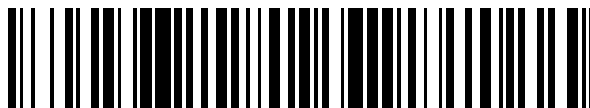


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 717 548**

21 Número de solicitud: 201700755

51 Int. Cl.:

H04L 9/08

(2006.01)

12

PATENTE DE INVENCION CON EXAMEN

B2

22 Fecha de presentación:

08.11.2017

43 Fecha de publicación de la solicitud:

21.06.2019

Fecha de modificación de las reivindicaciones:

06.07.2020

Fecha de concesión:

19.11.2020

45 Fecha de publicación de la concesión:

26.11.2020

73 Titular/es:

**UNIVERSIDADE DE VIGO (100.0%)
Campus Universitario de Vigo s/n
36310 Pontevedra (Pontevedra) ES**

72 Inventor/es:

**MARCOS CURTY, Alonso y
HOI-KWONG, Lo**

54 Título: **Acuerdo seguro de clave con dispositivos no confiables**

57 Resumen:

Los métodos tradicionales de generación de claves criptográficas en una red ruidosa suelen asumir que los dispositivos son confiables, y por lo tanto son vulnerables frente a numerosos ataques, incluidos ataques basados en canales encubiertos. La presente invención se diferencia de los métodos anteriores de generación de claves en el sentido de que presenta un mecanismo que permite la generación segura de claves criptográficas con dispositivos no confiables en una red ruidosa con una estructura de acceso prescrita.

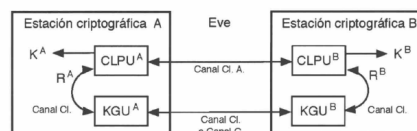


FIGURA 1

Aviso: Se puede realizar consulta prevista por el art. 41 LP 24/2015.
Dentro de los seis meses siguientes a la publicación de la concesión en el Boletín Oficial de la Propiedad Industrial cualquier persona podrá oponerse a la concesión. La oposición deberá dirigirse a la OEPM en escrito motivado y previo pago de la tasa correspondiente (art. 43 LP 24/2015).

ES 2 717 548 B2

DESCRIPCIÓN**ACUERDO SEGURO DE CLAVE CON DISPOSITIVOS NO CONFIABLES**

5

SECTOR DE LA TÉCNICA**DESCRIPCIÓN**10 **CAMPO DE LA INVENCION**

Esta invención se refiere a un método y a un sistema seguros para la generación de una clave criptográfica aleatoria en una red ruidosa.

ESTADO DE LA TÉCNICA

- 15 La criptografía es el arte de hacer códigos. En la criptografía, la seguridad de un protocolo depende a menudo del secreto de una clave criptográfica. Una clave criptográfica suele ser una secuencia aleatoria de números. Si dos partes, Alice y Bob, comparten una clave criptográfica secreta común, K, entonces pueden lograr tanto comunicaciones como autenticación de datos de manera segura (dos aplicaciones
- 20 importantes de la criptografía) utilizando diversos procedimientos criptográficos conocidos. La forma de generar y distribuir una clave criptográfica secreta entre dos o más partes supone un reto importante para la criptografía: es lo que se denomina el problema de la distribución de la clave. Se han propuesto varios métodos para resolver el problema de la distribución de la clave, de los que ofrecemos aquí algunos
- 25 ejemplos. El primer ejemplo es la criptografía de clave pública. En la criptografía de clave pública, el problema de la distribución de la clave se resuelve mediante suposiciones computacionales. Hay un par de claves, por ejemplo una para el cifrado y otra para el descifrado. Dada la clave de cifrado, en principio está disponible toda la información necesaria para obtener también la clave de descifrado. Sin embargo, si se
- 30 asume que los ordenadores convencionales no pueden resolver de manera eficiente ciertos problemas, como por ejemplo la factorización de números enteros grandes, en la práctica a un ordenador convencional le llevaría demasiado tiempo descubrir la clave de descifrado. El segundo ejemplo de un método para la distribución de claves es la criptografía cuántica, más específicamente, la distribución cuántica de claves
- 35 (QKD, del término anglosajón "Quantum Key Distribution"). Para ver un resumen de la

QKD, consúltese, por ejemplo, Lo, H.-K., Curty, M. and Tamaki, K. Secure quantum key distribution. *Nat. Photon.* 8, 595-604 (2014). En la QKD, el problema de la distribución de la clave se resuelve utilizando el teorema de no clonación en mecánica cuántica. Por ejemplo, en el conocido protocolo estándar de QKD propuesto por

5 Bennett-Brassard, y denominado BB84 (véase, por ejemplo, Bennett, C. H. and Brassard, G. Quantum cryptography: public key distribution and coin tossing. *Proc. IEEE Int. Conf. Comp. Systems Signal Processing* 175- 179 (IEEE, 1984)), una parte, digamos Alice, envía a una segunda parte, Bob, una secuencia de fotones preparados en diferentes estados de polarización, los cuales se eligen entre dos posibles bases

10 conjugadas, X y Z. Para cada fotón recibido, Bob selecciona aleatoriamente una de las dos bases conjugadas y realiza una medición del fotón con la base elegida. Asimismo, Bob registra el resultado de su medición y la elección de la base. A continuación, a través de un canal autenticado, Alice y Bob transmiten las bases en las que prepararon y midieron los fotones respectivamente. Descartan todos los datos

15 asociados a eventos donde los fotones fueron enviados y medidos en bases diferentes, y utilizan los datos restantes para generar una clave que se denomina "clave filtrada" (del término anglosajón "sifted key"). Para comprobar si se ha producido algún ataque en el canal de comunicaciones Alice y Bob calculan la tasa cuántica de error de bit (QBER, del término anglosajón "quantum bit error rate") de un subconjunto

20 de datos seleccionado aleatoriamente entre la "clave filtrada", y verifican que la QBER está por debajo de un determinado valor límite. En caso de que sea así, Alice y Bob emplean protocolos clásicos de post-procesado de datos como, por ejemplo, técnicas de corrección de errores y de amplificación de privacidad y generan una clave criptográfica segura a partir de la "clave filtrada". De esta manera, si se produce un

25 ataque en el canal, la persona que intenta efectuarlo, digamos Eve (del término anglosajón "eavesdropper"), al no disponer de la información sobre las bases empleadas, introduciría una perturbación inevitable en las señales transmitidas, que sería detectable por los usuarios legítimos, Alice y Bob. El tercer ejemplo de un método para la distribución de claves es el protocolo de acuerdo de clave pública de

30 Maurer (véase Maurer U. M. Secret key agreement by public discussion from common information. *IEEE Trans. on Inf. Theory* 39, 733-742 (1993)). En él, por ejemplo, dos partes, Alice y Bob, podrían recibir señales de una fuente común, Charles, en una galaxia lejana. Si asumimos que el ruido que experimentan los dispositivos receptores de las dos partes es independiente del ruido sufrido por los dispositivos receptores de

35 Eve, entonces las dos partes, Alice y Bob, pueden extraer una clave común de tal

manera que ésta sea segura frente a las escuchas de Eve.

En criptografía se utilizan a menudo esquemas de compartición de secretos. En un esquema de compartición de secretos, suele ser importante definir una estructura de acceso. Por ejemplo, en un esquema de compartición de secretos de tipo umbral (n, r) , un distribuidor divide una secuencia de datos secreta, digamos K , en diferentes porciones que se distribuyen entre n partes de tal manera que cualquier subconjunto de r partes que colaboren juntas sea capaz de recuperar el valor de la secuencia de datos, K , pero cualquier subconjunto de menos de r partes no tendría absolutamente ninguna información sobre el valor de la secuencia de datos, K . Nótese que existen esquemas de compartición de secretos de tipo umbral (n, r) , un distribuidor de compartición de secretos que no son de tipo umbral y que permiten estructuras de acceso más generales. Para afrontar la posibilidad de que el propio distribuidor sea deshonesto, y con el fin de garantizar que las porciones de datos distribuidas durante los diferentes pasos del protocolo de compartición de secretos sean consistentes, se han propuesto esquemas de compartición de secretos verificables. En cuanto a posibles aplicaciones, los esquemas de compartición de secretos se pueden utilizar por ejemplo para dividir una clave criptográfica generada en un módulo de seguridad hardware (HSM, del término anglosajón "hardware secure module"). Nótese que, en estos casos, a menudo se asume que los canales de comunicaciones no introducen ruido, pero alguna de las partes puede ser no confiable.

En cambio, tanto en el modelo de QKD como en el protocolo de acuerdo de clave pública de Maurer, se asume que los canales de comunicaciones utilizados para generar una clave criptográfica pueden ser ruidosos. Por ejemplo, los sistemas de QKD a menudo utilizan como señales, pulsos de luz coherente fuertemente atenuados o pares de fotones entrelazados. Estas señales pueden enviarse a largas distancias a través de espacio libre o mediante fibras ópticas. Por ejemplo, recientemente se ha implementado un sistema de QKD sobre un canal de fibra óptica de muy bajas pérdidas de 404 km de longitud. Para ello se ha empleado un protocolo de QKD que proporciona seguridad sin necesidad de caracterizar el dispositivo de medición que mide los fotones recibidos ("measurement-device-independent quantum key distribution protocol" en la terminología anglosajona). Asimismo, se han distribuido pares de fotones entrelazados a través de 1200 km (empleando un enlace vía satélite de comunicaciones cuánticas). Nótese que en los sistemas de QKD un posible

atacante, Eve, puede tratar de obtener información sobre las señales transmitidas por el canal cuántico de comunicaciones (un canal cuántico de comunicaciones es un canal de comunicaciones que puede transmitir información cuántica). Debido al ruido intrínseco introducido por los canales y a posibles ataques de Eve, un canal cuántico presenta a menudo una tasa cuántica de error de bit (QBER) del orden del 0,5% al 13%. Para poder generar una clave criptográfica secreta en esta situación, los sistemas de QKD normalmente asumen que, además de un canal cuántico de comunicaciones, entre Alice y Bob hay un canal clásico autenticado (o un canal público). Dicho canal clásico autenticado puede ser utilizado en la fase de discusión pública de los sistemas de QKD. Por ejemplo, este canal permite a Alice y Bob comparar un subconjunto de señales cuánticas para estimar la QBER en una transmisión cuántica. Si la QBER es demasiado alta, Alice y Bob pueden simplemente abortar el protocolo de QKD. En caso contrario, esto es, cuando la QBER está por debajo de un cierto valor, Alice y Bob proceden a extraer una clave criptográfica segura mediante el empleo de algún protocolo clásico de post-procesado de datos. Este protocolo clásico de post-procesado de datos puede incluir varios pasos o protocolos como, por ejemplo, la post-selección de datos, la adición de ruido, la estimación de ciertos parámetros, la reconciliación de la información (que típicamente incluye un paso de corrección de errores y un paso de verificación), y la amplificación de privacidad. Esto es así porque existen ciertos problemas que necesitan ser resueltos para poder extraer una clave criptográfica segura. El primero de estos problemas es que las "claves filtradas" que obtienen Alice y Bob, k_A^{raw} y k_B^{raw} respectivamente, pueden ser diferentes entre sí. Es por ello que es importante que Alice y Bob realicen algún procedimiento para corregir posibles discrepancias entre sus "claves filtradas"; este proceso se denomina reconciliación de la información. Una manera de realizar la reconciliación de la información es que Alice emplee algún código de corrección de errores convencional y que calcule y anuncie públicamente el síndrome de error resultante. Bob, al recibir la información sobre el síndrome de error, puede reconciliar su clave con la de Alice. Como último paso, Alice y Bob pueden realizar un proceso de verificación de errores para confirmar que las nuevas claves obtenidas, que denominaremos k_A^{rec} y k_B^{rec} respectivamente, están reconciliadas, esto es, ambas son iguales con alta probabilidad. En segundo lugar, hay que tener en cuenta que un posible atacante, Eve, podría tener información parcial sobre el contenido de la clave reconciliada. El objetivo del procedimiento de amplificación de privacidad consiste precisamente en eliminar con alta probabilidad cualquier

información residual que Eve pudiese tener sobre la clave reconciliada. La amplificación de privacidad es un proceso en el que una secuencia larga de números parcialmente seguros se comprime en una secuencia más corta de números que es casi perfectamente segura frente a un adversario, Eve. La amplificación de privacidad puede lograrse aplicando una función hash 2-universal, véase por ejemplo Bennett, C. H., Brassard, G., Crepeau, C. y Maurer, U. M. Generalized privacy amplification. IEEE Trans. on Inf. Theory 41, 1915-1923 (1995). Un ejemplo particularmente simple de amplificación de privacidad consiste en el uso de "hashing" aleatorio. Más concretamente, dada una secuencia binaria de entrada de n bits representada por un vector columna, X , se podría generar primero una matriz, M , de $m \times n$ (donde $m < n$) bits aleatorios y luego calcular la secuencia binaria de salida de m bits, $Y=M \cdot X$, simplemente realizando la multiplicación de las matrices M y X en álgebra de módulo 2. En el caso de los sistemas de QKD, Alice podría elegir las entradas de M y luego transmitir sus valores a Bob a través de un canal clásico autenticado.

Los esquemas de compartición de secretos, los sistemas de QKD y el protocolo de acuerdo de clave pública de Maurer pueden proporcionar seguridad en el paradigma de teoría de la información (del término anglosajón "information-theoretic security"), también denominada seguridad incondicional. Se trata de seguridad basada en principios de la teoría de la información y que no depende de ninguna hipótesis sobre la capacidad computacional del posible atacante, Eve.

El hackeo cuántico de implementaciones prácticas de sistemas de QKD ha suscitado gran interés recientemente. Estos ataques suelen explotar el hecho de que los dispositivos reales de los sistemas de QKD se comportan de manera distinta a la considerada en los modelos matemáticos que se emplean normalmente para demostrar la seguridad de estos sistemas. Esto abre brechas de seguridad que podrían ser utilizadas por Eve para atacar las implementaciones de los sistemas de QKD. Por ejemplo, se ha propuesto un ataque que, mediante el empleo de una memoria, puede quebrantar la seguridad de incluso aquellos sistemas de QKD cuya seguridad teórica no requiere la caracterización de sus dispositivos (estos sistemas de QKD se denominan DI-QKD, del término anglosajón "device-independent QKD"). En este tipo concreto de ataques, Eve esconde una memoria en el sistema transmisor de, por ejemplo, Alice con el objetivo de almacenar todas las claves criptográficas generadas en cada una de las ejecuciones del sistema de QKD. A continuación, esta

memoria envía la información sobre las claves criptográficas generadas al canal de comunicaciones, esto es, a Eve, escondiéndola por ejemplo en la decisión de abortar o no una sesión concreta de QKD posterior, o escondiéndola en la discusión pública de alguno de los protocolos de post-procesado de datos empleado en ejecuciones posteriores del sistema de QKD. Esto podría hacerse mediante la modificación adecuada de la "clave bruta" (esto es, de la clave a partir de la cual se obtiene la "clave filtrada") y/o la información del protocolo que se envía, en cada ejecución, a la unidad/unidades clásicas de post-procesado de datos (CLPU, del término anglosajón "classical post-processing unit") del sistema de QKD. Este tipo de ataques basado en la utilización de memorias, constituye un ejemplo de los ataques basados en "canales encubiertos" (del término anglosajón "covert channels") que, junto con los caballos de Troya, o troyanos, son conocidos por representar retos mayúsculos para la seguridad de los sistemas criptográficos convencionales. Nótese que los caballos de Troya pueden ser ocultados tanto en modificaciones hardware como en modificaciones software de los sistemas criptográficos.

RESUMEN DE LA INVENCION

Se resuelven o eluden los problemas encontrados en el estado de la técnica anterior, y se obtienen normalmente ventajas técnicas, a través de las realizaciones expuestas que proporcionan un método y un sistema según el cual dos o más estaciones criptográficas generan una clave segura común (compartida) en el marco de una estructura de acceso prescrita, en presencia de un canal ruidoso en una red y posiblemente de componentes no confiables dentro de una o varias estaciones criptográficas.

Teniendo en cuenta la descomposición funcional de una estación criptográfica en unidades de generación de claves y unidades clásicas de post-procesado de datos, la presente invención proporciona seguridad en presencia tanto de unidades de generación de claves no confiables como en presencia de unidades clásicas de post-procesado de datos no confiables. En una realización, para contrarrestar a las unidades de generación de claves no confiables, las unidades clásicas de post-procesado de datos emplean técnicas de amplificación de privacidad y garantizan de este modo seguridad pese a la información obtenida por los componentes no confiables dentro de las estaciones criptográficas. Para contrarrestar a las unidades

clásicas de post-procesado de datos no confiables se divide una clave criptográfica "bruta" (del término anglosajón "raw key") en varias porciones y se distribuyen éstas entre múltiples unidades clásicas de post-procesado de datos. La invención puede utilizarse para evitar ataques que exploten canales encubiertos y también para
 5 protegerse frente a la presencia de troyanos tanto en el hardware como en el software. De esta manera, se puede garantizar la seguridad de la clave criptográfica generada en el paradigma de la teoría de la información. Esto es, la seguridad de la clave generada no depende de ninguna hipótesis sobre la capacidad computacional del posible atacante. Esta clave puede emplearse posteriormente para lograr
 10 comunicaciones y autenticación de datos incondicionalmente seguras.

En un primer aspecto, la presente invención propone un método para la generación de claves criptográficas seguras en presencia de unidades no confiables en un sistema criptográfico. Este sistema criptográfico incluye una primera y una segunda estaciones
 15 criptográficas (A, B), y cada estación incluye n unidades de generación de claves brutas (del término anglosajón "key generation unit"), KGU^A_i , KGU^B_i con $i=1, 2, \dots, n$, y donde $n>1$, y, como mínimo, una unidad clásica de post-procesado de datos $CLPU^A$, $CLPU^B$, incluyendo el método los siguientes pasos:

- 20 - Cada par de unidades de generación de claves brutas, KGU^A_i y KGU^B_i , con $i=1, 2, \dots, n$, genera un par de secuencias de datos (clave bruta) y envía (a través de canales de comunicaciones seguros) la secuencia de datos generada por KGU^A_i a, como mínimo, una unidad clásica de post-procesado de datos de la primera estación criptográfica, y envía la secuencia de datos generada por KGU^B_i a, como mínimo, una unidad clásica
 25 de post-procesado de datos de la segunda estación criptográfica. Cada par de secuencias de datos (una generada por una unidad de generación de claves brutas KGU^A_i de la primera estación criptográfica y otra generada por una unidad de generación de claves brutas KGU^B_i de la segunda estación criptográfica, con $i=1, 2, \dots, n$,) normalmente presentará correlaciones en el sentido de que estas secuencias de
 30 datos se generarán utilizando un mecanismo (por ejemplo, QKD, distribución cuántica de claves) que permite que cada par de secuencias de datos de la primera y segunda estaciones criptográficas estén estadísticamente correlacionadas. Esto es, cada par de secuencias de datos no son estadísticamente independientes entre sí, sino que son estadísticamente dependientes, por lo que si se conoce el valor de una de las
 35 secuencias de datos se podría obtener información sobre el valor de la otra.

- Las unidades clásicas de post-procesado de datos (al menos una) de la primera y la segunda estaciones criptográficas, CLPU^A, CLPU^B:

5 - aplican un procedimiento de post-procesado de datos a cada secuencia de datos recibida y generan una clave criptográfica, K^A_i, K^B_i , con $i=1, 2, \dots, n$, o un símbolo de error (se generará un símbolo de error si la unidad de generación de claves brutas no ha podido generar una secuencia de datos a partir de la cual pueda obtenerse una clave criptográfica) por cada unidad de generación de claves brutas, donde el procedimiento de post-procesado de datos incluye al
10 menos una operación de reconciliación de la información entre las unidades clásicas de post-procesado de datos de ambas estaciones criptográficas a través de un canal de comunicaciones autenticado y un primer procedimiento de amplificación de privacidad para extraer una clave más corta;

15 - concatenan las claves criptográficas generadas para formar una primera clave criptográfica concatenada $K^{A'}=[K^A_1, K^A_2, \dots, K^A_M]$ y una segunda clave criptográfica concatenada $K^{B'}=[K^B_1, K^B_2, \dots, K^B_M]$ donde M es el número de pares de claves criptográficas generadas en ambas estaciones criptográficas que difieren del símbolo de error;

20 - aplican un procedimiento adicional de amplificación de privacidad a la primera clave criptográfica concatenada y a la segunda clave criptográfica concatenada para extraer una primera y una segunda claves criptográficas seguras, respectivamente, K^A y K^B .

25 Según otro aspecto, se propone un método para la generación de claves criptográficas seguras en presencia de unidades no confiables de un sistema criptográfico, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), y cada estación criptográfica incluye al menos una unidad de generación de claves brutas,
30 KGU^A, KGU^B respectivamente y más de una unidad clásica de post-procesado de datos CLPU^A_l, CLPU^B_{l'}, con $l=1, 2, \dots, s$ y $l'=1, 2, \dots, s'$, donde el método incluye los siguientes pasos:

35 - KGU^A genera s secuencias de datos y cada secuencia de datos generada es enviada a una unidad diferente CLPU^A_l, y KGU^B genera s' secuencias de datos y cada

secuencia de datos generada es enviada a una unidad diferente CLPU^{B₁}. Cada par de secuencias de datos (una generada por la unidad de generación de claves brutas KGU^A de la primera estación criptográfica y otra generada por la unidad de generación de claves brutas KGU^B de la segunda estación criptográfica) presentará correlaciones en el sentido de que estas secuencias de datos se generarán utilizando un mecanismo (por ejemplo, QKD) que permite que cada par de secuencias de datos de la primera y segunda estaciones criptográficas estén estadísticamente correlacionadas. Esto es, cada par de secuencias de datos no son estadísticamente independientes entre sí, sino que son estadísticamente dependientes, por lo que si se conoce el valor de una de las secuencias de datos se podría obtener información sobre el valor de la otra.

- Cada unidad clásica de post-procesado de datos de la primera y segunda estaciones criptográficas:

- aplica un procedimiento de post-procesado de datos a cada secuencia de datos recibida y genera o bien una clave criptográfica o un símbolo de error por cada secuencia de datos recibida, donde el procedimiento de post-procesado de datos incluye al menos una operación de reconciliación de la información entre las unidades clásicas de post-procesado de datos de las dos estaciones criptográficas a través de un canal de comunicaciones autenticado y un primer procedimiento de amplificación de privacidad para extraer una clave más corta;

- divide las claves criptográficas generadas en dos o más porciones y las distribuye entre el resto de las unidades clásicas de post-procesado de datos de la primera y segunda estaciones criptográficas, respectivamente;

- genera una porción de una clave criptográfica segura aplicando un procedimiento de verificación de errores y una operación adicional de amplificación de privacidad a las porciones de las claves criptográficas recibidas;

Según otro aspecto, se propone un método para la generación de claves criptográficas seguras en presencia de unidades no confiables de un sistema criptográfico, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), y cada estación criptográfica incluye al menos una unidad de generación de claves brutas,

KGU^A , KGU^B respectivamente, y más de una unidad clásica de post-procesado de datos $CLPU_i^A$, $CLPU_{i'}^B$, con $i = 1, 2, \dots, s$ y $i' = 1, 2, \dots, s'$, donde el método incluye los siguientes pasos:

- 5 - Las unidades de generación de claves brutas (una como mínimo) de la primera y la segunda estaciones criptográficas generan una secuencia de datos, R^A , R^B respectivamente, y dividen las secuencias de datos generadas en dos o más porciones y las distribuyen entre las unidades clásicas de post-procesado de datos de la primera y la segunda estaciones criptográficas respectivamente, donde $K_{ij}^{A'}$ es la
10 porción j^a de R^A recibida por $CLPU_i^A$ y $K_{i'j'}^{B'}$ es la porción j'^a de R^B recibida por $CLPU_{i'}^B$. El par de secuencias de datos (una generada por la unidad de generación de claves brutas KGU^A de la primera estación criptográfica y otra generada por la unidad de generación de claves brutas KGU^B de la segunda estación criptográfica) presentará correlaciones en el sentido de que estas secuencias de datos se generarán utilizando
15 un mecanismo (por ejemplo, QKD) que permite que las secuencias de datos de la primera y segunda estaciones criptográficas estén estadísticamente correlacionadas. Esto es, las secuencias de datos generadas no son estadísticamente independientes entre sí, sino que son estadísticamente dependientes, por lo que si se conoce el valor de una secuencia de datos se podría obtener información sobre el valor de la otra.
20 El número de porciones j , j' , en las que se dividen las secuencias de datos podría ser igual al número de unidades CLPU (de tal modo que cada CLPU reciba una porción) o inferior o superior (de tal modo que cada CLPU podría recibir más de una porción).
- 25 - Cada unidad clásica de post-procesado de datos de la primera y segunda estaciones criptográficas:

- obtiene de cada porción recibida de la secuencia de datos R^A , R^B una porción, $K_{ij, key}^{A'}$, $K_{i'j', key}^{B'}$, de una secuencia de datos que se utilizará para generar la
30 clave criptográfica. Las porciones obtenidas $K_{ij, key}^{A'}$, $K_{i'j', key}^{B'}$ podrían ser simplemente una parte de la porción recibida de las secuencias de datos R^A , R^B . Esto es, las porciones recibidas de R^A , R^B podrían dividirse en varias subsecuencias y una de ellas se utilizará para la generación de claves; de cada porción recibida de R^A , R^B también se podría obtener una porción, $K_{ij, est}^{A'}$,
35 $K_{i'j', est}^{B'}$, de una secuencia de datos que se utilizará para la estimación de

parámetros y estas porciones $K'_{ij,est}^A$, $K'_{ij,est}^B$, se enviarán al resto de unidades clásicas de post-procesado de datos de las dos estaciones criptográficas.

- 5 - aplica un procedimiento de post-procesado de datos a las porciones $K'_{ij,key}^A$, $K'_{ij,key}^B$ y genera porciones de una clave criptográfica segura, donde el procedimiento de post-procesado de datos incluye al menos una operación de reconciliación de la información entre las unidades clásicas de post-procesado de datos de ambas estaciones criptográficas a través de un canal de comunicaciones autenticado, y un procedimiento de amplificación de privacidad para extraer una clave más corta.
- 10

En esta última realización, la operación de reconciliación de la información podría incluir un procedimiento de corrección de errores que implique:

- 15 - aplicar determinadas matrices predefinidas M_{EC} a las porciones de datos $K'_{ij,key}^A$, $K'_{ij,key}^B$ para obtener las secuencias de datos $s_{ij}^A = M_{EC} * K'_{ij,key}^A$, $s_{ij}^B = M_{EC} * K'_{ij,key}^B$ respectivamente;
- 20 - obtener en cada una de las unidades clásicas de post-procesado de datos una secuencia de datos reconstruida s^A , s^B definida como $s^A = s_1^A \oplus \dots \oplus s_q^A$ y $s^B = s_1^B \oplus \dots \oplus s_q^B$ respectivamente, donde s_j^A se obtiene a partir de s_{ij}^A utilizando una estrategia de decisión basada en votación por mayoría y s_j^B se obtiene a partir de s_{ij}^B utilizando una estrategia de decisión basada en votación por mayoría;
- 25 - modificar el valor de las secuencias de datos $K'_{ij,key}^A$, $K'_{ij,key}^B$ dependiendo de los valores obtenidos de s^A y s^B ;
- repetir los tres pasos del procedimiento de corrección de errores hasta que la tasa de error se encuentre por debajo de un umbral predefinido;

En esta última realización, la operación de reconciliación de la información podría incluir un procedimiento de verificación de errores que implique:

- 30 - que las unidades clásicas de post-procesado de datos de la primera estación criptográfica seleccionen aleatoriamente una función hash 2-universal, que denominaremos hash, y la apliquen a las porciones $K'_{ij,key}^A$ obtenidas a partir de $K'_{ij,key}^A$ mediante el procedimiento de corrección de errores, para obtener $h_{ij}^A = \text{hash}(K'_{ij,key}^A)$, y cada unidad clásica de post-procesado de datos de la
- 35 segunda estación criptográfica obtiene $h_{ij}^B = \text{hash}(K'_{ij,key}^B)$ donde las porciones

- $K_{ij, key}^B$ se obtienen a partir de $K_{ij, key}^B$ mediante el procedimiento de corrección de errores y, a continuación, cada unidad clásica de post-procesado de datos envía las porciones h_{ij}^A y h_{ij}^B a todas las unidades clásicas de post-procesado de datos de su propia estación criptográfica y a todas las unidades clásicas de post-procesado de datos de la otra estación criptográfica;
- 5 - obtener en cada unidad clásica de post-procesado de datos dos secuencias de datos reconstruidas h^A , h^B respectivamente y definidas como $h^A = h_1^A \oplus \dots \oplus h_q^A$ y $h^B = h_1^B \oplus \dots \oplus h_q^B$, respectivamente, donde h_j^A se obtiene a partir de h_{ij}^A utilizando una estrategia de decisión basada en votación por mayoría y h_j^B se obtiene a partir de h_{ij}^B utilizando una estrategia de decisión basada en votación por mayoría.
- 10 - cada una de las unidades clásicas de post-procesado de datos comprueba si $h^A = h^B$ y si son iguales pasan al procedimiento de amplificación de privacidad, de lo contrario producen un símbolo de abortar.

15

En esta última realización, el procedimiento de amplificación de privacidad puede incluir:

- que las unidades clásicas de post-procesado de datos de la primera estación criptográfica seleccionen aleatoriamente una función hash 2-universal, que denominaremos hashPA, y la apliquen a las porciones $K_{ij, key}^A$ para obtener porciones de una clave criptográfica segura como $K_{ij}^A = \text{hashPA}(K_{ij, key}^A)$, y cada unidad clásica de post-procesado de datos de la segunda estación criptográfica obtiene porciones de una clave criptográfica segura como $K_{ij}^B = \text{hashPA}(K_{ij, key}^B)$.
- 20
- 25

25

Según otro aspecto, se propone un método para la generación de claves criptográficas seguras en presencia de unidades no confiables en un sistema criptográfico, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), y cada estación criptográfica incluye una pluralidad de unidades de generación de claves brutas, KGU_i^A , KGU_i^B con $i=1, 2, \dots, n$, $n>1$ y una pluralidad de unidades clásicas de post-procesado de datos $CLPU_l^A$, $CLPU_l^B$, $l=1, 2, \dots, s$, $l'=1, 2, \dots, s'$, donde el método incluye los siguientes pasos:

30

- Cada unidad de generación de claves brutas de la primera y segunda estaciones criptográficas genera secuencias de datos, R_i^A , R_i^B con $i=1, 2, \dots, n$, respectivamente,
- 35

y divide las secuencias de datos generadas en dos o más porciones y las distribuye entre las unidades clásicas de post-procesado de datos de la primera y segunda estaciones criptográficas respectivamente. Cada par de secuencias de datos (R^A_i , R^B_i con $i = 1, 2, \dots, n$) presentará correlaciones en el sentido de que estas secuencias de datos se generarán utilizando un mecanismo (por ejemplo, QKD) que permite que R^A_i y R^B_i estén estadísticamente correlacionadas, con $i = 1, 2, \dots, n$. Esto es, cada par de secuencias de datos no son estadísticamente independientes entre sí, sino que son estadísticamente dependientes, por lo que si se conoce el valor de una de las secuencias de datos, por ejemplo R^A_i , se podría obtener información sobre el valor de la otra, R^B_i .

- Cada unidad clásica de post-procesado de datos aplica un procedimiento de post-procesado de datos a cada porción de datos recibida, generando o bien porciones, K'^{A}_{ij} , K'^{B}_{ij} , de una primera clave criptográfica, o un símbolo de error por cada porción de la secuencia de datos recibida, donde el procedimiento de post-procesado de datos incluye al menos una operación de reconciliación de la información entre las unidades clásicas de post-procesado de datos de ambas estaciones criptográficas a través de un canal de comunicaciones autenticado y un primer procedimiento de amplificación de privacidad, donde K'^{A}_{ij} es la porción j^a de la clave criptográfica correspondiente a la secuencia R^A_i obtenida por la unidad $CLPU^A_i$, y K'^{B}_{ij} es la porción j^a de la clave criptográfica correspondiente a la secuencia R^B_i obtenida por la unidad $CLPU^B_i$;

- Cada unidad clásica de post-procesado de datos $CLPU^A_i$ obtiene porciones de una clave criptográfica segura concatenando primero las porciones K'^{A}_{ij} de la clave criptográfica obtenida en el paso previo, y cada unidad clásica de post-procesado de datos $CLPU^B_i$ obtiene porciones de una clave criptográfica segura concatenando las porciones K'^{B}_{ij} de la clave criptográfica obtenida en el paso previo, y aplicando a continuación un procedimiento adicional de amplificación de privacidad a las secuencias concatenadas obtenidas. Este último paso puede incluir:

- Sub-paso de concatenación: cada $CLPU^A_l$, con $l = 1, \dots, s$ obtiene secuencias de datos $K''^A_{ij} = [0_1, \dots, 0_{i-1}, K'^{A}_{ij}, 0_{i+1}, \dots, 0_M]$, donde 0_i , con $i = 1, \dots, M$, representa un vector cero, y M es el número de pares de unidades de generación de claves brutas (cada par se compone por una unidad de generación de clave bruta de cada estación criptográfica) que generan secuencias de datos que, tras la ejecución de los pasos previos, resultan

en una clave y no en el símbolo de error, y cada $CLPU_{l'}^B$, con $l'=1,...,s'$ obtiene secuencias de bits $K''_{l'ij}^B=[0_1, ..., 0_{i-1}, K_{l'ij}^B, 0_{i+1}, ..., 0_M]$, con $i=1,..., M$;

5 - Sub-paso de amplificación de privacidad: Las unidades $CLPU_l^A$, con $l=1,...,s$, seleccionan aleatoriamente una función hash 2-universal, hashPA, y luego obtienen porciones de una clave criptográfica segura como $K_{l'ij}^A = \text{hashPA}(K''_{l'ij}^A)$, y cada $CLPU_{l'}^B$, con $l'=1,...,s'$ obtiene porciones de una clave criptográfica segura como $K_{l'ij}^B = \text{hashPA}(K''_{l'ij}^B)$.

10 El par de secuencias de datos generado por cada par de unidades de generación de claves brutas (una de la primera estación criptográfica y una de la segunda estación criptográfica respectivamente) puede generarse utilizando un mecanismo de distribución cuántica de claves.

15 Para dividir una secuencia de datos en porciones podría emplearse un esquema de compartición de secretos o un esquema de compartición de secretos verificable. Al finalizar los procedimientos descritos en los distintos métodos, podría aplicarse un protocolo de reconstrucción para obtener, a partir de las porciones de la clave criptográfica segura generadas, una clave criptográfica final en cada una de las
20 estaciones criptográficas.

En otros aspectos, se proponen sistemas para la generación de claves criptográficas seguras. Dichos sistemas incluirán una primera y una segunda estaciones criptográficas (A, B), y cada estación criptográfica incluirá al menos una unidad de
25 generación de claves brutas y al menos una unidad clásica de post-procesado de datos $CLPU^A$ $CLPU^B$, con medios para implementar los métodos propuestos anteriormente.

En un último aspecto de la presente invención, se divulga un programa informático que
30 incluye medios de código de programa informático adaptados para realizar los pasos de los métodos descritos, cuando dicho programa se ejecuta en medios de procesamiento de una entidad de red de una red OFDM; dichos medios de procesamiento son por ejemplo un ordenador, un procesador digital de señales, una matriz de puertas programables (FPGA, del término anglosajón "field-programmable
35 gate array"), un circuito integrado de aplicación específica (ASIC, del término

anglosajón “application-specific integrated circuit”), un microprocesador, un microcontrolador o cualquier otra forma de hardware programable. También se proporciona un medio de almacenamiento de datos digital no transitorio para almacenar un programa informático que incluye instrucciones, que hacen que un
 5 ordenador que ejecute el programa lleve a cabo todos los pasos de los métodos divulgados cuando el programa se ejecuta en un ordenador.

Descripción de las figuras

10

Para completar la descripción que se hace y con el fin de facilitar la comprensión de las características de la invención, de acuerdo con el ejemplo preferente de realizaciones prácticas de ésta y acompañando dicha descripción como parte integrante de la misma, se adjunta un conjunto de esquemas a modo de ilustración y
 15 no limitativo que representan lo siguiente:

La Figura 1 muestra un diagrama esquemático de bloques de una configuración criptográfica que representa el estado de la técnica anterior.

20

La Figura 2 muestra un diagrama esquemático de bloques que ilustra un ataque basado en la utilización de una memoria contra una configuración criptográfica correspondiente al estado de la técnica anterior con dos estaciones criptográficas, A y B.

25

La Figura 3 muestra un diagrama esquemático de bloques de un procedimiento de generación de claves criptográficas seguras según una realización de la invención, en un escenario donde al menos una unidad de generación de claves KGU no es confiable.

30

La Figura 4 muestra un diagrama esquemático de bloques de un procedimiento de generación de claves criptográficas seguras según una realización de la invención, en un escenario donde al menos una unidad clásica de post-procesado de datos no es confiable.

35

La Figura 5 muestra diagramas esquemáticos de bloques que ilustran los protocolos

de distribución y de reconstrucción para un ejemplo de esquema de compartición de secretos verificable.

La Figura 6 muestra un diagrama esquemático de bloques de un procedimiento seguro de generación de claves criptográficas según una realización de la invención, en un escenario donde cada estación criptográfica contiene más de una unidad de generación de claves, KGU, y más de una unidad clásica de post-procesado de datos, CLPU.

10 Descripción de la invención

La invención describe métodos y sistemas para, en términos generales, generar una clave criptográfica segura común entre dos o más estaciones criptográficas (una estación que es capaz de generar claves criptográficas, por ejemplo, para comunicaciones o autenticación de datos seguras entre dos dispositivos electrónicos). Dicha clave común se genera en presencia de un canal ruidoso en una red de comunicaciones (comunicando las estaciones criptográficas) y posiblemente componentes no confiables dentro de las estaciones criptográficas.

Considérese una red de comunicaciones que permite la comunicación entre varias estaciones criptográficas junto con cualquier nodo intermedio o auxiliar. Cada estación criptográfica puede representar, por ejemplo, un nodo electrónico como sitio seguro como, por ejemplo, un centro de datos o unas instalaciones compartidas, un proveedor de servicios o un dispositivo electrónico de comunicaciones de un usuario final, como un teléfono inteligente o, por ejemplo, un teléfono móvil, un ordenador portátil, un iPad, una tableta, un PC... o, en términos generales, cualquier otro tipo de dispositivo electrónico que pueda generar claves criptográficas para comunicaciones o autenticación de datos con otro dispositivo. Para lograr comunicaciones o autenticación de datos de forma incondicionalmente segura entre, por ejemplo, dos estaciones criptográficas, tradicionalmente llamadas, A y B o Alice y Bob (donde se supone que Alice y Bob son las partes o usuarios que controlan cada estación criptográfica respectivamente) en presencia de un adversario o espía, Eve, es importante que Alice y Bob compartan una clave criptográfica, K, que sea aleatoria y secreta.

35

Una estación criptográfica contiene una o varias unidades de generación de claves (KGU) y una o varias unidades clásicas de post-procesado de datos (CLPU). Las unidades de generación de claves (también llamadas unidades de generación de claves brutas) generan datos estadísticamente correlacionados (también conocidos como datos primarios, o clave bruta, es decir, datos obtenidos directamente de una fuente de datos) entre partes distantes, los cuales son procesados posteriormente por las CLPUs para obtener la clave criptográfica. Por ejemplo, en el protocolo de acuerdo de clave pública de Maurer, las unidades de generación de claves podrían ser antenas que están recibiendo señales de una fuente común en una radiogalaxia distante. Las unidades clásicas de post-procesado de datos (también llamadas simplemente unidades de post-procesado) pueden ser cualquier dispositivo de procesamiento de datos convencional (también llamados módulos de procesamiento o unidades de procesamiento), como por ejemplo unidades centrales de procesamiento (CPUs), unidades de procesamiento gráfico (GPUs) y matrices de puertas programables (FPGAs) o cualquier otra unidad electrónica con capacidades de cálculo. Un par de KGUs distribuidas entre las dos estaciones criptográficas, A y B, utilizan algunos medios físicos para generar datos ruidosos (brutos) estadísticamente correlacionados entre las estaciones criptográficas. Estos medios físicos para generar datos estadísticamente correlacionados podrían ser, por ejemplo, sistemas de QKD. En el caso de los sistemas de QKD, estos datos brutos podrían ser datos relacionados con el estado de polarización en el que se preparan fotones individuales o con los estados de polarización que se obtienen tras medir éstos. Cada fotón individual tiene una polarización. Por ejemplo, en la base rectilínea Z, un fotón polarizado verticalmente podría representar un "0" mientras que un fotón polarizado horizontalmente podría representar un "1". Del mismo modo, en la base diagonal X, un fotón polarizado a 45 grados podría representar un "0", mientras que un fotón polarizado a 135 grados podría representar un "1". Cada KGU pasa entonces los datos brutos ruidosos estadísticamente correlacionados a una o varias unidades de post-procesado de datos para que sean procesados, obteniéndose la clave criptográfica K. Si bien éste es un argumento general, para facilitar su ilustración tomaremos el ejemplo de los sistemas de QKD. En los sistemas de QKD, Alice y Bob están conectados por dos canales de comunicaciones, uno cuántico y otro clásico. El canal cuántico se emplea para la transmisión de señales cuánticas, como por ejemplo fotones individuales preparados en diferentes estados de polarización. Estas señales cuánticas se pueden utilizar para generar una clave bruta. Eve puede atacar el canal cuántico, que puede ser una fibra

óptica o espacio libre u otro medio (por ejemplo, agua). El término “canal clásico” significa simplemente un canal de comunicaciones convencional que puede transmitir información convencional. El canal clásico puede ser de cualquier forma, lo que incluye líneas telefónicas, Internet, Ethernet, los canales conocidos de una red de comunicaciones móviles (GSM, UMTS, LTE, 3G, 4G o cualquier otro) o una red wifi, incluso un cable directo o, en general, cualquier canal de comunicaciones de cualquier red convencional de comunicaciones con cable o inalámbrica. El canal clásico también puede estar en el mismo medio que el canal cuántico, por ejemplo, a través de la multiplexación espacial, temporal o de frecuencia de las señales. A menudo se asume que el canal clásico está autenticado utilizando para ello cualquier mecanismo de autenticación de datos conocido. Las CLPUs reciben los datos ruidosos y estadísticamente correlacionados de la clave bruta y pueden aplicarles varias operaciones de procesamiento de datos (que pueden incluir, por ejemplo, la post-selección de datos, la adición de ruido, la estimación de parámetros, la reconciliación de la información, que suele incluir un paso de corrección de errores junto con un paso de verificación de errores, y la amplificación de privacidad). A continuación se describirán estas operaciones de forma más detallada.

A. Un método de distribución de claves del estado de la técnica anterior y su inseguridad

La Figura 1 muestra un método de distribución de claves del estado de la técnica anterior con dos estaciones criptográficas, A y B, en presencia de un espía, Eve, en el que cada estación criptográfica contiene una sola unidad de generación de claves (KGU) y una sola unidad clásica de post-procesado de datos (CLPU). En otras palabras, la estación criptográfica A contiene KGU^A y $CLPU^A$ mientras que la estación criptográfica B contiene KGU^B y $CLPU^B$. En las figuras, un superíndice indica si los dispositivos y las claves criptográficas corresponden a la estación criptográfica A o a la estación criptográfica B. Cada KGU genera una clave bruta, R, y cada CLPU produce una clave criptográfica, K. $CLPU^A$ y $CLPU^B$ están conectadas a través de un canal de comunicaciones convencional autenticado (“Canal Cl. A.” en la figura, también denominado canal clásico autenticado), y, dentro de cada estación criptográfica, KGU y CLPU están conectados a través de un canal clásico (convencional) (“Canal Cl.” en la figura). En los sistemas de QKD, un canal cuántico (“Canal C.” en la figura) conecta KGU^A y KGU^B ; en el caso del protocolo de acuerdo de clave pública de Maurer, las

KGUs podrían recibir señales clásicas difundidas por una fuente común.

- Por ejemplo, en un protocolo de QKD conocido (el protocolo estándar Bennett-Brassard BB84), dos partes distantes, Alice y Bob, desean establecer una clave
- 5 criptográfica segura entre ellas. La estación criptográfica A está controlada por una de las partes, Alice (por ejemplo, el usuario de un primer dispositivo electrónico de comunicaciones donde se encuentra la estación criptográfica A), mientras que la estación criptográfica B está controlada por otra parte, Bob (por ejemplo, el usuario de un segundo dispositivo electrónico de comunicaciones donde se encuentra la estación
- 10 criptográfica B). Ahora, KGU^A prepara y envía a través de un canal cuántico una secuencia de fotones preparados en diferentes estados de polarización a KGU^B , estos estados de polarización son elegidos por KGU^A entre dos posibles bases conjugadas, X y Z. Para cada fotón recibido, KGU^B selecciona aleatoriamente una de las dos bases conjugadas y realiza una medición. KGU^B registra el resultado de la medición y la
- 15 elección de la base. A continuación, KGU^A envía los datos de polarización, R^A , y otra información auxiliar relevante como la información sobre las bases a $CLPU^A$, y KGU^B envía los datos de polarización, R^B , y otra información auxiliar relevante como la información sobre las bases a $CLPU^B$.
- 20 A través de un canal autenticado ("Canal Cl. A." en la Figura 1), $CLPU^A$ y $CLPU^B$ transmiten sus bases de preparación y medición respectivamente. Descartan todos los datos de polarización enviados y recibidos en bases diferentes y utilizan los datos restantes para generar una "clave filtrada". Para comprobar si se ha producido algún
- 25 ataque, $CLPU^A$ y $CLPU^B$ calculan la tasa cuántica de error de bit (QBER) de un subconjunto de datos seleccionado aleatoriamente entre la clave filtrada y verifican que la QBER está por debajo de un determinado valor umbral. Aplicando protocolos clásicos de post-procesado de datos, como por ejemplo reconciliación de la información (que normalmente incluye un paso de corrección de errores junto con un paso de verificación de errores), y amplificación de privacidad, $CLPU^A$ y $CLPU^B$
- 30 generan una clave criptográfica segura, K^A y K^B , donde con una alta probabilidad $K^A=K^B$ y K^A es segura frente a un espía.

En este tipo de configuración, se asume comúnmente que todas las KGUs (es decir, KGU^A y KGU^B) y todas las CLPU (es decir, $CLPU^A$ y $CLPU^B$) son confiables. Por

35 ejemplo, en este marco se incluyen tanto los sistemas de QKD que no requieren

caracterizar los dispositivos (del término anglosajón “device-independent QKD”) como aquellos que requieren caracterizar los dispositivos (del término anglosajón “device-dependent QKD”). Los sistemas estándar de QKD requieren caracterizar los dispositivos y asumen que los dispositivos de QKD funcionan correctamente, por ejemplo, preparando el estado correcto y realizando mediciones perfectas de acuerdo con algunos protocolos teóricos. Por el contrario, los sistemas de QKD que no requieren caracterizar los dispositivos presentan la ventaja de permitir que los dispositivos de QKD funcionen de forma arbitraria siempre que no haya fugas de información (por ejemplo, acerca de la clave criptográfica final) desde las estaciones criptográficas, A y B, a un espía. Desafortunadamente, esta configuración del estado de la técnica anterior es altamente vulnerable frente a la presencia de programas y/o dispositivos maliciosos en el software y/o en el hardware. Por ejemplo, si la espía Eve planta una memoria digamos, por ejemplo, en KGU^A, entonces la seguridad de dicho método de distribución de claves correspondiente al estado de la técnica anterior puede verse comprometida. Esto se ilustra en la Figura 2. Eve podría usar esta memoria para almacenar la clave criptográfica generada en una sesión de QKD y luego filtrar esta información al canal de comunicaciones en ejecuciones posteriores del sistema de QKD. Para ello, Eve podría explotar por ejemplo el hecho de que cada ejecución de QKD suele asociarse con una decisión de abortar o no abortar dependiendo de la QBER observada. La memoria podría entonces decidir si hacer o no que KGU^A produzca una clave bruta, R^A, con una QBER elevada (y así obligar al protocolo a abortar) dependiendo del valor de un bit determinado de la clave generada en una ejecución de QKD previa. Alternativamente, la memoria también podría filtrar la clave criptográfica generada en una determinada ejecución de QKD simplemente ocultándola en la discusión pública de las ejecuciones de QKD posteriores.

Por lo tanto, la manera de distribuir una clave criptográfica segura cuando algunas de las KGUs o CLPUs no son confiables es un problema importante que no está resuelto por los sistemas de distribución de clave del estado de la técnica anterior.

30

B. Escenario 1: Un método de distribución de claves con unidades de generación de claves no confiables en una estación criptográfica.

El objetivo principal de la invención es lograr seguridad en la distribución de claves en presencia de dispositivos no confiables. Estos dispositivos no confiables podrían ser,

35

por ejemplo, KGUs o CLPUs o una combinación de ambas. En esta sección, se considera el caso en el que algunas KGUs pueden no ser confiables pero se asume que las CLPUs sí lo son.

- 5 En este caso, consideramos un nuevo protocolo de distribución de claves con múltiples KGUs, como se muestra en la Figura 3. Esta figura muestra dos estaciones criptográficas, A y B, en presencia de un espía Eve (por ejemplo, en los canales que conectan las estaciones criptográficas A y B y también en algunas de las KGUs de Alice y Bob, lo que hace que estas KGUs no sean confiables). Cada estación
- 10 criptográfica contiene n (más de una) unidades de generación de claves, KGUs, además de al menos una unidad clásica de post-procesado de datos, CLPU. La i^{a} KGU genera una clave bruta, R_i . El canal que conecta $CLPU^A$ y $CLPU^B$ es un canal de comunicaciones autenticado convencional ("Canal Cl. A." en la figura, también llamado canal clásico autenticado). El canal que conecta KGU_i^A y $CLPU^A$ así como el canal que
- 15 conecta KGU_i^B y $CLPU^B$ son canales clásicos (convencionales) seguros ("Canal Cl. S." en la figura) para todos los $i=1, \dots, n$. Además, en un sistema QKD, el canal que conecta KGU_i^A y KGU_i^B es un canal cuántico ("Canal C." en la figura) para todos los $i=1, \dots, n$.
- 20 Esto es, en esta figura, la estación criptográfica A tiene n (donde $n > 1$) unidades $KGU_1^A, KGU_2^A, \dots, KGU_n^A$, y la estación criptográfica B tiene n unidades $KGU_1^B, KGU_2^B, \dots, KGU_n^B$. Como comentario adicional, nótese que existen KGUs de bajo coste debido al desarrollo de transmisores para QKD basados en chips. Nuestra invención puede aprovechar estos dispositivos. Asimismo, nótese que los distintos pares de
- 25 KGUs se pueden comprar a diferentes proveedores. Nuestra invención nos permite aumentar la confiabilidad y construir un sistema seguro de generación de claves con componentes baratos de proveedores no confiables.

- Obviamente, si todas las KGUs están comprometidas es imposible lograr seguridad,
- 30 por lo que supondremos que al menos una de ellas no está comprometida (es decir, supondremos que existe al menos una KGU confiable). Así pues, resulta útil definir la estructura de acceso o la estructura adversaria. Por estructura adversaria nos referimos a qué subconjuntos de pares de unidades KGU_i^A y KGU_i^B con $i=1, \dots, n$, podrían ser deshonestos (no confiables) sin que ello afecte a la seguridad de la clave
- 35 final (un par de unidades KGU_i^A y KGU_i^B es deshonesto si al menos una de las

unidades lo es). En la definición de la estructura adversaria se podría distinguir entre los dispositivos deshonestos que están controlados pasivamente por Eve y los que están controlados activamente por Eve. Por ejemplo, un dispositivo controlado pasivamente por Eve puede filtrarle su información, pero por lo demás sigue
 5 correctamente todas las indicaciones del protocolo. Un dispositivo controlado activamente por Eve, por otra parte, puede filtrarle su información y, además, no tiene por qué seguir necesariamente las prescripciones del protocolo, pero su comportamiento está completamente controlado por Eve. Un ejemplo de estructura adversaria podría ser que a lo sumo 3 de 4 pares de unidades KGU^A_i y KGU^B_i pudieran
 10 ser deshonestos y estar controlados activamente por Eve. Esto significa que al menos uno de los 4 pares de unidades KGU^A_i y KGU^B_i es honesto.

En la Figura 3 asumimos que cada estación criptográfica, A y B, tiene al menos una unidad clásica de post-procesado de datos, $CLPU^A$ y $CLPU^B$ respectivamente. En este
 15 escenario, se supone que todas las CLPUs son confiables. Además, cada KGU^A_i está conectada a $CLPU^A$ a través de un canal clásico (convencional) seguro (denominado "Canal Cl. S. i^A" en la Figura 3), y cada KGU^B_i está conectada a $CLPU^B$ a través de un canal clásico (convencional) seguro (denominado "Canal Cl. S. i^B" en la Figura 3). En todo este texto, se entiende por "canal clásico o convencional seguro" un canal que
 20 proporciona tanto confidencialidad como autenticación. Esto podría lograrse, por ejemplo, utilizando el protocolo de cifrado denominado "libreta de un solo uso" o también conocido como cifrado de Vernam, junto con el esquema de autenticación de Wegman-Carter, o cualquier otro mecanismo conocido. Nótese que en la práctica también se puede utilizar como canal clásico seguro, por ejemplo, un canal protegido
 25 físicamente (por ejemplo, un cable físico que protege contra daños e intrusiones) y que sólo conecta los dispositivos prescritos. Nótese que todos los canales clásicos seguros se utilizan para conectar dispositivos situados dentro de una estación criptográfica. Además, hay un canal clásico (convencional) autenticado que conecta $CLPU^A$ y $CLPU^B$ (denominado "Canal Cl. A." en la Figura 3). Esto podría lograrse utilizando cualquier sistema de autenticación conocido. Además, en un sistema QKD, las
 30 unidades KGU^A_i y KGU^B_i , con $i=1, \dots, n$, se conectan entre sí a través de un canal cuántico, denominado "Canal C. i" en la Figura 3, al que puede acceder Eve. Obsérvese que todas las KGU^A_i y KGU^B_i podrían incluso compartir el mismo canal cuántico físico (por ejemplo, una sola fibra óptica) mediante diversas técnicas de
 35 multiplexación (por ejemplo, multiplexación en longitud de onda y multiplexación

espacial).

Cada unidad KGU^A_i de la estación criptográfica A envía la clave bruta generada, R^A_i , a $CLPU^A$ a través del "Canal Cl. S. i^A " mostrado en la Figura 3. Si la estación
 5 criptográfica A tiene más de una CLPU, entonces cada KGU^A_i puede o bien enviar su clave bruta R^A_i a cada CLPU o porciones de ésta a una o más de las CLPUs de la estación criptográfica A. Del mismo modo, cada unidad KGU^B_i envía la clave bruta generada, R^B_i , a $CLPU^B$ a través del "Canal Cl. S. i^B ". Si la estación criptográfica B tiene más de una CLPU, entonces cada KGU^B_i puede o bien enviar su clave bruta R^B_i
 10 o porciones de ésta a una o más de las CLPUs de la estación criptográfica B. Si se asume que aquí todas las CLPUs son confiables, entonces la transmisión directa de las claves brutas (es decir, sin dividir las en porciones) generadas por las diferentes KGUs a las CLPUs (dentro de cada estación criptográfica) no compromete la seguridad.

15

Las CLPUs pueden aplicar varios protocolos clásicos de post-procesado de datos a las claves brutas recibidas. Debido a las perturbaciones ambientales y a los posibles ataques de espionaje por parte de Eve, la clave bruta R^A_i generada por KGU^A_i podría ser diferente de la clave bruta R^B_i generada por KGU^B_i . Por lo tanto, estos protocolos
 20 clásicos del post-procesado de datos deberían incluir generalmente un protocolo para la reconciliación de la información y un protocolo de amplificación de privacidad para poder garantizar que las claves finales generadas, K^A y K^B , en las estaciones criptográficas A y B sean secretas y probablemente iguales entre sí. En un protocolo de reconciliación de la información, una o más de las CLPUs de las estaciones
 25 criptográficas A y B pueden primero intercambiar información para la corrección de errores, que se utiliza para corregir sus datos de tal manera que K^A y K^B sean con una alta probabilidad iguales entre sí. Esto se puede hacer utilizando cualquier procedimiento conocido (véase, por ejemplo, Brassard G. and Salvail, L. Secret key reconciliation by public discussion. Proc. of Advances in Cryptology (Eurocrypt 93),
 30 410-23 (1993)). A continuación, las estaciones criptográficas A y B podrían realizar un paso de verificación de errores para confirmar que K^A y K^B son efectivamente iguales entre sí con una alta probabilidad. El paso de amplificación de privacidad se utiliza para garantizar que la clave final, K^A y K^B , sea de hecho probablemente secreta (es decir, Eve apenas tiene información sobre ella). Para ello, la amplificación de
 35 privacidad elimina la información parcial que Eve pudiera haber obtenido sobre, por

- ejemplo, las claves brutas R_i^A , con $i=1,\dots,n$. Eve podría haber obtenido información sobre estas claves brutas espiando los canales cuánticos que conectan las unidades KGU_i^A y KGU_i^B para todos los i , así como escuchando el contenido del canal clásico autenticado que conecta $CLPU^A$ y $CLPU^B$. Además, todos los pares KGU_i^A y KGU_i^B que sean deshonestos podrían filtrarle directamente R_i^A a Eve. Como se mencionó anteriormente, la amplificación de privacidad se puede realizar aplicando una función hash 2-universal a la clave reconciliada (por ejemplo, multiplicando la clave reconciliada por una matriz aleatoria de entradas binarias).
- Además, los protocolos clásicos de post-procesado de datos pueden incluir, por ejemplo, la post-selección de datos, la adición de ruido, la estimación de parámetros y la verificación de errores. En cada uno de estos protocolos clásicos de post-procesado de datos, una o más de las CLPUs de las estaciones criptográficas A y B pueden utilizar el canal clásico autenticado entre ellas para intercambiar información y post-procesar sus datos. Por ejemplo, en el paso de post-selección de datos las CLPUs dividen las claves brutas R_i^A y R_i^B en diferentes conjuntos de datos. A modo de ejemplo, las CLPUs pueden dividir las claves brutas en tres conjuntos principales de datos. El primer conjunto de datos contiene los datos de las claves brutas que se post-procesarán para obtener una clave segura, K^A y K^B ; el segundo conjunto de datos (también denominado conjunto de datos para la estimación de parámetros) contiene los datos de las claves brutas que se utilizarán para la estimación de parámetros con el fin de determinar los parámetros necesarios para generar K^A y K^B a partir de los datos del primer conjunto de datos; y el tercer conjunto de datos contiene los datos de las claves brutas que se desechan. Por ejemplo, en el protocolo estándar de Bennett-Brassard BB84, Alice y Bob suelen descartar los datos de las claves brutas asociados a aquellos eventos en los que Alice y Bob emplean bases diferentes. Además, las CLPUs pueden añadir ruido a las claves brutas realizando la operación NOT o de negación lógica de algunos de sus bits (véase, por ejemplo, Renner, R., Gisin, N. and Kraus B. An information-theoretic security proof for QKD protocols. Phys. Rev. A 72, 012332 (2005)) o cualquier otro mecanismo conocido para añadir ruido. En el paso de estimación de parámetros, por otra parte, las CLPUs utilizan los datos del conjunto de datos para la estimación de parámetros con el objetivo de determinar cotas superiores o inferiores para aquellas cantidades o parámetros que son necesarios para generar una clave segura. Estos parámetros pueden incluir, por ejemplo, la QBER, la tasa cuántica de error de fase, el número de pulsos que contienen un solo fotón emitidos

por Alice y que contribuyen al conjunto de datos que se utiliza para generar una clave segura, etc. Finalmente, en el paso de verificación de errores, como ya se ha mencionado, las CLPUs confirman que K^A y K^B son iguales con cierta probabilidad. Un protocolo de reconciliación de la información puede constar de muchos pasos y la verificación de errores puede ser el último paso. El propósito de la verificación de errores es confirmar que el protocolo de reconciliación de la información ha tenido éxito. Para ello, podrían utilizar, por ejemplo, una función hash 2-universal para calcular un "hash" tanto de K^A como de K^B y, a continuación, comprobar si ambos "hash" son iguales.

10

A continuación describimos, a modo de ejemplo, un protocolo para la distribución de claves que logra seguridad frente a KGUs deshonestas. El protocolo se puede descomponer en dos pasos conceptuales principales. En particular, supongamos que todas las unidades KGU_i^A de la estación criptográfica A han enviado los datos de la clave bruta R_i^A a $CLPU^A$, y supongamos también que todas las unidades KGU_i^B de la estación criptográfica B han enviado los datos de la clave bruta R_i^B a $CLPU^B$. Entonces, en un primer paso, las unidades $CLPU^A$ y $CLPU^B$ post-procesan los datos R_i^A y R_i^B para obtener una clave secreta, K_i^A y K_i^B , o el símbolo de abortar $\perp_i$ para todo i . En caso de que se utilice un sistema QKD para la generación de claves brutas, nótese que se puede generar un símbolo de abortar por ejemplo cuando la tasa cuántica de error de bit (QBER) es superior a algún valor prescrito (por ejemplo, 11% en el caso de la prueba de seguridad de Shor-Preskill para el protocolo Bennett-Brassard BB84). Para ello, $CLPU^A$ y $CLPU^B$ pueden emplear algunos de los protocolos clásicos de post-procesado de datos descritos anteriormente. Estos protocolos clásicos de post-procesado de datos pueden incluir un paso de amplificación de privacidad para eliminar la información parcial que Eve pudiera tener sobre K_i^A y K_i^B debido a un ataque sobre el "Canal C. i" (véase la Figura 3) que conecta KGU_i^A y KGU_i^B , así como al hecho de que Eve escucha el contenido del canal clásico autenticado que conecta las unidades $CLPU^A$ y $CLPU^B$. A continuación viene un paso clave en esta realización de la invención. En particular, en un segundo paso, $CLPU^A$ y $CLPU^B$ aplican un paso adicional de amplificación de privacidad a $K^{A'} = [K^{A_1}, K^{A_2}, \dots, K^{A_M}]$ y $K^{B'} = [K^{B_1}, K^{B_2}, \dots, K^{B_M}]$, donde M representa el número de pares K_i^A y K_i^B que son diferentes del símbolo de abortar $\perp_i$. La finalidad de este segundo paso de amplificación de privacidad es eliminar la información que los pares deshonestos de KGUs pudieran filtrar a Eve acerca de, digamos, $K^{A'}$. Para simplificar la discusión,

consideremos por ejemplo que la longitud de todas las claves K_i^A y K_i^B es N bits para todos los $i=1, \dots, M$. Consideremos asimismo por ejemplo una estructura adversaria donde a lo sumo $t < n$ pares de unidades, KGU_i^A y KGU_i^B , pudieran ser deshonestos y estar controlados activamente por Eve. Esto significa que como mucho $t \cdot N$ bits de $K^{A'}$ podrían estar comprometidos y ser conocidos por Eve. Entonces, la aplicación de un

5 paso de amplificación de privacidad a $K^{A'}$ y $K^{B'}$ elimina esta información comprometida y se traduce en una clave final, K^A y K^B , de aproximadamente $(M-t) \cdot N$ bits. A continuación describimos la implementación paso a paso de este ejemplo específico. El objetivo es generar una clave criptográfica segura en el escenario estándar de

10 composabilidad de la criptografía. Más concretamente, el objetivo es generar una clave criptográfica epsilon-segura, K^A y K^B . Es decir, K^A y K^B deben ser epsilon_cor-correctas y epsilon_sec-secretas. Epsilon, epsilon_cor y epsilon_sec son parámetros de diseño del sistema de generación de claves que satisfacen la condición $\text{epsilon_cor} + \text{epsilon_sec} \leq \text{epsilon}$ (a grandes rasgos, una clave es epsilon_cor-

15 correcta si $K^A = K^B$ excepto por una pequeña probabilidad de, como mucho, epsilon_cor. Y una clave es epsilon_sec-secreta si es aleatoria y secreta frente a una espía, Eve, excepto por una pequeña probabilidad de fallo de, como mucho, epsilon_sec).

El protocolo para la generación de una clave criptográfica segura puede incluir los

20 pasos siguientes (esto es solamente una posible realización, y no todos los pasos citados son esenciales y obligatorios en todas las realizaciones de la presente invención):

Protocolo 1:

- 25
1. Distribución de las claves brutas: cada KGU_i^A envía a $CLPU^A$ la clave bruta R_i^A o el símbolo de abortar $\perp_i$. Además, cada KGU_i^B envía a $CLPU^B$ la clave bruta R_i^B o el símbolo de abortar $\perp_i$.
 - 30 2. Generación de una clave epsilon_cor-correcta $K^{A'}$ y $K^{B'}$: $CLPU^A$ y $CLPU^B$ utilizan un protocolo de post-procesado de datos para generar una clave (epsilon_cor/M)-correcta y (epsilon_sec/M)-secreta (con $\text{epsilon_cor} + \text{epsilon_sec} \leq \text{epsilon}$), K_i^A y K_i^B , a partir de R_i^A y R_i^B , o generar el símbolo de abortar $\perp_i$ (para indicar que el protocolo de post-procesado de datos
 - 35 no ha tenido éxito y que no ha sido posible generar una clave válida a partir de

los claves brutas) para todos los $i = 1, \dots, n$. Después, $CLPU^A$ concatena las M claves K_i^A que sean diferentes de $\perp_i$ para formar $K^{A'} = [K_1^A, K_2^A, \dots, K_M^A]$. Del mismo modo, $CLPU^B$ concatena las M claves K_i^B que sean diferentes de $\perp_i$ para formar $K^{B'} = [K_1^B, K_2^B, \dots, K_M^B]$.

5

3. Generación de una clave epsilon-segura K^A y K^B : $CLPU^A$ y $CLPU^B$ aplican un paso de amplificación de privacidad para extraer de $K^{A'}$ y $K^{B'}$ una clave más corta, K^A y K^B , de una longitud aproximada de $(M-t)*N$ bits. Esto se puede hacer usando un hashing aleatorio con una matriz aleatoria, como se explicó anteriormente, o usando cualquier otro mecanismo conocido de amplificación de privacidad.

10

Nótese que el Protocolo 1 es sólo un ejemplo no limitativo de una realización de la invención. Por ejemplo, nótese que un método simplista para implementar el paso de amplificación de privacidad adicional que se aplica a $K^{A'}$ es, sencillamente, que Alice realice la operación XOR bit a bit entre las distintas claves, K_i^A (e igualmente Bob). Además, en general, tampoco es necesario aplicar la amplificación de privacidad en dos pasos diferentes (es decir, para generar primero $K^{A'}$ y $K^{B'}$ y, posteriormente, para generar K^A y K^B a partir de $K^{A'}$ y $K^{B'}$), sino que podría aplicarse en un solo paso. Esto es, se podría generar directamente en un solo paso una clave criptográfica epsilon-segura, K^A y K^B , a partir de todas las claves brutas, R_i^A y R_i^B , juntas. Es importante destacar que el Protocolo 1 ilustra que la amplificación de privacidad puede utilizarse para garantizar el secreto de la clave final, K^A y K^B , en presencia de KGUs deshonestas con una estructura de acceso prescrita.

20

C. Escenario 2: Un método de distribución de claves con unidades clásicas de post-procesado de datos no confiables en una estación criptográfica.

25

En esta sección, se considera un escenario en el que algunas CLPUs de una estación criptográfica pueden no ser confiables, pero las KGUs sí lo son. Una vez más, para mejorar la seguridad, se considera el uso de múltiples CLPUs en una estación criptográfica, como se muestra en la Figura 4. Esta figura muestra dos estaciones criptográficas, A y B, en presencia de un espía Eve (por ejemplo, en los canales que conectan las estaciones criptográficas A y B y también en algunas de las CLPUs de Alice y Bob, lo que hace que dichas CLPUs no sean confiables). Cada estación

30

criptográfica contiene más de una unidad de procesamiento (conocidas como unidades clásicas de post-procesado de datos, CLPUs). Cada CLPU genera porciones de una clave criptográfica, K .

- 5 Si bien el ámbito de protección de la invención es general y se aplica a una estructura adversaria general con CLPUs deshonestas que podrían ser controladas pasiva o activamente por Eve, describiremos aquí con fines ilustrativos el caso sencillo de una estructura adversaria de umbral con un número fijo de CLPUs deshonestas controladas activamente por Eve (sin embargo, la invención puede ser aplicada a otras
- 10 estructuras adversarias más complejas). Más concretamente, consideremos la situación en la que las estaciones criptográficas A y B tienen una KGU confiable cada una y, además, la estación criptográfica A tiene " s " unidades clásicas de post-procesado de datos $CLPU^A_1, CLPU^A_2, \dots, CLPU^A_s$, y la estación criptográfica B tiene " s " unidades clásicas de post-procesado de datos $CLPU^B_1, CLPU^B_2, \dots, CLPU^B_{s'}$.
- 15 Supongamos además que hasta $t < s/3$, $CLPU^A_i$ y hasta $t' < s'/3$ $CLPU^B_j$ podrían ser deshonestas y estar activamente controladas por Eve, con $i=1, \dots, s$, y $j=1, \dots, s'$. Nótese que las distintas CLPUs pueden comprarse a proveedores diferentes. La presente invención permite construir un sistema de generación de claves criptográficas seguro con componentes de vendedores no confiables.

- 20 Como algunas CLPUs son deshonestas (no confiables), la presente invención no les permite acceder a la clave final, K^A y K^B , sino que únicamente se les permite producir algunas porciones de la clave final, K^A y K^B . Por ejemplo, en la Figura 4 indicamos las porciones de K^A generadas por $CLPU^A_i$ (para $i=1, 2, \dots, s$) como K^A_{ixy} y, del mismo modo, indicamos las porciones de K^B generadas por $CLPU^B_j$ (para $j=1, 2, \dots, s'$) como K^B_{jxy} para determinados índices x e y .

- KGU^A está conectada a cada $CLPU^A_i$ a través de un canal clásico (convencional) seguro (denominado "Canal Cl. S. i^A" en la Figura 4), y KGU^B está conectada a cada
- 30 $CLPU^B_j$ a través de un canal clásico (convencional) seguro (denominado "Canal Cl. S. j^B" en la Figura 4). Además, cada par de unidades $CLPU^A_i$ y $CLPU^A_{i'}$, con $i, i'=1, \dots, s$, se conectan entre sí a través de un canal clásico (convencional) seguro (denominado "Canal Cl. S. ii^A" en la Figura 4), y cada par de unidades $CLPU^B_j$ y $CLPU^B_{j'}$, con $j, j'=1, \dots, s'$, se conectan entre sí mediante un canal clásico seguro (denominado "Canal Cl. S. jj^B" en la Figura 4). Asimismo, cada par de unidades $CLPU^A_i$ y $CLPU^B_j$ se
- 35

conectan entre sí a través de un canal clásico (convencional) autenticado (denominado “Canal Cl. A. ij^{AB} ” en la Figura 4). Y, en el caso de un sistema QKD, KGU^A y KGU^B se conectan entre sí a través de un canal cuántico (denominado “Canal C.” en la Figura 4), que es accesible a Eve.

5

En esta situación, se considera un nuevo protocolo de distribución de claves criptográficas en el que KGU^A envía porciones de la clave bruta generada, R^A , a una o más de las $CLPU^A_i$. Del mismo modo, KGU^B envía porciones de la clave bruta, R^B , a una o más de las $CLPU^B_j$. En la Figura 4, S^A_i indica las porciones de R^A que KGU^A envía a $CLPU^A_i$ y S^B_j son las porciones de R^B que KGU^B envía a $CLPU^B_j$. Si la estación criptográfica A y/o B tiene más de una KGU, entonces cada KGU puede enviar porciones de su clave bruta a una o más de las CLPUs de esa estación criptográfica. A continuación, las unidades $CLPU^A_i$ y $CLPU^B_j$ post-procesan la clave bruta, R^A y R^B , en un entorno distribuido actuando sobre las porciones, S^A_i y S^B_j , recibidas y generan porciones, $K^A_{i,1}, K^A_{i,2}, \dots, K^A_{i,n_i}$ y $K^B_{j,1}, K^B_{j,2}, \dots, K^B_{j,m_j}$, de la clave final, K^A y K^B . Aquí n_i indica el número total de porciones de K^A generadas por la unidad $CLPU^A_i$ y m_j indica el número total de porciones de K^B generadas por la unidad $CLPU^B_j$. Una vez más, este post-procesado clásico de datos puede incluir varios pasos como, por ejemplo, la post-selección de datos, la adición de ruido, la estimación de parámetros, la reconciliación de la información y la amplificación de privacidad.

20

Para dividir la clave bruta, R^A y R^B , en porciones, las KGUs podrían utilizar cualquier método conocido, por ejemplo, esquemas de compartición de secretos o esquemas de compartición de secretos verificables. En un esquema de compartición de secretos, la persona o el dispositivo que divide el secreto se denomina distribuidor. En caso de que un distribuidor pudiera ser deshonesto, es importante verificar que los valores de las porciones enviadas por el distribuidor sean consistentes entre sí. Además, en presencia de partes deshonestas, es importante poder garantizar que todas las partes honestas puedan reconstruir el mismo secreto a partir de las porciones recibidas y, además, si el distribuidor es honesto, el secreto reconstruido debe ser igual al distribuido originalmente por el distribuidor. Estas condiciones pueden garantizarse mediante un esquema de compartición de secretos verificable. En un esquema de compartición de secretos verificable, un distribuidor divide cada porción en porciones adicionales o “porciones de una porción” y envía dichas porciones de una porción a varios participantes. Esos participantes pueden entonces verificar la consistencia de

30

35

los valores de dichas porciones de porciones.

En presencia de CLPUs deshonestas, el uso de un esquema de compartición de secretos verificable puede garantizar la consistencia de las porciones distribuidas. Por lo tanto, el uso de un protocolo de reconciliación de la información (que podría incluir un paso de verificación de errores) puede garantizar que la clave final, K^A y K^B , sea correcta.

A continuación, incluimos la implementación paso a paso de un ejemplo de esquema de compartición de secretos verificable propuesto en Maurer, U. Secure multi-party computation made simple. Discrete Appl. Math. 154, 370-381 (2006). Utiliza un esquema de compartición de secretos de tipo umbral (q,q) . Este último sistema podría implementarse, por ejemplo, dividiendo el mensaje X , a ser distribuido, en una suma aleatoria de q porciones X_i , con $i = 1, \dots, q$. Esto podría hacerse, por ejemplo, seleccionando las primeras $q-1$ porciones X_i de X al azar, y luego eligiendo $X_q = X \oplus X_1 \oplus \dots \oplus X_{q-1}$, donde el símbolo $\oplus$ indica una suma en álgebra de módulo 2. Un esquema de compartición de secretos verificable generalmente puede descomponerse en dos protocolos: el protocolo de distribución y el de reconstrucción. El ejemplo de esquema de compartición de secretos verificable presentado a continuación permite dividir un mensaje X entre n partes, y proporciona seguridad en el paradigma de la teoría de la información contra una estructura adversaria activa de umbral con un máximo de $t < n/3$ partes deshonestas. Una vez más, por estructura adversaria (general) entendemos un conjunto de subconjuntos que identifica qué combinaciones de partes podrían estar pasivamente corrompidas, y un conjunto de subconjuntos que identifica qué combinaciones de partes podrían estar activamente corrompidas. Nótese que también existen esquemas de compartición de secretos verificables que son seguros frente a estructuras adversarias generales. Con el fin de simplificar la explicación, aquí se asume una estructura adversaria activa de umbral en la que podría haber un máximo de t partes activamente corrompidas (sin embargo, la invención se puede aplicar a otras estructuras adversarias más complejas). Por lo tanto, el ejemplo del esquema de compartición de secretos verificable descrito anteriormente y cuyos protocolos de distribución y reconstrucción se detallan a continuación es seguro frente a dicha estructura adversaria activa de umbral, pero no siempre es seguro frente a estructuras adversarias generales. Sin embargo, si se desea, puede modificarse para hacerlo seguro frente a una estructura adversaria

general. De hecho, dicha modificación se introdujo en el documento antes citado, Maurer, U. Secure multi-party computation made simple. Discrete Appl. Math. 154, 370-381 (2006).

5 Protocolo de distribución:

1. El distribuidor utiliza un esquema de compartición de secretos de tipo umbral (q, q) para dividir el mensaje X en $q = n! / [(n-t)!t!]$ porciones X_i , con $i = 1, \dots, q$.
- 10 2. Sean $\{\sigma_1, \dots, \sigma_q\}$ todas las $(n-t)$ -combinaciones del conjunto de n partes. Entonces, para cada $i=1, \dots, q$, el distribuidor envía X_i a través de un canal seguro a cada parte del conjunto σ_i . Si una parte no recibe su porción, toma como porción predeterminada, por ejemplo, una secuencia de bits con todos sus componentes iguales a cero.
- 15 3. Todos los pares de partes en σ_i se envían mutuamente sus porciones X_i a través de un canal seguro para comprobar que sus porciones son efectivamente iguales. Si se encuentra una inconsistencia, se quejan utilizando un canal de difusión.
- 20 4. Si se presenta una queja en el conjunto σ_i , el distribuidor difunde X_i a todas las partes y éstas aceptan la porción recibida. Si el distribuidor se niega a difundir X_i a todas las partes (o es incapaz de hacerlo) cuando hay una queja en σ_i , el protocolo aborta.

25

Protocolo de reconstrucción:

1. Todos los pares de partes se envían mutuamente sus porciones a través de un canal clásico autenticado.
- 30 2. Cada parte utiliza como método de decisión una votación por mayoría para reconstruir las porciones $X_i \forall i$, y obtiene a continuación $X = X_1 \oplus \dots \oplus X_q$.

35 La Figura 5 proporciona una representación gráfica de los protocolos de distribución y

reconstrucción explicados anteriormente para el caso de una estructura adversaria de tipo umbral en la que como mucho 1 de 4 partes podría ser deshonestas (es decir, $n=4$ y $t=1$). En el esquema de compartición de secretos verificable mostrado en la Figura 5, el mensaje, X , se distribuye entre cuatro partes P_i , con $i=1, \dots, 4$, y proporciona
 5 seguridad en el paradigma de la teoría de la información contra una estructura adversaria activa de tipo umbral con un máximo de $t=1$ partes deshonestas. En la figura, $X_{i=\{x,y,z\}}$ indica las porciones X_x , X_y y X_z de X . En el protocolo de distribución todos los canales son canales clásicos (convencionales) seguros, mientras que en el protocolo de reconstrucción los canales son canales clásicos (convencionales)
 10 autenticados.

En el caso que se muestra en la figura, se definen los conjuntos $\sigma_1=\{P_2, P_3, P_4\}$, $\sigma_2=\{P_1, P_3, P_4\}$, $\sigma_3=\{P_1, P_2, P_4\}$ y $\sigma_4=\{P_1, P_2, P_3\}$. Como $q=n!/[(n-t)!t!]=4$, el mensaje X se ha dividido en 4 porciones, X_1 , X_2 , X_3 y X_4 . A continuación, el distribuidor envía X_i a través
 15 de un canal seguro a todas las partes en σ_i . Por ejemplo, envía X_1 a P_2 , P_3 y P_4 , y lo mismo para las demás porciones.

Nótese que el canal de difusión que se requiere para realizar los pasos 3 y 4 del protocolo de distribución no tiene que ser un canal físico, sino que podría ser un canal
 20 simulado.

El esquema de compartición de secretos verificable (en inglés, verifiable secret sharing scheme) presentado anteriormente es un ejemplo no limitativo, y podría utilizarse en su lugar cualquier otro mecanismo de compartición de secretos verificable conocido o
 25 todavía por desarrollar. Por ejemplo, si el número n de partes es elevado, existen otros esquemas eficientes de compartición de secretos verificables, o si hay un canal físico de difusión disponible existen esquemas de compartición de secretos verificables que pueden garantizar seguridad siempre que haya una mayoría de partes honestas (véase, por ejemplo, Rabin, T. and Ben-Or, M. Verifiable secret sharing and multiparty protocols with honest majority. Proc. 21th Annual ACM Symposium on Theory of Computing (STOC'89) 73-85 (ACM, New York, NY, USA, 1989)).
 30

Para implementar varios de los protocolos clásicos de post-procesado de datos que se requieren para generar una clave criptográfica segura, las CLPUs podrían necesitar
 35 generar números aleatorios entre partes que no son confiables entre sí. A continuación

presentamos la implementación paso a paso de un ejemplo de un protocolo que podría ser utilizado para resolver esta tarea. Se obtiene directamente a partir de los esquemas de compartición de secretos verificables y puede generar una secuencia común completamente aleatoria de l -bits entre n partes cuando hasta $t < n/3$ de ellas pudieran ser deshonestas y estar activamente controladas por Eve. Para mayor comodidad, lo denominamos protocolo RBS, del término anglosajón "Random Bit String" (o secuencia de bits aleatorios).

Protocolo RBS:

- 10 1. Digamos que cada una de las primeras $t+1$ partes produce localmente una secuencia aleatoria de l -bits r_i y la envía a todas las demás partes usando el protocolo de distribución de un esquema de compartición de secretos verificable.
- 15 2. Cada una de las partes utiliza un canal difusión para confirmar que ha recibido las porciones de las primeras $t+1$ partes. De lo contrario, el protocolo aborta.
3. Todas las partes utilizan el protocolo de reconstrucción de un esquema de compartición de secretos verificable para obtener r_i para todos los $i = 1, \dots, t+1$.
- 20 Posteriormente, cada parte calcula localmente $r = r_1 \oplus \dots \oplus r_{t+1}$.

Como en el caso anterior, nótese que el canal de difusión requerido en el paso 2 del protocolo RBS podría realizarse con un canal de difusión simulado. Además, nótese que para generar números aleatorios entre partes mutuamente no confiables, de manera que éstos sean seguros frente a estructuras adversarias generales, se podría simplemente utilizar en el protocolo RBS descrito arriba un protocolo de distribución (en el paso 1) y un protocolo de reconstrucción (en el paso 3) de un esquema de compartición de secretos verificable que sea seguro frente a estructuras adversarias generales, como por ejemplo el que se propuso en Maurer, U. Secure multi-party computation made simple. Discrete Appl. Math. 154, 370-381 (2006).

Para ilustrar la presente invención en este escenario, a continuación describimos dos protocolos (véanse los Protocolos 2 y 3, a continuación) que podrían utilizarse para lograr distribuir claves criptográficas seguras con CLPUs no confiables. Son dos ejemplos de realizaciones de la presente invención. Para facilitar su descripción,

consideraremos en estos dos ejemplos que el número de CLPUs de la estación criptográfica A es igual al de B, es decir, $s=s'$, aunque, por supuesto, esto no es una condición necesaria y el número de CLPUs de cada estación criptográfica puede ser diferente.

5

El primer ejemplo (véase el Protocolo 2 a continuación) puede descomponerse en dos pasos conceptuales principales. En un primer paso, KGU^A y KGU^B realizan s sesiones independientes de generación de claves criptográficas, y cada una de estas sesiones se realiza con un par de unidades $CLPU_i^A$ y $CLPU_i^B$ diferentes, con $i=1, \dots, s$. El objetivo de cada sesión es generar, por ejemplo, una clave (epsilon/s)-segura, K_i^A y K_i^B , o el símbolo de abortar $\perp_i$. Para facilitar la ilustración, consideremos que la longitud de cada clave K_i^A y K_i^B es N bits para todos los i . Si, por ejemplo, el par $CLPU_i^A$ y $CLPU_i^B$ es deshonesto, entonces se tiene que K_i^A y K_i^B podrían estar comprometidas y ser conocidas por Eve. Un par $CLPU_i^A$ y $CLPU_i^B$ es deshonesto si al menos una de sus unidades es deshonesto. A continuación, en un segundo paso, las claves K_i^A y K_i^B se concatenan para formar $K^A = [K_1^A, K_2^A, \dots, K_M^A]$ y $K^B = [K_1^B, K_2^B, \dots, K_M^B]$, donde M indica el número de claves K_i^A y K_i^B que son diferentes del símbolo de abortar, y las CLPUs aplican un protocolo de verificación de errores y un paso de amplificación de privacidad a K^A y K^B . Aquí viene una contribución fundamental de esta realización de la invención: este segundo paso se realiza en un entorno distribuido actuando sólo sobre porciones de K^A y K^B . Nótese que esto es posible porque todas las técnicas de post-procesado de datos descritas anteriormente suelen ser de naturaleza "lineal" (esto es, normalmente implican aplicar funciones simples de álgebra lineal como XOR de bits y multiplicaciones de matrices) y por lo tanto pueden ser fácilmente implementadas actuando sólo sobre porciones de K^A y K^B . En ciertos pasos, las CLPUs pueden necesitar información sobre porciones que están en manos de otras CLPUs. Para obtener esta información, podrían utilizar, por ejemplo, el protocolo de reconstrucción de un esquema de compartición de secretos verificable. Esto es, todas las unidades que tienen la información se limitan a enviarla a la unidad que la requiere y ésta puede emplear un método de decisión basado en votación por mayoría para discriminar qué información es la correcta. A continuación describimos el protocolo con más detalle:

Protocolo 2 (esto es sólo una posible realización, y no todos los pasos citados son esenciales y obligatorios en todas las realizaciones de la presente invención):

1. Generación de K^A_i y K^B_i : KGU^A y KGU^B realizan s sesiones independientes de generación de claves criptográficas, cada una de estas sesiones se realiza con un par de unidades diferentes $CLPU^A_i$ y $CLPU^B_i$, con $i=1, \dots, s$. Para esto utilizan el "Canal C.", el "Canal Cl. S. i^A ", el "Canal Cl. S. i^B ", y el "Canal Cl. A. ij^{AB} " mostrados en la Figura 4. El resultado de cada sesión de generación de claves serán dos secuencias de bits, K^A_i y K^B_i , que se supone que son una clave (ϵ_{cor}/s)-correcta y (ϵ_{sec}/s)-secreta (con $\epsilon_{cor}+\epsilon_{sec} \leq \epsilon$), o el símbolo que indica aborto $\perp_i$.
2. Distribución de porciones de K^A_i y K^B_i : cada $CLPU^A_i$ divide K^A_i en porciones y las envía a las otras CLPUs de la estación criptográfica A siguiendo, por ejemplo, el protocolo de distribución de un esquema de compartición de secretos verificable y utilizando los canales clásicos seguros "Canal Cl. S. il^A " mostrados en la Figura 4, y todas las $CLPU^A_l$, con $l=1, \dots, s$ y $l \neq i$, se confirman mutuamente que han recibido sus porciones. Sea K'^{A}_{ij} la j^a porción de K^A_i recibida por $CLPU^A_l$. Del mismo modo, las unidades $CLPU^B_i$ actúan de forma similar con K^B_i . Sea K'^{B}_{ij} la j^a porción de K^B_i recibida por $CLPU^B_l$.
3. Generación de K''^{A}_{ij} y K''^{B}_{ij} : cada $CLPU^A_l$, con $l=1, \dots, s$, define localmente las secuencias de bits $K''^{A}_{ij} = [0_1, \dots, 0_{i-1}, K'^{A}_{ij}, 0_{i+1}, \dots, 0_M]$, donde 0_i , con $i=1, \dots, M$, representa el vector cero de N bits, y M es el número de pares $CLPU^A_i$ y $CLPU^B_i$ que no produjeron el símbolo de abortar $\perp_i$. Del mismo modo, las unidades $CLPU^B_l$ actúan de forma similar y obtienen K''^{B}_{ij} .
4. Verificación de errores: Las $CLPU^A_l$, con $l=1, \dots, s$, usan por ejemplo el protocolo RBS para seleccionar una secuencia aleatoria de bits y con ella eligen al azar una función hash 2-universal entre un conjunto preestablecido de funciones hash 2-universales. Luego, cada una de ellas calcula localmente un hash $h^A_{ij} = \text{hash}(K''^{A}_{ij})$ de una longitud digamos $\lceil \log_2(4/\epsilon_{cor}) \rceil$ bits para todas sus secuencias de bits K''^{A}_{ij} , y por ejemplo las primeras $2t+1$ $CLPU^A_l$ envían la función hash elegida a todas las $CLPU^B_{l'}$ a través de los canales clásicos autenticados "Canal Cl. A. $l'l^{AB}$ " mostrados en la Figura 4, con $l'=1, \dots, s$. Cada $CLPU^B_{l'}$ reconstruye localmente la función hash usando un método de decisión

basado en votación por mayoría y obtiene $h_{l'ij}^B = \text{hash}(K_{l'ij}^{''B})$ para todas sus secuencias de bits $K_{l'ij}^{''B}$. A continuación, todas las $CLPU_{l'}^A$ y $CLPU_{l'}^B$ usan el protocolo de reconstrucción de un esquema de compartición de secretos verificable para obtener $h^A = h_{11}^A \oplus \dots \oplus h_{Mq}^A$ y $h^B = h_{11}^B \oplus \dots \oplus h_{Mq}^B$, donde $q = s! / [(s-t)!t!]$ es el número de porciones de cada $K_{l'}^A$ y $K_{l'}^B$. Para esto, se envían
5 unas a otras las secuencias de bits $h_{l'ij}^A$ y $h_{l'ij}^B$ a través de canales clásicos autenticados, y cada uno de ellos usa un método de decisión basado en votación por mayoría para obtener $h_{l'ij}^A$ y $h_{l'ij}^B$ a partir de $h_{l'ij}^A$ y $h_{l'ij}^B$. Finalmente, cada $CLPU_{l'}^A$ y $CLPU_{l'}^B$ comprueba localmente que $h^A = h^B$. Si no son iguales, el
10 resultado es el símbolo de abortar. Si son iguales, pasan al paso 5. Este paso de verificación de errores garantiza que $K^{''A} = K_{11}^{''A} \oplus \dots \oplus K_{Mq}^{''A}$ y $K^{''B} = K_{11}^{''B} \oplus \dots \oplus K_{Mq}^{''B}$ son iguales excepto con probabilidad, como mucho, ϵ_{cor} , donde $K_{l'ij}^{''A}$ indica la secuencia de bits que se obtendría de $K_{l'ij}^A$ usando votación por mayoría, y la definición de $K_{l'ij}^{''B}$ es análoga.

15

5. Amplificación de privacidad: las $CLPU_{l'}^A$ usan por ejemplo el protocolo RBS para seleccionar aleatoriamente una función hash 2-universal, hashPA , entre un conjunto preestablecido de ellas. Calculan $K_{l'ij}^A = \text{hashPA}(K_{l'ij}^{''A})$, y, por ejemplo, las $2t+1$ primeras $CLPU_{l'}^A$ envían la función hashPA a todas las
20 $CLPU_{l'}^B$ a través de los canales clásicos autenticados "A.C. channel l'^{AB} " mostrados en la Figura 4, con $l' = 1, \dots, s$. Luego, las $CLPU_{l'}^B$ usan un método de decisión basado en votación por mayoría para determinar hashPA a partir de la información recibida y calculan $K_{l'ij}^B = \text{hashPA}(K_{l'ij}^{''B})$. La función hashPA transforma las secuencias de $(M \cdot N)$ bits $K_{l'ij}^{''A}$ y $K_{l'ij}^{''B}$ en dos secuencias de bits más cortas, $K_{l'ij}^A$ y $K_{l'ij}^B$ respectivamente, de un tamaño aproximado de
25 $(M - 2^*t) \cdot N - \lceil \log_2(4/\epsilon_{cor}) \rceil$ bits. La razón para sustraer $2^*t \cdot N$ bits es que, en el peor de los casos, la presencia de t partes deshonestas $CLPU_{l'}^A$ y t partes independientes deshonestas $CLPU_{l'}^B$ podría resultar en 2^*t pares deshonestos $CLPU_{l'}^A$ y $CLPU_{l'}^B$, con $l' = 1, \dots, s$, y, por lo tanto, $2^*t \cdot N$ bits de $K^{''A}$ podrían estar
30 comprometidos.

Si $t < M_{l'}^A/3$ y $t < M_{l'}^B/3$ para todos los $l' = 1, \dots, M$, donde $M_{l'}^A$ indica el número de $CLPU_{l'}^A$ que no producen el símbolo de abortar, sino que generan porciones post-procesadas, $K_{l'ij}^A$, a partir de $K_{l'ij}^A$, y $M_{l'}^B$ indica el número de $CLPU_{l'}^B$ que no producen el símbolo de
35 abortar, sino que generan porciones post-procesadas, $K_{l'ij}^B$, a partir de $K_{l'ij}^B$, entonces

las secuencias de bits K_{ij}^A y K_{ij}^B producidas por las unidades $CLPU^A_i$ y $CLPU^B_i$ en el paso 5 del Protocolo 2 son porciones de una clave final epsilon-segura, K^A y K^B . Como se mencionó anteriormente, en este escenario a las unidades $CLPU^A_i$ y $CLPU^B_i$ sólo se les permite producir porciones de K^A y K^B . Por ejemplo, un laboratorio seguro ubicado en la estación criptográfica A podría usar votación por mayoría para obtener K_{ij}^A a partir de las secuencias K_{ij}^A . Del mismo modo, un laboratorio seguro ubicado en la estación criptográfica B podría usar votación por mayoría para obtener K_{ij}^B a partir de las secuencias K_{ij}^B . De este modo, la clave final podría obtenerse como $K^A = K_{11}^A \oplus \dots \oplus K_{Mq}^A$ y $K^B = K_{11}^B \oplus \dots \oplus K_{Mq}^B$.

10

A continuación presentamos un segundo ejemplo de protocolo que puede proporcionar una distribución segura de claves criptográficas con unidades clásicas de post-procesado de datos no confiables (véase el Protocolo 3, a continuación). Este protocolo presenta dos ventajas principales con respecto al Protocolo 2. En primer lugar, no requiere ejecutar s sesiones independientes de generación de claves, sino que puede generar una clave criptográfica epsilon-segura a partir de la ejecución una única sesión de generación de claves. Y, en segundo lugar, es más eficiente en términos de la tasa de generación de clave secreta, ya que proporciona una tasa de generación de clave secreta que podría ser en torno a $s/(s-2^*t)$ veces mayor que la proporcionada por el Protocolo 2. Si bien el ámbito de protección de la presente invención es general, para facilitar la descripción del Protocolo 3, consideraremos a continuación que el procedimiento de generación de clave criptográfica no usa una post-selección aleatoria de datos de la clave bruta (no obstante, la invención puede aplicarse a otros casos más complejos en los que se lleva a cabo una post-selección aleatoria de datos de la clave bruta). Además, para facilitar la ilustración, se supondrá que el protocolo clásico de post-procesado de datos no estima el valor de la QBER real, sino que utiliza un valor de QBER preestablecido para el protocolo de corrección de errores, seguido de un paso de verificación de errores. El Protocolo 3 que se indica a continuación puede descomponerse en dos pasos conceptuales principales. En un primer paso, KGU^A y KGU^B generan una clave bruta, R^A y R^B respectivamente. A continuación viene un paso clave en esta realización de la invención. KGU^A divide R^A en porciones utilizando, por ejemplo, el protocolo de distribución de un esquema de compartición de secretos verificable y luego envía esas porciones a las unidades $CLPU^A_i$. Este paso se ilustra en la Figura 4, donde S_i^A representa las porciones de R^A que se envían a $CLPU^A_i$ a través del canal seguro "Canal Cl. S. i^a". Del mismo modo,

35

se envía R^B a las unidades $CLPU^B_i$ dividida en porciones. Entonces, en un segundo paso, cada $CLPU^A_i$ y $CLPU^B_i$ con $i, i'=1, \dots, s$, aplica el protocolo clásico de post-procesado de datos a las porciones recibidas. Esto es, cada CLPU puede aplicar una post-selección de datos, añadir ruido, realiar la estimación de parámetros, la
 5 reconciliación de la información y la amplificación de privacidad en un entorno distribuido actuando directamente sobre las porciones recibidas. Como en el caso de los pasos 4 y 5 del Protocolo 2, esto es posible porque todas estas técnicas de post-procesado de datos emplean funciones simples de álgebra lineal y por lo tanto son fácilmente implementables actuando sólo sobre porciones. A continuación describimos
 10 los pasos del Protocolo 3 con más detalle:

Protocolo 3 (esto es solamente una posible realización, y no todos los pasos citados son esenciales y obligatorios en todas las realizaciones de la presente invención):

15 1. Distribución de R^A y R^B : KGU^A y KGU^B generan primero una clave bruta, R^A y R^B . A continuación, KGU^A utiliza el protocolo de distribución de un esquema de compartición de secretos verificable para crear $q=s!/[(s-t)!t!]$ porciones de R^A y luego las distribuye entre las $CLPU^A_i$, con $i = 1, \dots, s$, utilizando los canales clásicos seguros "Canal Cl. S. i^A " mostrados en la
 20 Figura 4. Por ejemplo, para ello KGU^A puede utilizar el protocolo de distribución descrito anteriormente. A su vez, KGU^B hace lo mismo con R^B y las unidades $CLPU^B_i$ con $i=1, \dots, s$. Sea K'^A_{ij} la j^a porción de R^A recibida por $CLPU^A_i$ y sea K'^B_{ij} la j^a porción de R^B recibida por $CLPU^B_i$ ($j=1, \dots, q$). El conjunto de porciones K'^A_{ij} recibidas por $CLPU^A_i$ y el conjunto de porciones
 25 K'^B_{ij} recibidas por $CLPU^B_i$ están representadas en la Figura 4 como S^A_i y S^B_i respectivamente.

2. Post-selección de datos: cada $CLPU^A_i$ para todos los i extrae de las porciones K'^A_{ij} recibidas dos secuencias de bits: $K'^A_{ij, \text{key}}$ y $K'^A_{ij, \text{est}}$. La primera
 30 secuencia de bits se utilizará para generar la clave (transformada en una clave segura gracias al post-procesado de datos) y la segunda secuencia de bits se utilizará para la estimación de parámetros. A su vez, cada $CLPU^B_i$ para todas las i hace lo mismo con K'^B_{ij} para obtener $K'^A_{ij, \text{key}}$ y $K'^B_{ij, \text{est}}$.

35 3. Estimación de parámetros: todas las $CLPU^A_i$ y $CLPU^B_i$, con $i, i'=1, \dots, s$, usan

el protocolo de reconstrucción de un esquema de compartición de secretos verificable para obtener tanto K_{est}^A como K_{est}^B , que son las partes de R^A y R^B que se utilizan para la estimación de parámetros. Para ello, se envían sus porciones $K_{ij,est}^A$ y $K_{ij,est}^B$ a través de canales clásicos autenticados. Esto es, cada unidad $CLPU_i^A$ recibe las porciones $K_{ij,est}^A$ y $K_{ij,est}^B$ que están en manos de todas las demás CLPUs. Del mismo modo, cada unidad $CLPU_i^B$ recibe las porciones $K_{ij,est}^A$ y $K_{ij,est}^B$ que están en manos de todas las demás CLPUs. A continuación, cada una de ellas utiliza votación por mayoría para obtener tanto $K_{j,est}^A$ como $K_{j,est}^B$ a partir de $K_{ij,est}^A$ y $K_{ij,est}^B$ para todos los $j=1, \dots, q$. Posteriormente, calculan $K_{est}^A = K_{1,est}^A \oplus \dots \oplus K_{q,est}^A$ y $K_{est}^B = K_{1,est}^B \oplus \dots \oplus K_{q,est}^B$. Con esta información, cada $CLPU_i^A$ y $CLPU_i^B$ realiza localmente el paso de estimación de parámetros del protocolo (por ejemplo, calculan la tasa cuántica de error de fase). Si la tasa cuántica de error de fase es demasiado alta, se genera el símbolo de abortar.

4. Corrección de errores: las $CLPU_i^A$ y $CLPU_i^B$ ejecutan un protocolo de corrección de errores en las partes de R^A y R^B que se utilizan para la extracción de la clave, K_{key}^A y K_{key}^B . Este proceso se realiza actuando sobre sus porciones $K_{ij,key}^A$ y $K_{ij,key}^B$ respectivamente. Para ello, cada $CLPU_i^A$ aplica ciertas matrices M_{EC} a $K_{ij,key}^A$ para obtener $s_{ij}^A = M_{EC} * K_{ij,key}^A$. Del mismo modo, cada $CLPU_i^B$ aplica M_{EC} a $K_{ij,key}^B$ para obtener $s_{ij}^B = M_{EC} * K_{ij,key}^B$. Después, $CLPU_i^A$ y $CLPU_i^B$ utilizan el protocolo de reconstrucción de un esquema de compartición de secretos verificable para garantizar que todas las $CLPU_i^B$ puedan obtener $s^A = M_{EC} * K_{key}^A$ y $s^B = M_{EC} * K_{key}^B$. Para ello, todas las $CLPU_i^A$ envían a todas las $CLPU_i^B$ las secuencias de bits s_{ij}^A a través del canal clásico autenticado "Canal Cl. A. ii'AB" mostrado en la Figura 4. Además, todas las $CLPU_i^B$ de la estación criptográfica B se envían mutuamente las secuencias de bits s_{ij}^B a través de canales clásicos autenticados. Luego, cada $CLPU_i^B$ usa votación por mayoría para reconstruir localmente s_j^A y s_j^B , para todos los j , a partir de s_{ij}^A y s_{ij}^B . Por último, obtienen $s^A = s_1^A \oplus \dots \oplus s_q^A$ y $s^B = s_1^B \oplus \dots \oplus s_q^B$. A continuación, las CLPUs de la estación criptográfica B corrigen K_{key}^B . Para esto, digamos que todas las $CLPU_i^B$ que tienen por ejemplo la j^a porción $K_{ij,key}^B$ de K_{key}^B para un índice prefijado $j=1, \dots, q$, realizan la operación NOT o de negación

- lógica de ciertos bits de esta porción dependiendo del valor de s^A y s^B . Todo este proceso se repite hasta que finaliza el protocolo de corrección de errores. El final del protocolo de corrección de errores viene determinado por el protocolo concreto de corrección de errores implementado, así como
- 5 por el valor de la QBER. Sean $K_{ij,key}^A$ y $K_{ij,key}^B$ las porciones $K_{ij,key}^{A'}$ y $K_{ij,key}^{B'}$ después de la corrección de errores, y sean $leak_{EC}$ bits la información de síndrome. Aquí la información de síndrome hace referencia a la información que las CLPUs de una estación criptográfica envían a las CLPUs de la otra estación criptográfica (y viceversa) durante el protocolo de corrección de
- 10 errores. Por ejemplo, en cada iteración del protocolo de corrección de errores las CLPUs de la estación criptográfica A envían $|s^A|$ bits de información de síndrome a las CLPUs de la estación criptográfica B.
5. Verificación de errores: las $CLPU^A_i$, con $i=1,\dots,s$, usan el protocolo RBS para seleccionar aleatoriamente una función hash 2-universal. Luego, cada una de ellas calcula localmente un hash $h_{ij}^A = \text{hash}(K_{ij,key}^A)$ de una longitud digamos de $\lceil \log_2(4/\epsilon_{cor}) \rceil$ bits y por ejemplo las primeras $2t+1$ $CLPU^A_i$ envían la función hash a todas las $CLPU^B_{i'}$, con $i'=1,\dots,s$, a través de los canales clásicos autenticados "Canal Cl. A. $i \rightarrow AB$ " mostrados en la
- 15 Figura 4. Cada $CLPU^B_{i'}$ reconstruye localmente la función hash mediante votación por mayoría y obtiene $h_{ij}^B = \text{hash}(K_{ij,key}^B)$. A continuación, todas las $CLPU^A_i$ y $CLPU^B_{i'}$ utilizan el protocolo de reconstrucción de un esquema de compartición de secretos verificable para obtener tanto $h^A = h_{11}^A \oplus \dots \oplus h_{1q}^A$ como $h^B = h_{11}^B \oplus \dots \oplus h_{1q}^B$ a partir de h_{ij}^A and h_{ij}^B . Para ello, se envían mutuamente h_{ij}^A y h_{ij}^B a través de canales clásicos autenticados. Esto es,
- 20 cada unidad $CLPU^A_{i'}$ recibe las porciones h_{ij}^A y h_{ij}^B que están en manos de todas las demás CLPUs. Del mismo modo, cada unidad $CLPU^B_{i'}$ recibe las porciones h_{ij}^A y h_{ij}^B que están en manos de todas las demás CLPUs. Luego, cada una de ellas utiliza votación por mayoría para determinar tanto h_j^A como h_j^B para todos los j a partir de h_{ij}^A y h_{ij}^B . Por último, cada una de ellas comprueba localmente si $h^A = h^B$ o no. Si no son iguales, generan el símbolo de abortar. Si son iguales, avanzan al paso 6. Este paso de verificación de errores garantiza que $K_{key}^{A'} = K_{1,key}^A \oplus \dots \oplus K_{q,key}^A$ y $K_{key}^{B'} = K_{1,key}^B \oplus \dots \oplus K_{q,key}^B$ sean iguales excepto con probabilidad, como mucho, ϵ_{cor} , donde
- 25 $K_{j,key}^A$ y $K_{j,key}^B$ indican, respectivamente, las secuencias de bits que se
- 30
- 35

obtendrían de $K_{ij,key}^A$ y $K_{ij,key}^B$ utilizando votación por mayoría.

- 5 6. Amplificación de privacidad: las $CLPU_i^A$ usan por ejemplo el protocolo RBS para seleccionar aleatoriamente una función hash 2-universal, hashPA. Luego, calculan $K_{ij}^A = \text{hashPA}(K_{ij,key}^A)$, y por ejemplo las $2t+1$ primeras $CLPU_i^A$ envían la función hashPA a todas las $CLPU_{i'}^B$ a través de los canales clásicos autenticados "Canal Cl. A. ii'^{AB} " mostrados en la Figura 4, con $i'=1, \dots, s$. A continuación, las $CLPU_{i'}^B$ usan votación por mayoría para determinar la función hashPA a partir de la información recibida y calculan $K_{ij}^B = \text{hashPA}(K_{ij,key}^B)$. La función hashPA elimina la información parcial que Eve podría tener sobre la clave final, K^A y K^B , lo cual incluye la información de síndrome leak_{EC} revelada durante el protocolo de reconciliación de la información, el valor de hash revelado durante el protocolo de verificación de errores, así como la información adicional que Eve podría tener sobre la clave y que puede calcularse a partir de la tasa cuántica de error de fase.
- 10 15

Nótese que si el número de unidades deshonestas $CLPU_i^A$ satisface $t < M_A/3$ y el número de unidades deshonestas $CLPU_{i'}^B$ satisface $t < M_B/3$, donde M_A es el número de unidades $CLPU_i^A$ que no abortan y M_B es el número de unidades $CLPU_{i'}^B$ que no abortan, entonces a partir de las secuencias de bits K_{ij}^A y K_{ij}^B producidas en el paso 6 del Protocolo 3 es posible reconstruir una clave epsilon-segura, K^A y K^B . Para ello, por ejemplo un laboratorio seguro situado en la estación criptográfica A podría utilizar votación por mayoría para obtener K_j^A a partir de K_{ij}^A para todos los $j=1, \dots, q$. Del mismo modo, por ejemplo, un laboratorio seguro situado en la estación criptográfica B podría utilizar votación por mayoría para obtener K_j^B a partir de K_{ij}^B para todos los $j=1, \dots, q$. Finalmente se tiene que $K^A = K_1^A \oplus \dots \oplus K_q^A$ y $K^B = K_1^B \oplus \dots \oplus K_q^B$.

20 25

Nótese que los Protocolos 2 y 3 son sólo ejemplos no limitativos de realizaciones de la invención. Asimismo, nótese que estos protocolos pueden ser modificados para garantizar seguridad frente a estructuras adversarias generales. Para ello, se podría básicamente sustituir el esquema de compartición de secretos verificable descrito arriba, y que es seguro frente a estructuras adversarias de tipo umbral, por otro conocido robusto frente a estructuras adversarias generales (véase, por ejemplo, Maurer, U. Secure multi-party computation made simple. Discrete Appl. Math. 154, 370-381 (2006)), y, además, el método para anunciar las funciones hash y hashPA

30 35

puede depender ahora de la estructura adversaria. Es importante señalar que los Protocolos 2 y 3 destacan una contribución fundamental de la presente invención, que es que el uso de esquemas de compartición de secretos podría garantizar la seguridad de los sistemas de distribución de claves criptográficas en presencia de CLPUs deshonestas.

D. Escenario 3: Un método de distribución de claves con unidades de generación de claves no confiables y unidades clásicas de post-procesado de datos no confiables en una estación criptográfica

Aquí se considera el escenario más general (realización preferente de la invención), en el que tanto las KGUs como las CLPUs de una estación criptográfica pueden no ser confiables. Una vez más, para garantizar la seguridad se utilizan múltiples CLPUs y KGUs en cada estación criptográfica, como se muestra en la Figura 6, donde hay dos estaciones criptográficas, A y B, en presencia de un espía, Eve, y cada estación criptográfica contiene más de una unidad de generación de claves criptográficas, KGU, y más de una unidad clásica de post-procesado de datos, CLPU. Como en los escenarios anteriores, con fines ilustrativos describiremos a continuación la situación más sencilla de una estructura adversaria activa de tipo umbral con un número fijo de KGUs y CLPUs deshonestas, todas controladas activamente por Eve. Más concretamente, consideremos la situación en la que la estación criptográfica A tiene n unidades $KGU^A_1, KGU^A_2, \dots, KGU^A_n$, así como s unidades $CLPU^A_1, CLPU^A_2, \dots, CLPU^A_s$, y la estación criptográfica B tiene n unidades $KGU^B_1, KGU^B_2, \dots, KGU^B_n$ y s' unidades $CLPU^B_1, CLPU^B_2, \dots, CLPU^B_{s'}$ (aquí se supone que las estaciones A y B tienen el mismo número de KGUs, aunque esto es sólo un ejemplo ilustrativo y pueden tener un número diferente de KGUs). Consideremos también que hasta $t < s/3$ unidades $CLPU^A_i$, hasta $t' < s'/3$ unidades $CLPU^B_j$ y hasta $t'' < n$ pares de unidades de generación de claves KGU^A_i y KGU^B_i con $i=1, \dots, n$, podrían ser deshonestos (no confiables) y estar controlados activamente por Eve. Un par de unidades KGU^A_i y KGU^B_i es deshonesto si al menos una de sus unidades es deshonestas. Esto es sólo un ejemplo no limitativo y, por supuesto, la invención puede ser aplicada a cualquier estructura adversaria general.

Una vez más, dado que algunas CLPUs son deshonestas, a ninguna de ellas se le permite acceder a la clave criptográfica final, K^A y K^B , sino que únicamente pueden

- producir porciones de ésta. Por ejemplo, en la Figura 6, las porciones de K^A que produce $CLPU^A_l$ (para $l=1,2,\dots,s$) se denotan como K^A_{lx} para determinados índices x e y , y del mismo modo las porciones de K^B que produce $CLPU^B_j$ (para $j=1,2,\dots,s'$) se denotan como K^B_{jy} . También, como en los casos anteriores, consideramos que cada
- 5 KGU^A_l está conectada a cada $CLPU^A_l$ a través de un canal de comunicaciones clásico (convencional) seguro (indicado como "Canal Cl. S. il^A " en la Figura 6), con $i=1,\dots,n$ y $l=1,\dots,s$, cada KGU^B_j está conectada a cada $CLPU^B_j$ a través de un canal de comunicaciones clásico (convencional) seguro (indicado como "Canal Cl. S. ij^B " en la Figura 6), con $j=1,\dots,s'$, cada par de unidades $CLPU^A_l$ y $CLPU^A_{l'}$, con $l,l'=1,\dots,s$, está
- 10 conectado entre sí a través de un canal de comunicaciones clásico (convencional) seguro (indicado como "Canal Cl. S. ll'^A " en la Figura 6), cada par de unidades $CLPU^B_j$ y $CLPU^B_{j'}$, con $j,j'=1,\dots,s'$, está conectado entre sí a través de un canal de comunicaciones clásico (convencional) seguro (indicado como "Canal Cl. S. jj'^B " en la Figura 6), cada par de unidades $CLPU^A_l$ y $CLPU^B_j$ está conectado entre sí a través de
- 15 un canal de comunicaciones clásico (convencional) autenticado (indicado como "Canal Cl. A. lj^{AB} " en la Figura 6), y, en los sistemas QKD, cada par de unidades KGU^A_l y KGU^B_j está conectado entre sí a través de un canal cuántico (indicado como "Canal C. i " en la Figura 6 con $i=1,\dots,n$), que es accesible a Eve.
- 20 Para ilustrar la invención, a continuación se describe un ejemplo de un protocolo (véase Protocolo 4) que podría ser utilizado para lograr distribución segura de claves criptográficas en este escenario según la presente invención. Para facilitar la descripción del protocolo, en este ejemplo consideramos que el número de CLPUs de la estación criptográfica A es igual al de B, es decir, $s=s'$ (esto es sólo un ejemplo a
- 25 título ilustrativo y las estaciones criptográficas pueden tener un número diferente de CLPUs). El Protocolo 4 puede descomponerse en dos pasos conceptuales principales. En un primer paso, cada par de unidades KGU^A_l y KGU^B_l genera una clave bruta y luego implementan, junto con las unidades clásicas de post-procesado de datos $CLPU^A_l$ y $CLPU^B_l$, el Protocolo 3 descrito anteriormente. Como resultado, las unidades
- 30 $CLPU^A_l$ y $CLPU^B_l$, con $l=1,\dots,s$, obtienen porciones de una clave (epsilon/n)-segura, K^A_l y K^B_l ($l=1,\dots,n$), o el símbolo de abortar $\perp_l$. En concreto, sea K'^A_{lij} la j^a porción de K^A_l recibida por la $CLPU^A_l$, y del mismo modo sea K'^B_{lij} la j^a porción de K^B_l recibida por la $CLPU^B_l$. Además, para simplificar la discusión (aunque no con fines limitativos), consideremos por ejemplo que la longitud de todas las claves K^A_l y K^B_l es N bits para
- 35 todos los l . A continuación, las CLPUs proceden de forma similar al Protocolo 2. Esto

es, las claves K^A_i y K^B_i se concatenan para formar $K^A=[K^A_1, K^A_2, \dots, K^A_M]$ y $K^B=[K^B_1, K^B_2, \dots, K^B_M]$, donde M indica el número de claves K^A_i y K^B_i que son diferentes del símbolo de abortar, y las CLPUs aplican un paso de amplificación de privacidad a K^A y K^B . Este paso de amplificación de la privacidad elimina la información que los pares deshonestos de KGUs podrían filtrar a Eve acerca de, por ejemplo, K^A . Este segundo paso se realiza en un escenario distribuido actuando sólo sobre las porciones $K^{A_{ij}}$ y $K^{B_{ij}}$. A continuación se describe el protocolo con más detalle:

Protocolo 4 (esto es solamente una posible realización, y no todos los pasos citados son esenciales y obligatorios en todas las realizaciones de la presente invención):

1. Generación y distribución de porciones de K^A_i y K^B_i : cada par KGU^A_i y KGU^B_i , con $i=1, \dots, n$, genera una clave bruta y luego implementa, junto con las unidades clásicas de post-procesado de datos $CLPU^A_i$ y $CLPU^B_i$, todos los pasos del Protocolo 3 descrito anteriormente. Como resultado, las unidades $CLPU^A_i$ y $CLPU^B_i$, con $l=1, \dots, s$, obtienen porciones de una clave (epsilon/n)-segura, K^A_i y K^B_i , o el símbolo de abortar $\perp_i$. Sea $K^{A_{ij}}$ la j^a porción de K^A_i recibida por la $CLPU^A_i$, y del mismo modo sea $K^{B_{ij}}$ la j^a porción de K^B_i recibida por la $CLPU^B_i$.
2. Generación de $K''^{A_{ij}}$ y $K''^{B_{ij}}$: cada $CLPU^A_i$, con $l=1, \dots, s$, define localmente las secuencias de bits $K''^{A_{ij}}=[0_1, \dots, 0_{i-1}, K^{A_{ij}}, 0_{i+1}, \dots, 0_M]$, donde 0_i , con $i=1, \dots, M$, representa el vector cero de N bits, y M es el número de claves K^A_i y K^B_i que son diferentes del símbolo de abortar $\perp_i$. Del mismo modo, las unidades $CLPU^B_i$ actúan de forma similar y obtienen $K''^{B_{ij}}$.
3. Amplificación de privacidad: las $CLPU^A_i$ usan el protocolo RBS para seleccionar aleatoriamente una función hash 2-universal, hashPA. Calculan $K^A_{ij}=\text{hashPA}(K''^{A_{ij}})$, y por ejemplo las $2t+1$ primeras $CLPU^A_i$ envían la función hashPA a todas las $CLPU^B_{i'}$ a través de los canales clásicos autenticados "Canal Cl. A. l^{AB} " que se muestran en la Figura 6, con $l'=1, \dots, s$. A continuación, las $CLPU^B_{i'}$ usan votación por mayoría para determinar la función hashPA a partir de la información recibida y calculan $K^B_{i'j}=\text{hashPA}(K''^{B_{ij}})$. La función hashPA transforma las secuencias $K''^{A_{ij}}$ y $K''^{B_{ij}}$ de $(M*N)$ bits en dos secuencias de bits más cortas, K^A_{ij} y $K^B_{i'j}$ respectivamente, de un tamaño aproximado de $(M-t'')*N$ bits.

Si $t < M^A_i/3$ y $t' < M^B_i/3$ para todos los $i=1, \dots, M$, donde M^A_i indica el número de $CLPU^A_i$ que no producen el símbolo de abortar, sino que generan porciones post-procesadas, K^A_{ij} , a partir de K^A_i , y M^B_i indica el número de $CLPU^B_i$ que no producen el símbolo de abortar, sino que generan porciones post-procesadas, K^B_{ij} , a partir de K^B_i , entonces

5 podrá reconstruirse una clave final epsilon-segura, K^A y K^B , a partir de las porciones K^A_{ij} y K^B_{ij} siguiendo el mismo procedimiento que reconstruye la clave criptográfica final del Protocolo 2. Nótese que el Protocolo 4 es sólo un ejemplo de una realización de la invención. Siguiendo ideas similares, se podrían definir protocolos alternativos que también permitan la distribución segura de claves criptográficas en este escenario. Por

10 ejemplo, se podría sustituir el primer paso del Protocolo 4 por un paso en el que cada grupo de unidades KGU^A_i , KGU^B_i , $CLPU^A_i$ y $CLPU^B_i$, con $i=1, \dots, n$, realice primero una sesión de generación de claves para producir una clave epsilon-segura, K^A_i y K^B_i , o el símbolo de abortar $\perp_i$, seguido por la distribución de K^A_i entre todas las $CLPU^A_i$ y la distribución de K^B_i entre todas las $CLPU^B_i$ usando el protocolo de distribución de un

15 esquema de compartición de secretos verificable. En este último caso, para poder garantizar que la clave final sea correcta, se podría incluir un paso de verificación de errores implementado en un entorno distribuido actuando sobre porciones de la clave. Asimismo, nótese que el Protocolo 4 y protocolos similares podrían modificarse para garantizar seguridad frente a estructuras adversarias generales, siguiendo las técnicas

20 explicadas para los dos escenarios anteriores. Es importante destacar que el ejemplo ofrecido por el Protocolo 4 ilustra una contribución fundamental de la invención, a saber, la combinación de esquemas de compartición de secretos y técnicas de amplificación de privacidad podrían garantizar la distribución segura de claves criptográficas con KGUs y CLPUs deshonestas.

25

En resumen, puede afirmarse que el objetivo principal de la presente invención es obtener un sistema de generación de claves criptográficas seguro con componentes fabricados y/o adquiridos a varios proveedores no confiables. La invención puede utilizarse como defensa tanto ante unidades de generación de claves (KGUs) no

30 confiables como ante unidades clásicas de post-procesado de datos (CLPUs) no confiables. Además, la presente invención tiene la ventaja adicional de ser robusta ante un ataque de denegación de servicio, ya que la confianza está distribuida entre múltiples KGUs que podrían estar usando canales totalmente distintos.

35 En aras de la simplicidad, hasta ahora se ha discutido la invención para ejemplos

específicos donde solamente hay dos usuarios. Debe tenerse en cuenta que se puede aplicar a un entorno de red en el que hay múltiples usuarios (y por lo tanto múltiples estaciones criptográficas). Además, se puede aplicar a una configuración de red en la que los múltiples usuarios se conectan entre sí a través de varias rutas, y permite a los usuarios combinar las múltiples claves generadas a partir de las múltiples rutas en una clave final que es segura frente a dispositivos no confiables y también es segura frente a rutas comprometidas que contengan nodos controlados por un espía. Además, en aras de la simplicidad, hemos asumido que cada usuario se encuentra físicamente en una sola estación criptográfica local. Nótese sin embargo que lo que representamos como una sola estación criptográfica en las Figuras 3, 4 y 6 podría ser potencialmente un conjunto de nodos físicos que se distribuyen en ubicaciones distantes en una red de comunicaciones.

La invención puede combinarse con relés tanto confiables como no confiables. Además, la invención puede aplicarse a una configuración de QKD que no necesita caracterizar el dispositivo de medición que mide los fotones recibidos (del término anglosajón "measurement-device-independent QKD") en la que un intermediario no confiable, Charles, realiza algunas mediciones (por ejemplo, mediciones para determinar el estado de Bell) en los estados cuánticos enviados por las estaciones criptográficas de Alice y Bob. Además, la invención puede combinarse con repetidores cuánticos.

La invención puede combinarse con varios protocolos clásicos de post-procesado de datos, incluidos protocolos de post-selección o, como ya se mencionó anteriormente, protocolos que añaden ruido.

La invención es compatible con varios protocolos de QKD, incluidos, por ejemplo, QKD con estados señuelo, el protocolo COW, el protocolo QKD RR-DPS y QKD basada en estados cuánticos entrelazados. También se puede aplicar a diversos esquemas de codificación de la información y a sistemas de QKD basados tanto en variable discreta como en variable continua.

Aunque hemos discutido nuestra invención en el marco de seguridad de la composabilidad (en el que un protocolo puede combinarse con otros de forma arbitraria), la presente invención también se puede aplicar a otros marcos de

seguridad, entre ellos uno que implica suposiciones computacionales.

Si bien se ha descrito la invención con referencia a su realización preferente y realizaciones alternativas, se observará que se pueden efectuar diversas modificaciones a las partes y métodos que la componen, sin alejarse de su espíritu ni de su ámbito de protección.

Una persona experta en la técnica reconocería fácilmente que algunos pasos de varios de los métodos descritos anteriormente pueden ser realizados por computadoras programadas. Aquí, algunas realizaciones también están pensadas para abarcar dispositivos de almacenaje de programas, por ejemplo, medios de almacenaje de datos digitales, que son legibles por máquinas o computadoras y codifican programas de instrucciones ejecutables por máquinas o computadoras, en los que dichas instrucciones realizan algunos o todos los pasos de dichos métodos descritos anteriormente. Los dispositivos de almacenaje de programas pueden ser, por ejemplo, memorias digitales, medios de almacenaje magnéticos, como discos magnéticos y cintas magnéticas, discos duros, o medios de almacenaje de datos digitales de lectura óptica. Las realizaciones también están pensadas para abarcar computadoras programadas para llevar a cabo dichos pasos de los métodos descritos anteriormente.

La descripción y los dibujos se limitan a ilustrar los principios de la invención. Aunque la presente invención se ha descrito haciendo referencia a realizaciones específicas, los expertos en la materia entenderán que pueden efectuarse modificaciones, omisiones y adiciones a la forma y los detalles descritos en este documento, sin apartarse del ámbito de la invención, según se define mediante las siguientes reivindicaciones.

Además, todos los ejemplos que aquí se enumeran se ofrecen principal y expresamente con fines pedagógicos, para ayudar al lector a entender los principios de la invención y los conceptos que aporta(n) el/los inventores para extender la técnica, y deben interpretarse de manera no limitativa a dichos ejemplos y condiciones específicamente enumerados. Además, se entienden que todas las menciones del presente documento a principios, aspectos y realizaciones de la invención, así como ejemplos específicos suyos, abarcan asimismo equivalentes de éstos.

Los expertos en la materia entenderán que cualquier diagrama de bloques que figure en este documento representa vistas conceptuales de circuitos ilustrativos que incorporan los principios de la invención. Del mismo modo, se entenderá que cualquier representación gráfica de funcionamiento, diagrama de flujo, diagrama de transición de estado, pseudocódigo y similares representan varios procesos que pueden ser representados sustancialmente en un medio legible por ordenador y ejecutados de esta manera por un ordenador o procesador, independientemente de que dicho ordenador o procesador se muestre o no de forma explícita.

10

REIVINDICACIONES

Reivindicación 1: un método para la generación de claves criptográficas seguras en presencia de unidades no confiables en un sistema criptográfico, el sistema criptográfico incluye una primera y una segunda estaciones criptográficas (A, B), y cada estación incluye n unidades de generación de claves brutas, KGU^A_i , KGU^B_i con $i=1, 2, \dots, n$, y donde $n>1$, siendo al menos una de estas unidades de generación de claves brutas no confiable, y, como mínimo, una unidad clásica de post-procesado de datos $CLPU^A$, $CLPU^B$, incluyendo el método los siguientes pasos:

- Cada par de unidades de generación de claves brutas, KGU^A_i y KGU^B_i , con $i=1, 2, \dots, n$, genera un par de secuencias de datos (clave bruta) y envía (a través de canales de comunicaciones seguros) la secuencia de datos generada por KGU^A_i a, como mínimo, una unidad clásica de post-procesado de datos de la primera estación criptográfica, y envía la secuencia de datos generada por KGU^B_i a, como mínimo, una unidad clásica de post-procesado de datos de la segunda estación criptográfica;

- Las unidades clásicas de post-procesado de datos (al menos una) de la primera y la segunda estaciones criptográficas, $CLPU^A$, $CLPU^B$:

- aplican un procedimiento de post-procesado de datos a cada secuencia de datos recibida y generan una clave criptográfica, K^A_i , K^B_i , con $i=1, 2, \dots, n$, o un símbolo de error (se generará un símbolo de error si la unidad de generación de claves brutas no ha podido generar una secuencia de datos a partir de la cual pueda obtenerse una clave criptográfica) por cada unidad de generación de claves brutas, donde el procedimiento de post-procesado de datos incluye al menos una operación de reconciliación de la información entre las unidades clásicas de post-procesado de datos de ambas estaciones criptográficas a través de un canal de comunicaciones autenticado y un primer procedimiento de amplificación de privacidad para extraer una clave más corta;

- concatenan las claves criptográficas generadas para formar una primera clave criptográfica concatenada $K^{A'}=[K^{A_1}, K^{A_2}, \dots, K^{A_M}]$ y una segunda clave criptográfica concatenada $K^{B'}=[K^{B_1}, K^{B_2}, \dots, K^{B_M}]$ donde M es el número de pares de claves

criptográficas generadas en ambas estaciones criptográficas que difieren del símbolo de error;

- aplican un procedimiento adicional de amplificación de privacidad a la primera clave criptográfica concatenada y a la segunda clave criptográfica concatenada para extraer una primera y una segunda claves criptográficas seguras, respectivamente, K^A y K^B .

Reivindicación 2: un método para la generación de claves criptográficas seguras en presencia de unidades no confiables de un sistema criptográfico, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), y cada estación criptográfica incluye al menos una unidad de generación de claves brutas, KGU^A , KGU^B respectivamente y más de una unidad clásica de post-procesado de datos $CLPU^A_l$, $CLPU^B_{l'}$, con $l = 1, 2, \dots, s$ y $l' = 1, 2, \dots, s'$, siendo al menos una de estas unidades clásicas de post-procesado de datos no confiable, donde el método incluye los siguientes pasos:

- KGU^A genera s secuencias de datos y cada secuencia de datos generada es enviada a una unidad diferente $CLPU^A_l$ y KGU^B genera s' secuencias de datos y cada secuencia de datos generada es enviada a una unidad diferente $CLPU^B_{l'}$;

- Cada unidad clásica de post-procesado de datos de la primera y segunda estaciones criptográficas:

- aplica un procedimiento de post-procesado de datos a cada secuencia de datos recibida y genera o bien una clave criptográfica o un símbolo de error por cada secuencia de datos recibida, donde el procedimiento de post-procesado de datos incluye al menos una operación de reconciliación de la información entre las unidades clásicas de post-procesado de datos de las dos estaciones criptográficas a través de un canal de comunicaciones autenticado y un primer procedimiento de amplificación de privacidad para extraer una clave más corta;

- divide las claves criptográficas generadas en dos o más porciones y las distribuye entre el resto de las unidades clásicas de post-procesado de datos de la primera y segunda estaciones criptográficas, respectivamente;

- genera una porción de una clave criptográfica segura aplicando un procedimiento de verificación de errores y una operación adicional de amplificación de privacidad a las porciones de las claves criptográficas recibidas.

Reivindicación 3: un método para la generación de claves criptográficas seguras en presencia de unidades no confiables de un sistema criptográfico, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), y cada estación criptográfica incluye al menos una unidad de generación de claves brutas, KGU^A , KGU^B respectivamente, y más de una unidad clásica de post-procesado de datos $CLPU_i^A$, $CLPU_{i'}^B$, con $i = 1, 2, \dots, s$ y $i' = 1, 2, \dots, s'$, siendo al menos una de estas unidades clásicas de post-procesado de datos no confiable, donde el método incluye los siguientes pasos:

- Las unidades de generación de claves brutas (una como mínimo) de la primera y la segunda estaciones criptográficas generan una secuencia de datos, R^A , R^B respectivamente, y dividen las secuencias de datos generadas en dos o más porciones y las distribuyen entre las unidades clásicas de post-procesado de datos de la primera y la segunda estaciones criptográficas respectivamente, donde K'^A_{ij} es la porción j^a de R^A recibida por $CLPU_i^A$ y $K'^B_{i'j'}$ es la porción j'^a de R^B recibida por $CLPU_{i'}^B$;

- Cada unidad clásica de post-procesado de datos de la primera y segunda estaciones criptográficas:

- obtiene de cada porción recibida de la secuencia de datos R^A , R^B una porción, $K'^A_{ij, key}$, $K'^B_{i'j', key}$;

- aplica un procedimiento de post-procesado de datos a las porciones $K'^A_{ij, key}$, $K'^B_{i'j', key}$ y genera porciones de una clave criptográfica segura, donde el procedimiento de post-procesado de datos incluye al menos una operación de reconciliación de la información entre las unidades clásicas de post-procesado de datos de ambas estaciones criptográficas a través de un canal de comunicaciones autenticado y un procedimiento de amplificación de privacidad para extraer una clave más corta.

Reivindicación 4: Un método según la reivindicación 3:

- en el que la operación de reconciliación de la información incluye un procedimiento de corrección de errores que implica:

- aplicar determinadas matrices predefinidas M_{EC} a las porciones de datos $K_{ij,key}^A$, $K_{ij',key}^B$ para obtener las secuencias de datos $s_{ij}^A = M_{EC} * K_{ij,key}^A$, $s_{ij'}^B = M_{EC} * K_{ij',key}^B$ respectivamente;
- obtener en cada una de las unidades clásicas de post-procesado de datos una secuencia de datos reconstruida s^A , s^B definida como $s^A = s_{i1}^A \oplus \dots \oplus s_{iq}^A$ y $s^B = s_{i1'}^B \oplus \dots \oplus s_{iq'}^B$ respectivamente, donde s_{ij}^A se obtiene a partir de s_{ij}^A utilizando una estrategia de decisión basada en votación por mayoría y $s_{ij'}^B$ se obtiene a partir de $s_{ij'}^B$ utilizando una estrategia de decisión basada en votación por mayoría;
- modificar el valor de las secuencias de datos $K_{ij,key}^A$, $K_{ij',key}^B$ dependiendo de los valores obtenidos de s^A y s^B ;
- repetir los tres pasos del procedimiento de corrección de errores hasta que la tasa de error se encuentre por debajo de un umbral predefinido;

- en el que la operación de reconciliación de la información incluye un procedimiento de verificación de errores que implica:

- que las unidades clásicas de post-procesado de datos de la primera estación criptográfica seleccionen aleatoriamente una función hash 2-universal, que denominaremos hash, y la apliquen a las porciones $K_{ij,key}^A$ obtenidas a partir de $K_{ij,key}^A$ mediante el procedimiento de corrección de errores, para obtener $h_{ij}^A = \text{hash}(K_{ij,key}^A)$, y cada unidad clásica de post-procesado de datos de la segunda estación criptográfica obtiene $h_{ij'}^B = \text{hash}(K_{ij',key}^B)$ donde las porciones $K_{ij',key}^B$ se obtienen a partir de $K_{ij',key}^B$ mediante el procedimiento de corrección de errores y, a continuación, cada unidad clásica de post-procesado de datos envía las porciones h_{ij}^A y $h_{ij'}^B$ a todas las unidades clásicas de post-procesado de datos de su propia estación criptográfica y a todas las unidades de post-procesado de datos de la otra estación criptográfica;
- obtener en cada unidad clásica de post-procesado de datos dos secuencias de datos reconstruidas h^A , h^B respectivamente y definidas como $h^A = h_{i1}^A \oplus \dots \oplus h_{iq}^A$ y $h^B = h_{i1'}^B \oplus \dots \oplus h_{iq'}^B$ respectivamente, donde h_{ij}^A se obtiene a partir de h_{ij}^A utilizando una estrategia de decisión basada en votación por mayoría y $h_{ij'}^B$ se obtiene a partir de $h_{ij'}^B$ utilizando una estrategia de decisión basada en votación por mayoría;

- cada una de las unidades clásicas de post-procesado de datos comprueba si $h^A=h^B$ y si son iguales pasan al procedimiento de amplificación de privacidad, de lo contrario producen un símbolo de abortar;

5 - en el que el procedimiento de amplificación de privacidad incluye:

10 - que las unidades clásicas de post-procesado de datos de la primera estación criptográfica seleccionen aleatoriamente una función hash 2-universal, que denominaremos hashPA, y la apliquen a las porciones $K_{ij,key}^A$ para obtener porciones de una clave criptográfica segura como $K_{ij}^A=\text{hashPA}(K_{ij,key}^A)$, y cada unidad clásica de post-procesado de datos de la segunda estación criptográfica obtiene porciones de una clave criptográfica segura como $K_{ij'}^B=\text{hashPA}(K_{ij',key}^B)$.

20 Reivindicación 5: Un método según las reivindicaciones 3 ó 4, en el que el método incluye además:

25 - que cada unidad clásica de post-procesado de datos de la primera y la segunda estaciones criptográficas obtiene de cada porción recibida de las secuencias de datos una porción de la subsecuencia para la estimación de parámetros $K_{ij,est}^A$, $K_{ij',est}^B$ y envía dichas porciones de subsecuencias para la estimación de parámetros al resto de unidades clásicas de post-procesado de datos de las dos estaciones criptográficas.

30 Reivindicación 6: un método para la generación de claves criptográficas seguras en presencia de unidades no confiables en un sistema criptográfico, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), y cada estación criptográfica incluye una pluralidad de unidades de generación de claves brutas, KGU_i^A , KGU_i^B con $i=1, 2, \dots, n$, $n>1$ y una pluralidad de unidades clásicas de post-procesado de datos $CLPU_l^A$, $CLPU_{l'}^B$, $l=1, 2, \dots, s$, $l'=1, 2, \dots, s'$, siendo al menos una de estas unidades de generación de claves brutas y al menos una de estas unidades clásicas de post-procesado de datos no confiables, donde el método incluye los siguientes pasos:

40 - Cada unidad de generación de claves brutas de la primera y segunda estaciones criptográficas genera secuencias de datos, R_i^A , R_i^B con $i= 1, 2, \dots, n$, respectivamente, y divide las secuencias

de datos generadas en dos o más porciones y las distribuye entre las unidades clásicas de post-procesado de datos de la primera y segunda estaciones criptográficas respectivamente;

- Cada unidad clásica de post-procesado de datos aplica un procedimiento de post-procesado de datos a cada porción de datos recibida, generando o bien porciones, $K'^{A_{ij}}$, $K'^{B_{l'ij}}$, de una primera clave criptográfica, o un símbolo de error por cada porción de la secuencia de datos recibida, donde el procedimiento de post-procesado de datos incluye al menos una operación de reconciliación de la información entre las unidades clásicas de post-procesado de datos de ambas estaciones criptográficas a través de un canal de comunicaciones autenticado y un primer procedimiento de amplificación de privacidad, donde $K'^{A_{ij}}$ es la porción j^a de la clave criptográfica correspondiente a la secuencia R^A_i obtenida por la unidad $CLPU^A_i$, y $K'^{B_{l'ij}}$ es la porción j^a de la clave criptográfica correspondiente a la secuencia $R^B_{l'}$ obtenida por la unidad $CLPU^{B_{l'}}$;

- Cada unidad clásica de post-procesado de datos $CLPU^A_i$ obtiene porciones de una clave criptográfica segura concatenando primero las porciones $K'^{A_{ij}}$ de la clave criptográfica obtenida en el paso previo, y aplicando un procedimiento adicional de amplificación de privacidad a las secuencias concatenadas obtenidas.

Reivindicación 7: Un método según la reivindicación 6, en el que el último paso de la reivindicación 6 incluye:

- cada $CLPU^A_i$, con $i=1, \dots, s$ obtiene secuencias de datos $K''^{A_{ij}}=[0_1, \dots, 0_{i-1}, K'^{A_{ij}}, 0_{i+1}, \dots, 0_M]$, donde 0_i , con $i=1, \dots, M$, representa un vector cero, y M es el número de pares de unidades de generación de claves brutas que generan secuencias de datos que resultan en una clave y no en el símbolo de error, y cada $CLPU^{B_{l'}}$, con $l'=1, \dots, s'$ obtiene secuencias de bits $K''^{B_{l'ij}}=[0_1, \dots, 0_{i-1}, K'^{B_{l'ij}}, 0_{i+1}, \dots, 0_M]$, con $i=1, \dots, M$;

- Las unidades $CLPU^A_i$, con $i=1, \dots, s$, seleccionan aleatoriamente una función hash 2-universal, hashPA, y luego obtienen porciones de una clave criptográfica segura como $K^A_{ij}=\text{hashPA}(K''^{A_{ij}})$, y cada $CLPU^{B_{l'}}$, con $l'=1, \dots, s'$ obtiene porciones de una clave criptográfica segura como $K^B_{l'ij}=\text{hashPA}(K''^{B_{l'ij}})$.

Reivindicación 8: Un método según las reivindicaciones 1 ó 6 ó 7 en el que el par de secuencias de datos que genera cada par de unidades de generación de claves brutas, KGU^A_i y KGU^B_i , $i=1, 2, \dots, n$, de la primera y segunda estaciones criptográficas respectivamente, se genera utilizando un mecanismo de distribución cuántica de claves.

5

Reivindicación 9: Un método según las reivindicaciones 2 ó 3 ó 4 ó 5 en el que cada par de secuencias de datos generado por cada par de unidades de generación de claves brutas KGU^A y KGU^B de la primera y segunda estaciones criptográficas respectivamente, se genera utilizando un mecanismo de distribución cuántica de claves.

10

Reivindicación 10: Un sistema para la generación de claves criptográficas seguras en presencia de unidades no confiables, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), donde cada estación criptográfica incluye n unidades de generación de claves brutas, KGU^A_i , KGU^B_i con $i=1, 2, \dots, n$, donde $n>1$, siendo al menos una de estas unidades de generación de claves brutas no confiable, y al menos una unidad clásica de post-procesado de datos $CLPU^A$, $CLPU^B$, donde:

15

20

- Cada par de unidades de generación de claves brutas, KGU^A_i y KGU^B_i , incluye medios para la generación de un par de secuencias de datos que están correlacionadas entre sí y envía la secuencia de datos generada por KGU^A_i a como mínimo una unidad clásica de post-procesado de datos de la primera estación criptográfica y envía la secuencia de datos generada por KGU^B_i a como mínimo una unidad clásica de post-procesado de datos de la segunda estación criptográfica;

25

30

- Al menos una de las unidades clásicas de post-procesado de datos de la primera y la segunda estaciones criptográficas, $CLPU^A$, $CLPU^B$ están configuradas para:

35

- aplicar un procedimiento de post-procesado de datos a cada secuencia de datos recibida para generar una clave criptográfica, K^A_i , K^B_i o un símbolo de error por cada unidad de generación de claves brutas, en el que el procedimiento de post-procesado de datos incluye al menos una operación de reconciliación de la información entre las unidades de post-procesado de ambas estaciones criptográficas a través de un canal de comunicaciones autenticado y un primer procedimiento de amplificación de privacidad para extraer una clave más corta;

40

45

- concatenar las claves criptográficas generadas para formar una primera clave criptográfica concatenada $K^{A'}=[K^{A_1}, K^{A_2}, \dots, K^{A_M}]$ y una segunda clave criptográfica concatenada $K^{B'}=[K^{B_1}, K^{B_2}, \dots, K^{B_M}]$ donde M es el número de pares de claves criptográficas generadas en ambas estaciones criptográficas que difieren del símbolo de error;

- aplicar un procedimiento adicional de amplificación de privacidad a la primera clave criptográfica concatenada y a la segunda clave criptográfica concatenada para extraer una primera y una segunda claves criptográficas seguras, respectivamente, K^A y K^B .

Reivindicación 11: Un sistema para la generación de claves criptográficas seguras en presencia de unidades no confiables, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), en el que cada estación criptográfica incluye al menos una unidad de generación de claves brutas, KGU^A , KGU^B respectivamente y más de una unidad clásica de post-procesado de datos $CLPU^A_l$, $CLPU^B_{l'}$, $l=1, 2, \dots, s$, $l'=1, 2, \dots, s'$, siendo al menos una de estas unidades clásicas de post-procesado de datos no confiable, donde:

- KGU^A incluye medios para generar s secuencias de datos y enviar una secuencia de datos generada a cada $CLPU^A_l$ y KGU^B incluye medios para generar s' secuencias de datos que están correlacionados con las secuencias de datos generados por KGU^A y enviar una secuencia de datos generada a cada $CLPU^B_{l'}$;

- Cada unidad clásica de post-procesado de datos de la primera y segunda estaciones criptográficas está configurada para:

- aplicar un procedimiento de post-procesado de datos a cada secuencia de datos recibida para generar una clave criptográfica o un símbolo de error por cada secuencia de datos recibida, donde el procedimiento de post-procesado incluye al menos una operación de reconciliación de la información entre las unidades de post-procesado de las dos estaciones criptográficas a través de un canal de comunicaciones autenticado y un primer procedimiento de amplificación de privacidad para extraer una clave más corta;

- dividir las claves criptográficas generadas en dos o más porciones y distribuirlas entre el resto de las unidades clásicas de post-procesado de datos de la primera y segunda estaciones criptográficas, respectivamente;

5 - generar una porción de una clave criptográfica segura aplicando un procedimiento de verificación de errores y una operación adicional de amplificación de privacidad a las porciones de las claves criptográficas recibidas.

10 Reivindicación 12: Un sistema para la generación de claves criptográficas seguras en presencia de unidades no confiables, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), en el que cada estación criptográfica incluye al menos una unidad de
15 generación de claves brutas, KGU^A , KGU^B y más de una unidad clásica de post-procesado de datos $CLPU_i^A$, $CLPU_{i'}^{B'}$, $i = 1, 2, \dots, s$, $i' = 1, 2, \dots, s'$, siendo al menos una de estas unidades clásicas de post-procesado de datos no confiable, donde:

20 - Al menos una de las unidades de generación de claves brutas de la primera y la segunda estaciones criptográficas generan una secuencia de datos, R^A , R^B respectivamente que están correlacionados entre sí, e incluyen medios para dividir las secuencias de datos generadas en
25 dos o más porciones y distribuirlas entre las unidades clásicas de post-procesado de datos de la primera y la segunda estaciones criptográficas respectivamente, donde $K_{ij}^{A'}$ es la porción j^a de R^A recibida por $CLPU_i^A$ y $K_{i'j'}^{B'}$ es la porción j'^a de R^B recibida por $CLPU_{i'}^{B'}$;

30 - Cada unidad clásica de post-procesado de datos de la primera y segunda estaciones criptográficas está configurada para:

35 - obtener de cada porción recibida de la secuencia de datos una porción de la subsecuencia de generación de claves $K_{ij, key}^{A'}$, $K_{i'j', key}^{B'}$;

40 - aplicar un procedimiento de post-procesado a las porciones de las subsecuencias de generación de claves para generar porciones de claves criptográficas seguras, donde dicho procedimiento de post-procesado incluye una operación de reconciliación de la
45 información entre las unidades de procesamiento de ambas estaciones criptográficas a través

de un canal de comunicaciones autenticado y un procedimiento de amplificación de privacidad.

Reivindicación 13: Un sistema para la generación de claves criptográficas seguras en presencia de unidades no confiables, el sistema incluye una primera y una segunda estaciones criptográficas (A, B), donde cada estación criptográfica incluye una pluralidad de unidades de generación de claves brutas, KGU^A_i , KGU^B_i con $i=1, 2, \dots, n$, $n>1$, y una pluralidad de unidades clásicas de post-procesado de datos $CLPU^A_l$, $CLPU^B_l$, $l=1, 2, \dots, s$, $l'=1, 2, \dots, s'$, siendo al menos una de estas unidades de generación de claves brutas y al menos una de estas unidades clásicas de post-procesado de datos no confiables, donde:

- Cada unidad de generación de claves brutas de la primera y segunda estaciones criptográficas incluye medios para generar secuencias de datos correlacionadas, R^A_i , R^B_i con $i=1, 2, \dots, n$, respectivamente, y dividir las secuencias de datos generadas en dos o más porciones y distribuir las entre las unidades clásicas de post-procesado de datos de la primera y segunda estaciones criptográficas respectivamente,

- Cada unidad clásica de post-procesado de datos está configurada para:

- aplicar un procedimiento de post-procesado de datos a cada porción de la secuencia de datos recibida, generando unas primeras porciones K'^{A}_{ij} , $K'^{B}_{l'ij}$ de una clave criptográfica o un símbolo de error por cada porción de la secuencia de datos recibida, donde el procedimiento de post-procesado de datos incluye al menos una operación de reconciliación de la información entre las unidades de procesamiento de ambas estaciones criptográficas a través de un canal de comunicaciones autenticado y un primer procedimiento de amplificación de privacidad, donde K'^{A}_{ij} es la porción j^a de la clave criptográfica asociada a R^A_i y obtenida por $CLPU^A_l$, y $K'^{B}_{l'ij}$ es la porción j^a de la clave criptográfica asociada a R^B_i obtenida por $CLPU^B_{l'}$;

- obtener porciones de una clave criptográfica segura concatenando las porciones de las primeras claves criptográficas obtenidas y aplicando un procedimiento adicional de amplificación de privacidad a la secuencia concatenada.

Reivindicación 14: Un medio de almacenamiento de datos digital no transitorio para almacenar un programa informático que incluye instrucciones, que hace que un ordenador que ejecute el programa lleve a cabo el método según cualquiera de las reivindicaciones 1-9.

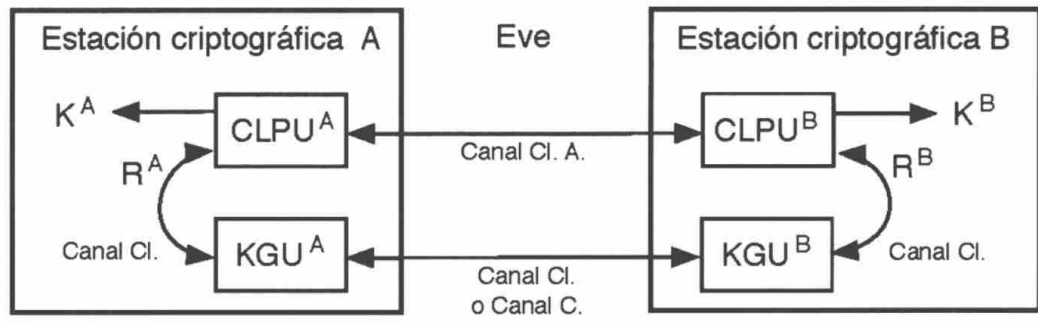


FIGURA 1

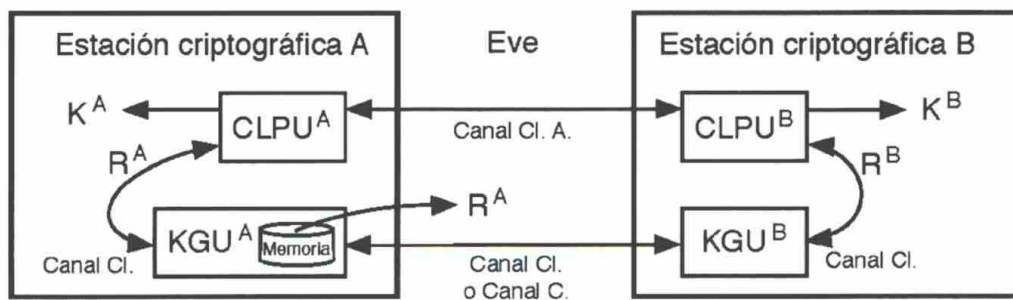


FIGURA 2

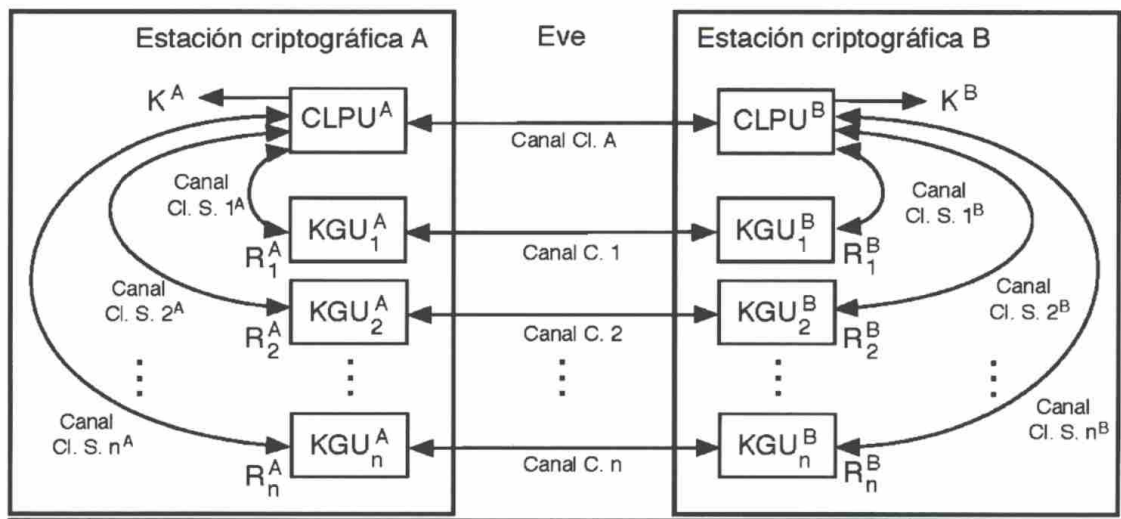


FIGURA 3

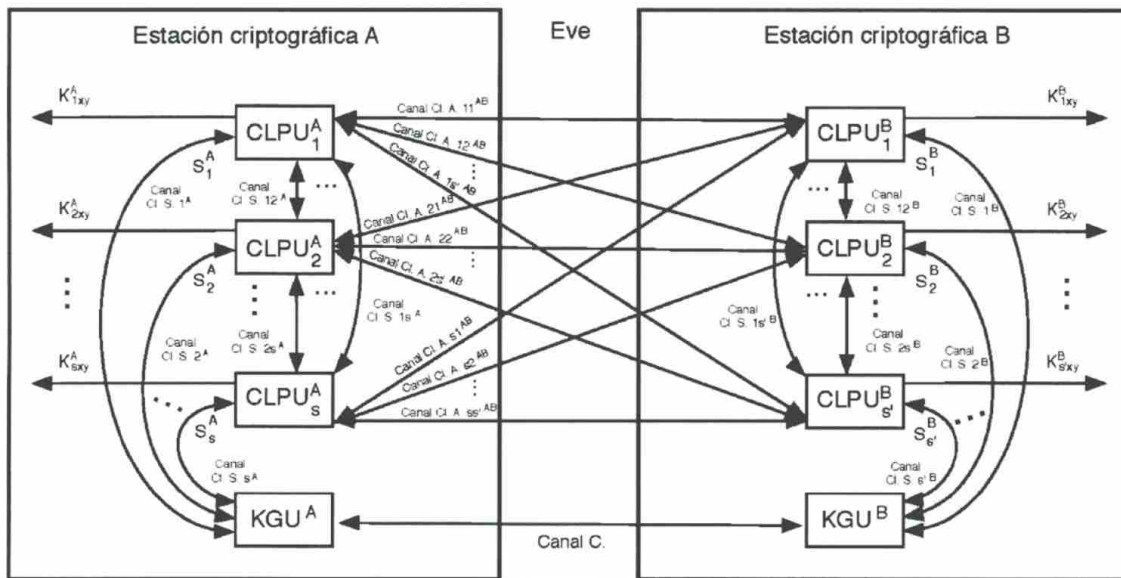
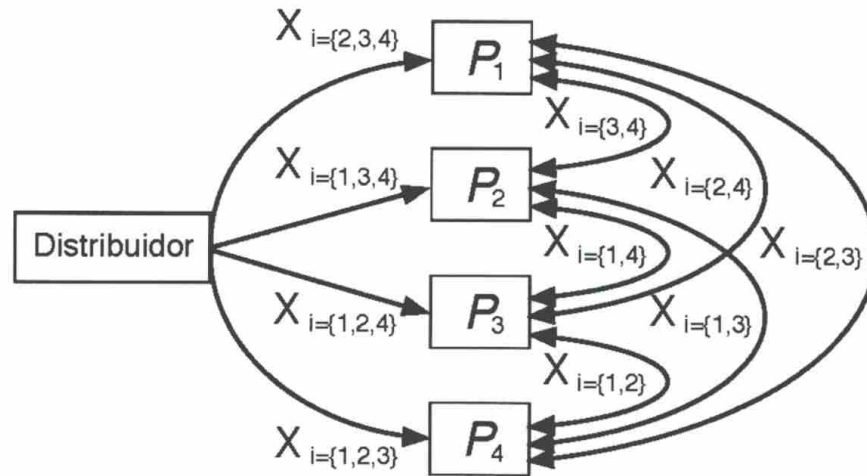


FIGURA 4

Protocolo de Distribución



Protocolo de Reconstrucción

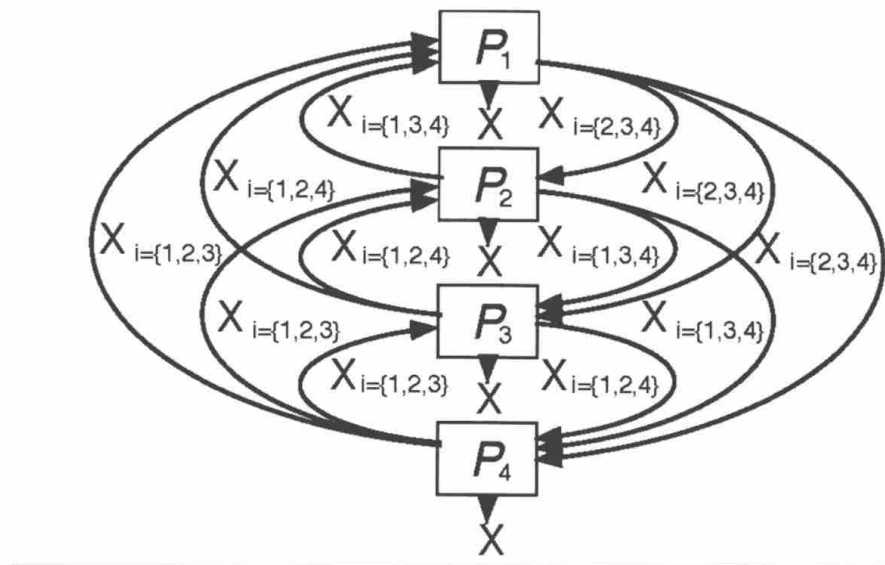


FIGURA 5

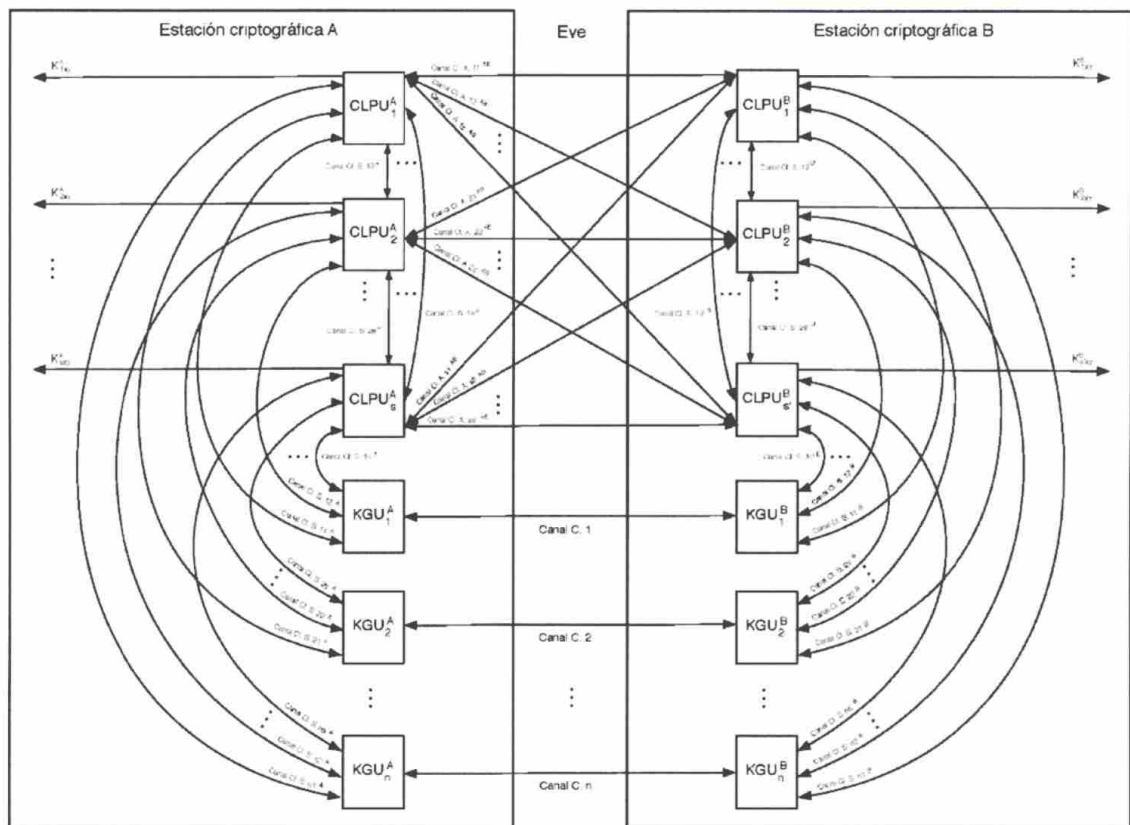


FIGURA 6