PROCEDE ET SYSTEME PERMETTANT DE TESTER UNE INTEGRITE CRYPTOGRAPHIQUE D'UNE DONNEE TOLERANTE AUX ERREURS

L'objet de la présente invention concerne un procédé et un système permettant de tester une intégrité de données tolérante aux erreurs.

Elle trouve notamment son application dans le domaine des télécommunications pour tester l'intégrité des messages, dans le domaine radio, dans le domaine de la cryptographie de manière plus générale pour tester l'intégrité de données.

Les informations transitant par les canaux radios sont le plus souvent réceptionnées avec des erreurs.

Du point de vue des télécommunications ce problème est résolu avec application d'un code correcteur qui permet de corriger les erreurs.

Ce que le code correcteur va considérer comme des erreurs peut résulter :

- d'erreurs de transmissions physiques,
- provenir de modifications réalisées intentionnellement par un attaquant dans le but de leurrer le récepteur.

La solution de sécurité maximale classique consiste à forger une empreinte du message émis et à transmettre cette empreinte conjointement au message. A la réception, une empreinte du message reçu est calculée par le récepteur et comparée avec l'empreinte reçue. Si un seul bit de ces deux empreintes diffère alors le message est rejeté car potentiellement modifié par un attaquant. En effet la couche télécom de correction des erreurs a déjà été appliquée et a corrigé les erreurs de transmission.

Cette solution de sécurité peut pour certaines applications être maximaliste. En effet, certaines applications peuvent ne pas être altérées de façon significative par un nombre minime (fixé) de bits modifiés et cela peut ne pas avoir d'impact sur la sécurité du système.

Le problème est donc de garantir la sécurité tout en permettant éventuellement que certaines applications puissent être altérées par des erreurs.

Ainsi en fonction des applications, la politique de sécurité du système sera stricte et ne tolèrera aucune erreur ou au contraire permettra un nombre pré-défini d'erreurs.

Une approche connue de l'art antérieur consiste à diviser les codes d'authentification de message ou en abrégé anglo-saxon MACs (en plusieurs blocs mais cela ne résout pas le problème. Cette méthode permet de rejeter des messages de taille inférieure et de plus la taille des MAC étant plus petite la probabilité d'erreurs est plus faible

L'article de Ch .G. Boncelet , intitulé « The NTMAC for authentication of noisy messages » IEEE Transaction on Information Forensics and Security vol.1 n1. March 2006 est une illustration de ce type d'approche

Malgré tous les avantages offerts par la mise en œuvre des systèmes de l'art antérieur, ces derniers présentent notamment les inconvénients suivants :

- les solutions antérieures ne permettent pas la correction des erreurs transmises,
- la couche de correction d'erreurs est appliquée en amont pour des motivations « TELECOM »,
- Les solutions antérieurs essayent de s'approcher d'une solution en diminuant la taille des MACs, sans toutefois résoudre le problème posé.

L'un des objectifs de la présente demande de brevet est de définir une fonction de type code d'authentification de message ou en anglo-saxon MAC (Message Authentification Code) pouvant tolérer des erreurs dans la transmission (ou la conservation) des messages. Plus exactement, la procédure de vérification ne fournit pas une décision OK/NOK mais fournit une note qui indique la proximité des données reçues avec le message valide,

tout en conservant la sécurité apportée par la fonction de protection cryptographique.

L'invention concerne un procédé pour tester l'intégrité cryptographique de données m caractérisé en ce qu'il comporte en combinaison au moins les étapes suivantes :

une première étape au cours de laquelle on génère une clé (C, Ksym) en exécutant les étapes suivantes :

- soit F(GF(p), n, M, d) une famille de codes correcteurs d'erreurs systématiques linéaires ou non linéaires sur GF(p) de longueur n, de cardinal M et distance d,
- tirer de manière aléatoire un code correcteur C de la famille F(GF(p), n,k,d) ayant une capacité de correction t, et définir la clé Ksym,

une deuxième étape comprenant les étapes suivantes :

- coder (20) lesdites données m en utilisant le code correcteur sélectionné lors de la première étape,
- pour un code systématique, calculer la redondance r, et utiliser la clé secrète Sk(C, Ksym, l) avec l un entier non nul inférieur à t qui représente le nombre d'erreurs accepté afin de protéger (21) la redondance r, on obtient la redondance protégée rp= SYM_{ksym} (r), avec SYM_{ksym} = SYM, un algorithme bloc de chiffrement symétrique et Ksym une clé secrète, transmettre (22) les données (m|rp) résultant de la deuxième étape, le message avec la redondance chiffrée,

une troisième étape comprenant les étapes suivantes :

pour un code systématique, soit (ma | ra) les données en entrée de cette fonction, inverser le mécanisme de protection en confidentialité de la redondance à l'aide de l'inverse de l'algorithme de chiffrement symétrique et de la clef secrète $SYM_{ksym}^{-1}(ra)$, (30), ra la redondance chiffrée reçue au niveau de l'étape de vérification d'intégrité,

- soit D_1 une fonction de décodage associée au code C choisi qui a en entrée une donnée quelconque et qui a en sortie le mot du code le

plus proche en termes de distance, et une information décodage réussie OK ou décodage non réussi,

- D_1 est appliquée sur l'ensemble constitué du message reçu ma concaténé avec l'inverse de la redondance $SYM_{ksym}^{-1}(ra)$:

  $D_1(ma|SYM_{ksym}^{-1}(ra))$,

- si le décodage n'est pas possible, il n'existe pas de mot du code à une distance suffisamment faible, (32), les données sont considérées comme non intègres,
- sinon, si le décodage est possible :
- si $d(h_{n-k}(D_1(ma \mid SYM_{ksym}^{-1}(ra)), ma) < I+1$, alors la réponse est OK avec comme message utile effectif reçu $h_{n-k}(D_1(ma \mid SYM_{ksym}^{-1}(ra))$ et l'information sur la distance $d(h_{n-k}(D_1(ma \mid SYM_{ksym}^{-1}(ra)), ma)$,
- sinon les données sont considérées comme non intègres d() est la distance au sens du nombre d'éléments de GF(p) différents, et h_{n-k} est la fonction qui prend une chaîne de caractère en entrée et lui associe les n-k premiers caractères de la chaîne.

Dans le cas d'un code systématique, le procédé comporte, par exemple, une étape de chiffrement des données chiffrant l'ensemble du message utile et de la redondance r au niveau de l'émission et une étape de déchiffrement desdites données au niveau de la réception.

Les données traitées peuvent être des messages utilisés dans des systèmes de transmission.

Selon une variante de réalisation, la redondance r est protégée en confidentialité grâce à un chiffrement par bloc ou un xor avec un Générateur de Pseudo-Aléa.

L'invention concerne aussi un système pour tester l'intégrité cryptographique de données m comprenant au moins les éléments suivants un module émetteur d'un message M, ledit module comprenant une mémoire permettant de stocker les paramètres utilisés pour exécuter les étapes du

procédé selon l'invention, tels que la clé, les données publiques, un médium de transmission, un module récepteur comprenant aussi des moyens de stockage d'au moins les mêmes paramètres qu'à l'émission. Le système peut comprendre des moyens de stockage de données confidentielles telles que les clés secrètes, un processeur adapté à exécuter les étapes du procédé selon l'invention.

D'autres caractéristiques et avantages de la présente invention apparaîtront mieux à la lecture de la description qui suit d'un exemple de réalisation donné à titre illustratif et nullement limitatif annexé des figures qui représentent :

- La figure 1, un exemple de système permettant la mise en œuvre du procédé selon l'invention dans le domaine des télécommunications,
- La figure 2, un exemple de configuration de l'équipement,
- La figure 3, un exemple d'étapes mises en œuvre à l'émission d'un message,
- La figure 4, un exemple d'étapes du procédé selon l'invention mises en œuvre à la réception d'une trame.

Afin de mieux faire comprendre le principe mis en œuvre dans le procédé selon l'invention, la description qui suit est donnée dans un contexte d'émission-réception d'un message dans un système de télécommunications. Le procédé selon l'invention est valable pour un code systématique linéaire ou non linéaire.

La figure 1 schématise un exemple d'architecture d'un système comprenant un module 1 émetteur d'un message M, ledit module comprenant une mémoire 2 permettant de stocker les paramètres utilisés pour exécuter les étapes du procédé selon l'invention, tels que la clé, les données publiques, qui seront explicités ci-après, un médium 3 de transmission, un module récepteur 4 comprenant aussi des moyens de stockage 5 d'au moins les mêmes paramètres qu'à l'émission et un processeur 7, 8 adapté à exécu-

ter les étapes mises en œuvre par le procédé selon l'invention. Le système peut comprendre des moyens de stockage 6 de données confidentielles telles que les clés secrètes.

La solution mise en œuvre par le procédé consiste, notamment, à utiliser de façon non-standard des codes correcteurs d'erreurs ou plus précisément une famille de codes correcteurs d'erreurs.

Le code utilisé à un instant, parmi la famille de codes, est considéré comme une partie de la clef secrète. Considérons tout d'abord le cas des codes systématiques, ce qui englobe notamment les codes linéaires. Le résultat du codage est donc la concaténation du message et d'une redondance. La redondance r est ensuite protégée en confidentialité grâce à un chiffrement par bloc ou un xor avec un Générateur de Pseudo-Aléa.

Dans un second temps, considérons le cas de codes quelconques, dans ce cas, l'ensemble du résultat du codage est protégé en confidentialité.

La description qui suit est donnée en relation avec les figures 2, 3 et 4.

Notations

Soit p un nombre premier, GF(p) le corps à p éléments.

Soient n, k, d des entiers naturels non nuls, C(GF(p),n,k,d) un code correcteur d'erreurs sur GF(p) de longueur n, dimension k et distance d. Soit Fam(GF(p),n,k,d) une famille de code correcteur d'erreurs sur GF(p) de longueur n, dimension k et distance d.

Nous noterons x| y la concaténation de la chaîne de caractères x avec la chaîne y. Le sigle | exprime l'opération de concaténation. Ces paramètres concernent la donnée dont on souhaite tester l'intégrité, ou le message dont l'intégrité va être testée par le procédé selon l'invention.

Définition

Soit un message m de longueur n-k, le procédé selon l'invention va mettre en œuvre au moins les 3 étapes suivantes, ce schéma permettant de construire un code d'authentification de message avec correction d'erreur

ou en anglo-saxon Message Authentification Code with Error Correcting Code (MACECC).

Première étape explicitée avec la figure 2

La première étape Gen 10, figure 2, consiste à générer une clé (C, Ksym) qui va être utilisée pour chiffrer la partie supplémentaire ou redondance r produite par application du code correcteur choisi, la partie supplémentaire ou supM étant utilisée pour corriger les erreurs.

Soit F(GF(p),n,k,d) une famille de code correcteur d'erreurs sur GF(p) de longueur n, dimension k et distance d (une composante de l'espace des clefs).

Le procédé exécute un tirage aléatoire 11 d'un code correcteur d'erreurs C de la famille F(GF(p),n,k,d) de capacité de correction t, le tirage aléatoire étant effectué selon une méthode connue de l'Homme du métier.

Soit l un entier non nul inférieur à t ; l représente le nombre d'erreurs accepté pour la partie utile ou partie supplémentaire ou de manière plus générale pour l'ensemble du message. L'étape suivante est de définir la clé symétrique Ksym.

Un algorithme bloc symétrique SYM et sa clef secrète Ksym sont utilisés pour chiffrer la partie supplémentaire pour chiffrer la redondance ou encore pour chiffrer l'ensemble des données ou du message.

Les paramètres publics sont (F(GF(p),n,k,d), SYM). Ces paramètres sont mémorisés dans l'espace mémoire 5.

La clef secrète est : Sk(C, Ksym, l) ; elle est, par exemple stockée dans une zone mémoire 6 protégée selon des moyens connus de l'Homme du métier.

Deuxième étape création du MAC explicité avec la figure 3

Lors de la deuxième étape, le procédé va encoder 20 ou coder le message ou les données m en utilisant le code correcteur défini à la première étape.

Considérons dans un premier temps le cas d'un code systématique (message + redondance). Le résultat du codage est donc la concaténation de m avec une redondance r, notée m|r.

Au niveau de l'émission, le procédé comporte une étape de protection en confidentialité (seule) de la redondance r en utilisant la clé secrète Sk(C, Ksym, I). La redondance protégée est notée rp= SYM_{ksym} (r), 21.

La protection de la redondance en confidentialité est assurée par un mécanisme de chiffrement. Pour tolérer des erreurs aussi au niveau de la redondance, il est préférable d'utiliser un mode de chiffrement ne propageant pas les erreurs. Dans le cas contraire, l'invention fonctionne aussi, mais les erreurs tolérées sont limitées aux erreurs sur le message. Un mode de chiffrement ne propageant pas les erreurs est typiquement réalisé par un xor avec un pseudo-aléa généré, par exemple avec un algorithme bloc en mode compteur. La redondance protégée rp est appelée MACECC.

Dans le cas d'une application de transmission où le message lui-même doit être protégé en confidentialité, pour tirer profit de l'invention, il faudra utiliser un mode de chiffrement du message m ne propageant pas les erreurs.

Le message m|rp (m concaténé avec rp) est ensuite transmis dans cet exemple au récepteur ou de manière plus générale à une étape de traitement des données.

Troisième étape explicitée en relation avec la figure 4

A la réception du message, 30 le procédé exécute une première étape de vérification Verify(messagetransmis,Sk) comprenant les étapes décrites ci-après.

Les données reçues *messagetransmis* sont considérées comme la concaténation d'un message ma et d'une redondance ra (ma|ra). Cette vérification est, par exemple, réalisée de la manière suivante au niveau du récepteur. Le mécanisme de protection en confidentialité de la redondance

est inversé à l'aide de la clef secrète $SYM_{ksym}^{-1}(ra)$,30, on retrouve r, ra la redondance chiffrée reçue au niveau de l'étape de vérification d'intégrité

Soit D1 une fonction de décodage qui a en entrée une donnée quelconque et qui a en sortie le mot du code le plus proche en termes de distance (ce qui inclut la redondance) et en sortie une information décodage réussie ou décodage non réussi. D1 est appliquée sur l'ensemble du message reçu ma, concaténé avec l'inverse de la redondance $SYM_{ksym}^{-1}(ra)$:

$D_1(ma| SYM_{ksym}^{-1}(ra))$.

Si le décodage n'est pas possible (il n'existe pas de mot unique du code à une distance suffisamment faible) 32, la réponse est NOK, les données ne sont pas considérées comme intègres.

Sinon, si le décodage est possible :

Si $d(h_{n-k}(D_1(ma \mid SYM_{ksym}^{-1}(ra)), ma) < l+1$, alors la réponse est OK avec comme message utile effectif reçu $h_{n-k}(D_1(ma \mid SYM_{ksym}^{-1}(ra))$; sinon la réponse est NOK, les données sont considérées comme non intègres. d() est la distance au sens du nombre d'éléments de GF(p) différents, et h_{n-k} est la fonction qui prend une chaîne de caractère en entrée et lui associe les n-k premiers caractères de la chaîne.

Une solution alternative consiste à opérer le test non pas sur la différence entre les parties utiles des messages, mais sur l'ensemble incluant la redondance. La condition $d(h_{n-k}(D_1(ma \mid SYM_{ksym}^{-1}(ra)), ma) < l+1$, devient alors : $d(D_1(ma \mid SYM_{ksym}^{-1}(ra)), (ma| SYM_{ksym}^{-1}(ra)) < l+1$

Les étapes décrites précédemment peuvent aussi s'appliquer pour des données autres que les messages.

Dans le cas où le code n'est pas systématique, le procédé est modifié de la manière décrite ci-après.

Codes quelconques : un schéma d'intégrité tolérant aux erreurs

Notations et définitions

Soit p un nombre premier, $GF(p)$ le corps à p éléments

Soit d_H la distance de Hamming.

Soit f une application, avec $Im(f)$ son image.

Soient n, d deux entiers strictement positifs. Soit M un ensemble de vecteurs de longueur n définis sur $GF(p)$. Si pour tout couple d'éléments distincts de M, leur distance de Hamming est supérieure ou égale à d et que d est le plus grand entier vérifiant cette propriété alors l'ensemble des vecteurs de M est un code noté $C(GF(p),n,M,d)$.

Soit C un élément du code $C(GF(p),n,M,d)$. A C est associé un encodeur injectif défini de $GF(p)^{n-k}$ vers C et noté Enc_C. Ainsi la fonction inverse notée Enc_C^{-1} est définie de $Im(Enc_C)$ vers $GF(p)^{n-k}$.

Soit C un $C(GF(p),n,M=p_k,d)$ code (linéaire ou non).

Soit π une permutation de $GF(p)^n$ et $p_{n,k}$ la projection de $GF(p)^n$ sur $GF(p)k$ telle que $p_{n,k}(u_1, \ldots, u_n) = (u_1, \ldots, u_k)$. Soit un encodeur Enc_C associé au code $_C$, s'il existe π_C et $p_{n,k}$ tel que pour $m \in GF(p)^{n-k}$, $p_{n,k}(\ _C(Enc_C(m))) = m$, alors l'encodeur est dit systématique. Le code sera dit systématique ou k-systématique.

Les codes de Reed-Solomon sont des exemples de codes pour lesquels il existe un encodeur non systématique.

Soit $F(GF(p),n,M,d,k)$ une famille de code correcteur d'erreurs sur $GF(p)$ de longueur n, de cardinal M, de distance d et de redondance de longueur k. Si $M = p^k$, cette famille peut contenir des codes linéaires et non linéaires.

Soit $C \in F(GF(p),n,M,d,k)$, $m \in GF(p)^{n-k}$, alors nous noterons $Enc_C(m) \in GF(p)^n$ le mot de code de C associé à m. L'opération de décodage d'un mot $m' \in GF(p)^n$ sera notée $Dec_c(m')$ si cette opération admet une solution unique et alors $Dec_c(m') \in C$.

x | y est la concaténation de la chaîne de caractères x avec la chaîne y.

Soit f une fonction inversible, son inverse est notée f^{-1}. Sym_{ksym} est un algorithme de chiffrement symétrique agissant sur des mots de longueur n avec la clef secrète *ksym*. L'opération de chiffrement du message $m \in GF(p)^n$ avec cet algorithme sera notée $Sym_{ksym}(m)$. L'opération de déchiffrement d'un mot $m' \in GF(p)^n$ avec cet algorithme sera notée : $Sym^{-1}_{ksym}(m')$.

Définition d'un schéma d'intégrité tolérant aux erreurs

Soit un message m de longueur $n - k$. Nous appellerons schéma d'intégrité tolérant aux erreurs le schéma défini par les 3 procédés (*Gen*, MAC^3, et *Verify*) suivant. Ce schéma permet de construire un motif d'intégrité tolérant aux erreurs.

Gen

Soit $F(GF(p),n,M,d,k)$ une famille de code correcteur d'erreurs sur $GF(p)$ de longueur n, de cardinal M, de distance d et de redondance k. t est sa capacité de correction.

Soit $0 < l \leq t$, cet entier l représentant le nombre d'erreurs accepté pour les applications. Un code C est tiré aléatoirement dans cette famille de code.

Soit Sym_{ksym} un algorithme de chiffrement. Cet algorithme de chiffrement peut être un algorithme de chiffrement de type bloc ou un algorithme de type GPA. Soit donc *ksym* la clef secrète utilisée pour cet algorithme. Les paramètres publics sont $\{F(GF(p),n,M,d,k), n, M, d, K), Sym, l\}$.

La clef secrète notée S_k est la suivante : $S_k = (C, ksym)$.

MAC

Les différentes étapes sont les suivantes.

1) Encodage du message m avec le code correcteur C : $Enc_C(m) \in GF(p)^n$

2) Protection en confidentialité $Sym_{ksym}(Enc_C(m))$.

Le résultat est donc le MAC^3 résultant $MAC^3(m) = Sym_{ksym}(Enc_C(m))$

Verify

Soit m_{recu} le message reçu ($m_{recu} \in GF(p)^n$)

Les étapes suivantes sont réalisées.

1) Le mécanisme de protection en confidentialité est inversé $Sym^{-1}_{ksym}(m_{recu})$.

2) Le résultat est décodé $Dec_C(Sym^{-1}_{ksym}(m_{recu}))$.

3) La réponse est définie de la manière suivante :

- si le décodage aboutit et fournit un élément du code valide $Enc_C^{-1}(Dec_C((Sym^{-1}_{ksym}(m_{recu}))$, alors, soit d la distance $d = d_H(Sym^{-1}_{ksym}(m_{recu}), Dec(Sym^{-1}_{ksym}(m_{recu})))$:
  - Si $d \leq t$, alors l'information retournée est :

    (OK, d, $Enc_C^{-1}(Dec_C((Sym^{-1}_{ksym}(m_{recu}))$
  - Sinon, la réponse est NOK,
- sinon la réponse est NOK.

La famille de code choisie devra posséder une bonne capacité de correction et être efficacement décodable afin que la solution soit efficace.

Le code correcteur d'erreur a une capacité de correction notée t, cette capacité de correction est supérieure à la tolérance désirée pour la transmission du message.

Le code permet si le nombre d'erreur est inférieur à t de retrouver le message transmis. Le procédé de manière plus générale est efficace, sécuritaire, et peut être paramétré.

REVENDICATIONS

1 – Procédé pour tester l'intégrité cryptographique de données m caractérisé en ce qu'il comporte en combinaison au moins les étapes suivantes :
une première étape au cours de laquelle on génère une clé (C, Ksym) en exécutant les étapes suivantes :

- soit F(GF(p), n, M, d) une famille de codes correcteurs d'erreurs systématiques linéaires ou non linéaires sur GF(p) de longueur n, de cardinal M et distance d,
- tirer de manière aléatoire un code correcteur C de la famille F(GF(p), n,k,d) ayant une capacité de correction t, et définir la clé Ksym,

une deuxième étape comprenant les étapes suivantes :

- coder (20) lesdites données m en utilisant le code correcteur sélectionné lors de la première étape,
- pour un code systématique, calculer la redondance r, et utiliser la clé secrète Sk(C, Ksym, l) avec l un entier non nul inférieur à t qui représente le nombre d'erreurs accepté afin de protéger (21) la redondance r, on obtient la redondance protégée rp= $^{SYM_{ksym}}$ (r), avec $^{SYM_{ksym}}$ = SYM, un algorithme bloc de chiffrement symétrique et Ksym une clé secrète, transmettre (22) les données (m|rp) résultant de la deuxième étape, le message avec la redondance chiffrée,

une troisième étape comprenant les étapes suivantes :
pour un code systématique, soit (ma | ra) les données en entrée de cette fonction, inverser le mécanisme de protection en confidentialité de la redondance à l'aide de l'inverse de l'algorithme de chiffrement symétrique et de la clef secrète $SYM_{ksym}^{-1}(ra)$, (30), ra la redondance chiffrée reçue au niveau de l'étape de vérification d'intégrité,

- soit D_1 une fonction de décodage associée au code C choisi qui a en entrée une donnée quelconque et qui a en sortie le mot du code le

plus proche en termes de distance, et une information décodage réussie OK ou décodage non réussi,

- D_1 est appliquée sur l'ensemble constitué du message reçu ma concaténé avec l'inverse de la redondance $SYM_{ksym}^{-1}(ra)$:

  $D_1(ma|SYM_{ksym}^{-1}(ra))$,

- si le décodage n'est pas possible, il n'existe pas de mot du code à une distance suffisamment faible, (32), les données sont considérées comme non intègres,
- sinon, si le décodage est possible :
- si $d(h_{n-k}(D_1(ma \mid SYM_{ksym}^{-1}(ra)), ma) < I+1$, alors la réponse est OK avec comme message utile effectif reçu $h_{n-k}(D_1(ma \mid SYM_{ksym}^{-1}(ra))$ et l'information sur la distance $d(h_{n-k}(D_1(ma \mid SYM_{ksym}^{-1}(ra)), ma)$,
- sinon les données sont considérées comme non intègres d() est la distance au sens du nombre d'éléments de GF(p) différents, et h_{n-k} est la fonction qui prend une chaîne de caractère en entrée et lui associe les n-k premiers caractères de la chaîne.

2 – Procédé selon la revendication 1 caractérisé en ce que dans le cas d'un code systématique, le procédé comporte une étape de chiffrement chiffrant l'ensemble du message utile et de la redondance au niveau de l'émission et une étape de déchiffrement desdites données.au niveau de la réception.

3 – Procédé selon l'une des revendications précédentes caractérisé en ce que les données traitées sont des messages utilisés dans des systèmes de transmission.

4 – Procédé selon la revendication 1 caractérisé en ce que la redondance r est protégée en confidentialité grâce à un chiffrement par bloc ou un xor avec un Générateur de Pseudo-Aléa.

5 – Système pour tester l'intégrité cryptographique de données m comprenant au moins les éléments suivants :

- un module (1) émetteur d'un message M, ledit module comprenant une mémoire (2) permettant de stocker les paramètres utilisés pour exécuter les étapes du procédé selon l'invention, tels que les données publiques, un médium (3) de transmission,
- un module récepteur (4) comprenant aussi des moyens de stockage (5) d'au moins les mêmes paramètres qu'à l'émission,
- des moyens de stockage (6) de données confidentielles telles que les clés secrètes, et
- un processeur (7) adapté à exécuter les étapes du procédé selon l'une des revendications 1 à 4.