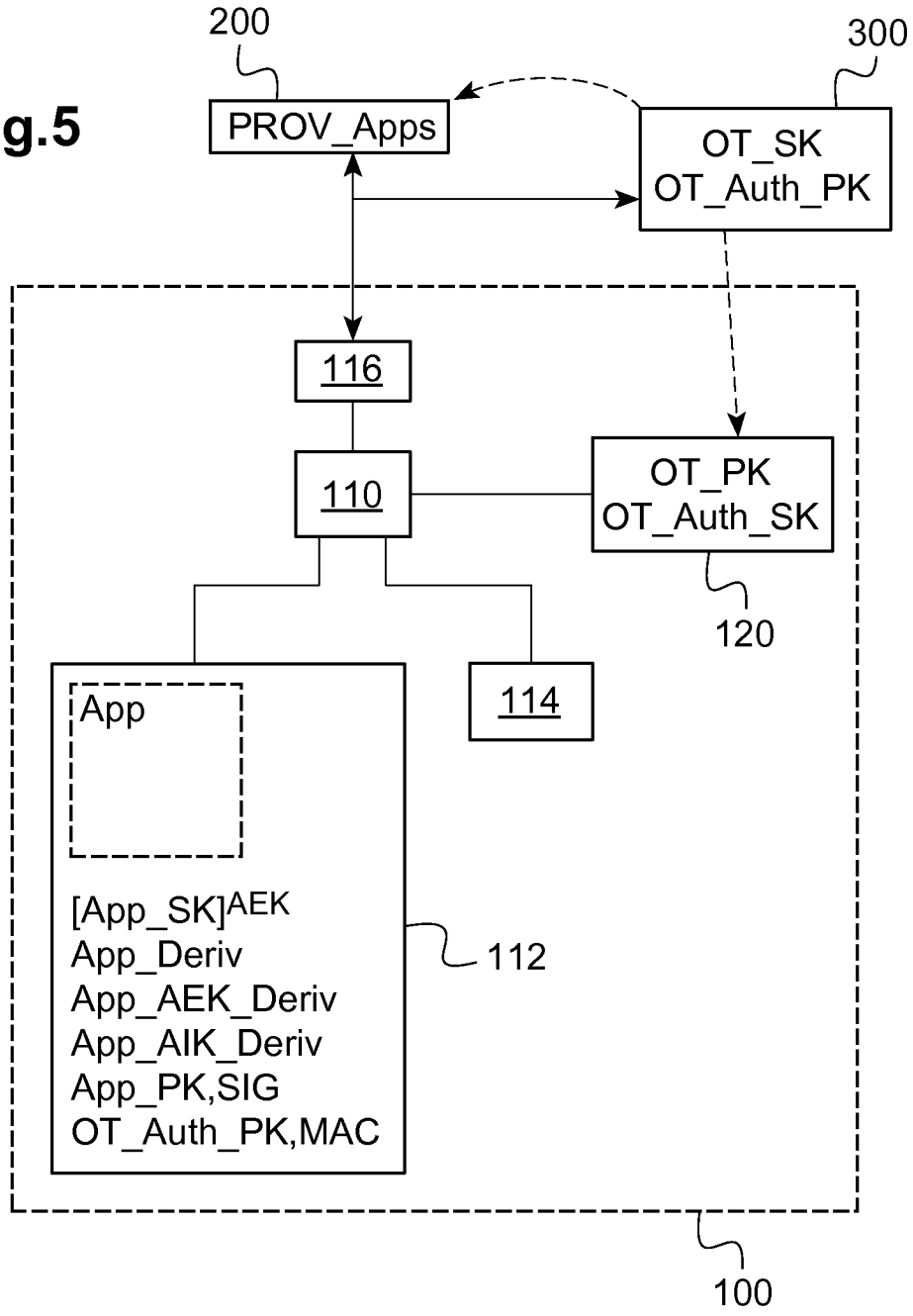
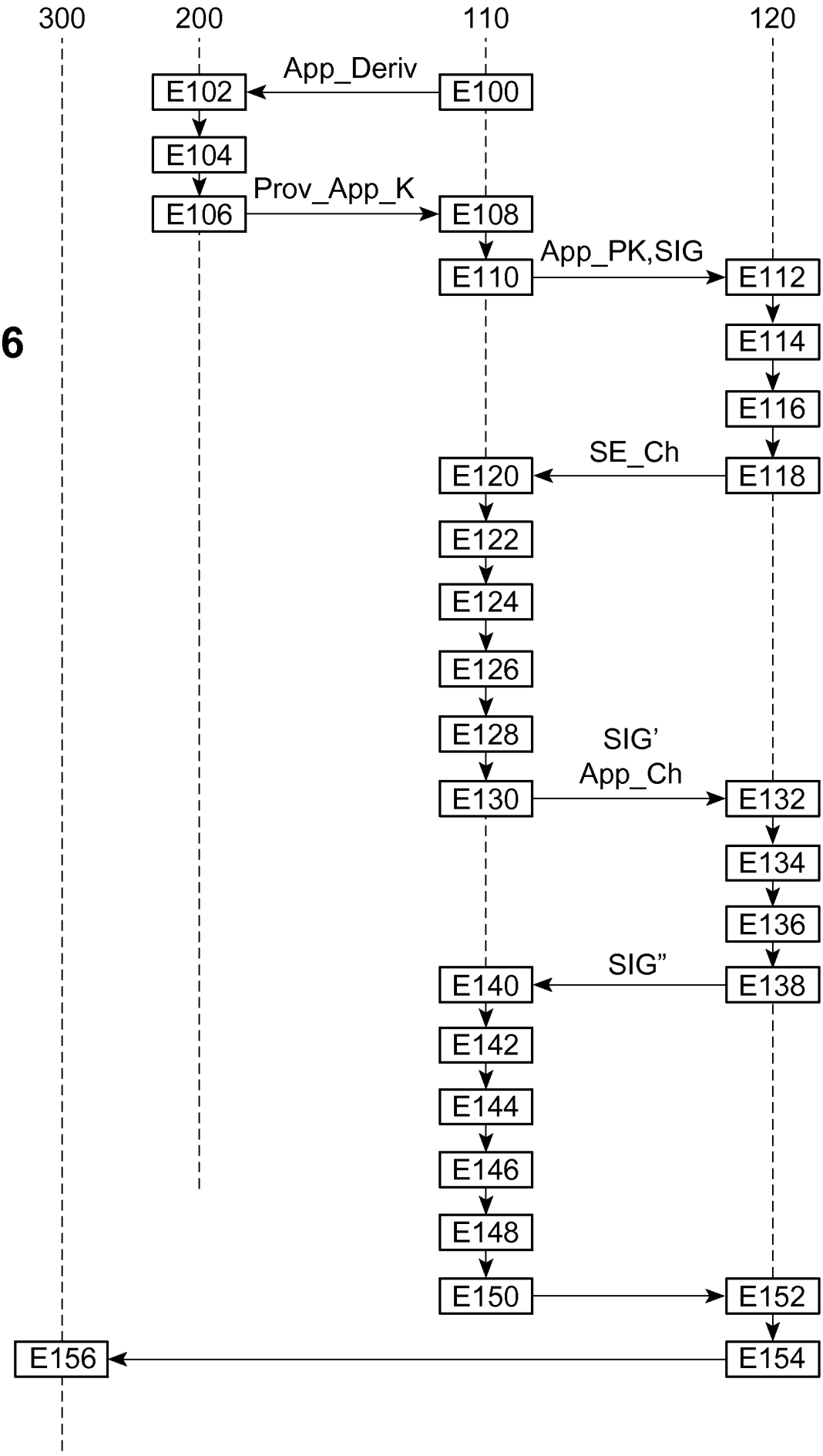

200
300
PROV_Apps
OT_SK
OT_Auth_PK
116
110
OT_PK
OT_Auth_SK
120
114
App
[App_SK]AEK
App_Deriv
App_AEK_Deriv
App_AIK_Deriv
App_PK,SIG
OT_Auth_PK,MAC
112
100

Fig.6


300
200
110
120
App_Deriv
E102
E100
E104
Prov_App_K
E106
E108
App_PK,SIG
E110
E112
E114
E116
SE_Ch
E120
E118
E122
E124
E126
E128
SIG'
App_Ch
E130
E132
E134
E136
SIG"
E140
E138
E142
E144
E146
E148
E150
E152
E156
E154

INTERNATIONAL SEARCH REPORT

International application No
PCT/FR2015/053586

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/44
ADD. G06F21/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2014/139318 A1 (MALPANI AMBARISH [US] ET AL) 22 May 2014 (2014-05-22) paragraphs [0061] - [0065] paragraphs [0023], [0027] - [0028], [0051] figures 1,2A	1-19
X	US 2013/124860 A1 (MAIDL MONIKA [DE] ET AL) 16 May 2013 (2013-05-16) paragraphs [0014] - [0021] paragraphs [0032] - [0045]	1-3,5,8, 12,18,19
Y	US 2011/131401 A1 (SINGH ANAND [US] ET AL) 2 June 2011 (2011-06-02) abstract paragraphs [0017] - [0020], [0022], [0024] paragraphs [0030] - [0033]	1-3,5-19

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 March 2016	24/03/2016

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Segura, Gustavo

2

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	"Encyclopedia of Cryptography and Security", 26 October 2011 (2011-10-26), Springer, XP055223776, ISBN: 978-1-4419-5905-8 pages 679-680, page 679, right-hand column, line 14 - page 680, right-hand column, last line	1-3,5-19
A	US 2006/211490 A1 (FALVEY GRAHAME M [AT] ET AL) 21 September 2006 (2006-09-21) paragraphs [0044] - [0049] paragraphs [0089] - [0107]	1-19

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/FR2015/053586

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014139318 A1	22-05-2014	NONE	
US 2013124860 A1	16-05-2013	CN 102986161 A	20-03-2013
		DE 102010027586 A1	19-01-2012
		EP 2567501 A1	13-03-2013
		US 2013124860 A1	16-05-2013
		WO 2012010380 A1	26-01-2012
US 2011131401 A1	02-06-2011	US 2011131401 A1	02-06-2011
		US 2013297925 A1	07-11-2013
US 2006211490 A1	21-09-2006	AU 2006201105 A1	05-10-2006
		CA 2533520 A1	17-09-2006
		EP 1703478 A2	20-09-2006
		RU 2310907 C1	20-11-2007
		US 2006211490 A1	21-09-2006

RAPPORT DE RECHERCHE INTERNATIONALE

Demande internationale n°
PCT/FR2015/053586

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
INV. G06F21/44
ADD. G06F21/12

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
G06F

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si cela est réalisable, termes de recherche utilisés)
EPO-Internal

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	US 2014/139318 A1 (MALPANI AMBARISH [US] ET AL) 22 mai 2014 (2014-05-22) alinéas [0061] - [0065] alinéas [0023], [0027] - [0028], [0051] figures 1,2A -----	1-19
X	US 2013/124860 A1 (MAIDL MONIKA [DE] ET AL) 16 mai 2013 (2013-05-16) alinéas [0014] - [0021] alinéas [0032] - [0045] -----	1-3,5,8, 12,18,19
Y	US 2011/131401 A1 (SINGH ANAND [US] ET AL) 2 juin 2011 (2011-06-02) abrégé alinéas [0017] - [0020], [0022], [0024] alinéas [0030] - [0033] -----	1-3,5-19

-/--

[X] Voir la suite du cadre C pour la fin de la liste des documents

[X] Les documents de familles de brevets sont indiqués en annexe

* Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
18 mars 2016	24/03/2016

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Segura, Gustavo

2

C(suite). DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie*	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
Y	"Encyclopedia of Cryptography and Security", 26 octobre 2011 (2011-10-26), Springer, XP055223776, ISBN: 978-1-4419-5905-8 pages 679-680, page 679, colonne de droite, ligne 14 - page 680, colonne de droite, dernière ligne	1-3,5-19
A	US 2006/211490 A1 (FALVEY GRAHAME M [AT] ET AL) 21 septembre 2006 (2006-09-21) alinéas [0044] - [0049] alinéas [0089] - [0107]	1-19

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Demande internationale n°
PCT/FR2015/053586

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2014139318 A1	22-05-2014	AUCUN	
US 2013124860 A1	16-05-2013	CN 102986161 A	20-03-2013
		DE 102010027586 A1	19-01-2012
		EP 2567501 A1	13-03-2013
		US 2013124860 A1	16-05-2013
		WO 2012010380 A1	26-01-2012
US 2011131401 A1	02-06-2011	US 2011131401 A1	02-06-2011
		US 2013297925 A1	07-11-2013
US 2006211490 A1	21-09-2006	AU 2006201105 A1	05-10-2006
		CA 2533520 A1	17-09-2006
		EP 1703478 A2	20-09-2006
		RU 2310907 C1	20-11-2007
		US 2006211490 A1	21-09-2006