D. Nuñez, I. Agudo, and J. Lopez, "Nuevas nociones de seguridad y transformaciones genéricas para criptosistemas de recifrado delegado", *XIV Reunión Española sobre Criptología y Seguridad de la Información*, pp. 174-179, 2016.
NICS Lab. Publications: `https://www.nics.uma.es/publications`

Nuevas nociones de seguridad y transformaciones genéricas para criptosistemas de recifrado delegado

David Nuñez, Isaac Agudo, Javier Lopez
Network, Information and Computer Security Laboratory
Universidad de Málaga, España
Email: {dnunez, isaac, jlm}@lcc.uma.es

Resumen

El recifrado delegado (*proxy re-encryption*) es un tipo de cifrado de clave pública que permite delegar la capacidad de transformar textos cifrados de una clave pública a otra, sin que se pueda obtener ninguna información sobre el mensaje subyacente. Por este motivo, representa un candidato natural para construir mecanismos criptográficos de control de acceso. En este artículo estudiamos algunos de los problemas de seguridad de este tipo de criptosistemas. En primer lugar, examinamos las nociones de seguridad e identificamos una nueva familia paramétrica de modelos de ataque, que considera la disponibilidad tanto del oráculo de descifrado como de recifrado. En segundo lugar, estudiamos la aplicabilidad de transformaciones genéricas para mejorar la seguridad, centrándonos en la transformación Fujisaki-Okamoto, y formulamos las condiciones que nos permiten aplicarla.

1. Introducción

Existe un creciente interés por el modelo de computación en la nube, debido en parte al abaratamiento de costes que supone, sobre todo para las pequeñas empresas, a la hora de escalar y redimensionar su negocio. La idea básica tras el concepto de *cloud* es la de proveer al usuario de una abstracción de un conjunto disponible de recursos de computación, almacenamiento y comunicaciones; en este caso, el modelo de negocio se basa en que los usuarios pagan en función del uso de los servicios ofrecidos por el cloud. Sin embargo, la externalización inherente a este paradigma implica también un riesgo para los usuarios, que ahora están obligadas a confiar en la honestidad y fiabilidad de un proveedor de servicios de cloud. En principio, nada impide a cualquier proveedor acceder a la información de los usuarios, por lo que estos últimos solo pueden confiar en que esto no suceda.

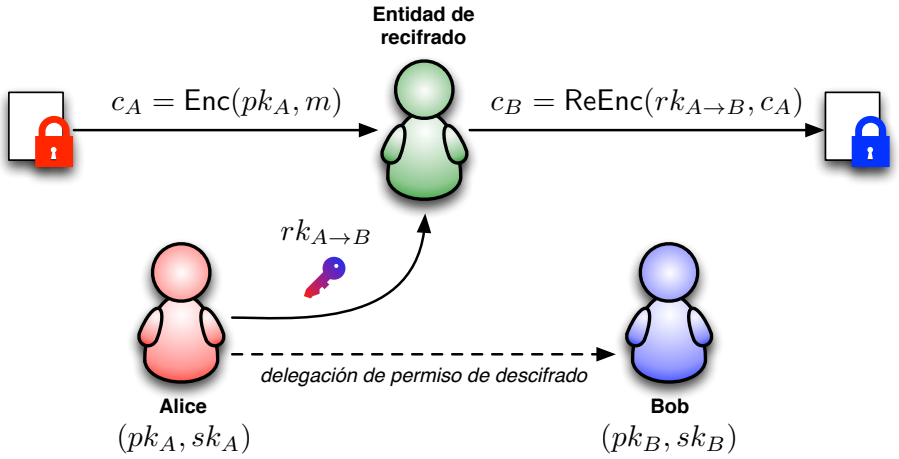


Figura 1: Recifrado Delegado

Si usáramos exclusivamente técnicas convencionales de cifrado (p. ej., de tipo simétrico como AES, o asimétrico como RSA) no podríamos resolver de forma adecuada la problemática de la compartición segura de datos, ya que nos encontraríamos con grandes problemas de gestión y distribución de claves debido a la configuración de actores existente. Este problema se vuelve inmensamente complejo cuando hay diversos productores, numerosos conjuntos de datos, y consumidores que no se conocen de antemano.

Una manera elegante de resolver este problema es mediante el uso de esquemas de *recifrado delegado* (en inglés, *proxy re-encryption*), un tipo de cifrado de clave pública que permite delegar en una entidad de recifrado (referida en inglés como *proxy*) la capacidad de transformar textos cifrados de una clave pública a otra, sin que pueda obtener ninguna información sobre el mensaje subyacente, como puede verse en la Figura 1. Para ello, la entidad de recifrado necesita que el destinatario original le envíe previamente una clave de recifrado, que le permite realizar la transformación. Desde un punto de vista funcional, el recifrado delegado puede verse como un medio de delegación segura de acceso a información cifrada.

En este artículo nos centraremos en aspectos teóricos relativos a la seguridad de los criptosistemas de recifrado delegado. En primer lugar, examinaremos las definiciones de seguridad y el modelado de adversarios. En los esquemas de recifrado delegado es habitual reutilizar nociones de seguridad heredadas de la criptografía de clave pública; aunque esto tiene sentido inicialmente, no se ha estudiado con detenimiento el impacto en las nociones de seguridad que surge de dotar a un adversario de la capacidad de recifrar. En concreto, en este artículo demostramos que esta capacidad por si misma es suficiente para impedir lograr nociones fuertes de seguridad. En segundo lugar, describimos técnicas genéricas para dotar de seguridad a los esquemas de recifrado, inspiradas por técnicas genéricas para esquemas de clave pública; al igual que en el caso ante-

rior, la reutilización de ideas provenientes del mundo de la criptografía de clave pública no puede hacerse de manera directa, por lo que es necesario estudiar bajo qué condiciones es esto posible. En particular, estudiamos la aplicación de la transformación Fujisaki-Okamoto y formulamos las condiciones que nos permiten aplicarla, incluyendo una nueva propiedad que llamamos "*sustitución perfecta de claves*".

2. Familia paramétrica de modelos de ataque

El recifrado delegado se puede considerar como un tipo de criptosistema de clave pública, por lo que es natural reutilizar sus nociones de seguridad. Por tanto, es habitual ver que los esquemas de recifrado intentan lograr nociones de seguridad del tipo IND-CPA [1], IND-CCA1 [2] e IND-CCA2 [3]. Sin embargo, a menudo estas nociones se definen de manera ad-hoc para cada esquema, con sutiles diferencias y restricciones, lo que dificulta la comparación y análisis. Esto está causado por una falta de definiciones comunes de las nociones de seguridad para el recifrado delegado, y en particular, de los modelos de ataque. En comparación con el caso de la criptografía de clave pública, los modelos de ataque para el recifrado delegado son potencialmente más complejos, ya que deben considerar no solo la disponibilidad del oráculo de descifrado, si no también el de recifrado.

En este trabajo exploramos estas sutilezas mediante una familia de modelos de ataque parametrizada por la disponibilidad de ambos oráculos. Esto permite, a su vez, definir una familia de nociones de seguridad, que analizamos en función de las relaciones de implicación y separación que surgen entre dichas nociones. Las separaciones que identificamos son consecuencia del estudio de una nueva propiedad del recifrado delegado que llamamos "*privacidad de las claves de recifrado*" y que formaliza el concepto de que las claves de recifrado no deben filtrarse mediante consultas al oráculo de recifrado. Finalmente, mostramos cómo un esquema de recifrado presentado en PKC 2014 [2] es vulnerable a raíz de no cumplir dicha propiedad.

2.1. Modelos de ataque y nociones de seguridad

Un modelo de ataque define las capacidades de un adversario hipotético, normalmente a través de la disponibilidad de oráculos con una determinada funcionalidad. Bellare et al. definen en [4] una regla mnemotécnica para identificar los modelos de ataque CCA2, CCA1 y CCA0 (es decir, CPA) en la criptografía de clave pública, según la cual el índice en CCAi especifica la última fase en la que el adversario tiene acceso al oráculo de descifrado. De forma análoga, en el contexto del recifrado delegado definimos una familia de modelos de ataque de la forma $\mathsf{CCA}_{i,j}$, con $i, j \in \{0, 1, 2\}$, en donde los índices i y j especifican la última fase en la que el adversario tiene acceso a los oráculos de descifrado y recifrado, respectivamente. Esto da lugar a un conjunto de nueve modelos. Como casos extremos tenemos $\mathsf{CCA}_{0,0}$, equivalente a CPA, y $\mathsf{CCA}_{2,2}$, que representa el

modelo de ataque más completo. La única diferencia entre los nueve modelos de ataque es la disponibilidad de los oráculos de descifrado y recifrado; el resto de oráculos habituales en el recifrado delegado se mantienen.

Al igual que en la criptografía de clave pública, las nociones de seguridad se construyen como una combinación entre un objetivo de seguridad (como p. ej, indistinguibilidad de los cifrados (IND), que formaliza la incapacidad del adversario de distinguir entre qué mensaje está cifrado en un determinado criptograma) y un modelo de ataque, que en este caso pertenece a la familia paramétrica. Si fijamos el objetivo de seguridad IND obtenemos nueve nociones de seguridad, de la forma $\mathsf{IND\text{-}CCA}_{i,j}$, con $i, j \in \{0, 1, 2\}$.

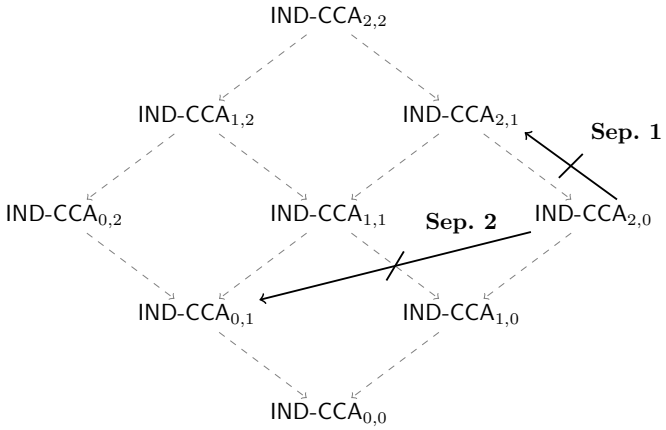


Figura 2: Relaciones entre nociones de seguridad

Las nociones se seguridad obtenidas pueden organizarse jerárquicamente, como muestra la Figura 2, en donde pueden verse las implicaciones triviales entre nociones, representadas por líneas grises punteadas. Estas implicaciones surgen trivialmente de la idea de que un esquema que es seguro en una determinada noción, lo sigue siendo en nociones inferiores en la jerarquía, en donde el adversario dispone de menos capacidades.

Aparte de estas implicaciones triviales, en este trabajo estudiaremos algunas separaciones – lo que es, si cabe, más interesante. En la Figura 2 se muestran resaltadas las relaciones que representan una separación. Estas separaciones son consecuencia de las vulnerabilidades de los esquemas de recifrado delegado que no satisfacen la propiedad de *privacidad de las claves de recifrado*. Esta propiedad, descrita primero en [3] de manera informal, captura la noción de que un adversario no debe de ser capaz de extraer una clave de recifrado a partir de un texto cifrado y su correspondiente recifrado. Nosotros hacemos una definición más completa de dicha propiedad, según la cual un esquema de recifrado delegado la cumple si un adversario que tiene acceso al oráculo de recifrado, tiene una probabilidad insignificante de computar una clave de recifrado válida.

Centrándonos ahora en los esquemas que no cumplen esta propiedad, en función del tipo de clave de recifrado obtenida, las consecuencias para el esquema pueden ser más o menos graves. Si la clave obtenida permite recifrar desde un usuario honesto[1] hacia otro usuario honesto, entonces podemos demostrar que el esquema no puede lograr seguridad $\mathsf{IND\text{-}CCA}_{2,1}$ o $\mathsf{IND\text{-}CCA}_{2,2}$ (Separación 1, en la figura). Sin embargo, si la clave obtenida permite recifrar desde un usuario honesto hacia uno corrupto, entonces el esquema no puede ser ni siquiera de tipo $\mathsf{IND\text{-}CCA}_{0,1}$, lo que produce una separación mucho más grande entre nociones (Separación 2, en la figura).

Para demostrar ambos teoremas se siguen estrategias similares. La primera es que, asumiendo que el adversario puede extraer una clave de recifrado desde el usuario objetivo hacia uno honesto mediante llamadas al oráculo de recifrado, entonces este puede ganar el juego si usa esta clave para recifrar localmente el reto c^* en c_h, para posteriormente utilizar el oráculo de descifrado sobre c_h; esto funciona porque se siguen cumpliendo las restricciones relativas a los derivados del reto, ya que no se ha utilizado el oráculo de recifrado para recifrarlo. La Separación 1, $\mathsf{IND\text{-}CCA}_{2,0} \not\Longrightarrow \mathsf{IND\text{-}CCA}_{2,1}$, surge al observar que no es posible seguir el ataque descrito cuando no se tiene disponibilidad del oráculo de recifrado para obtener la clave de recifrado.

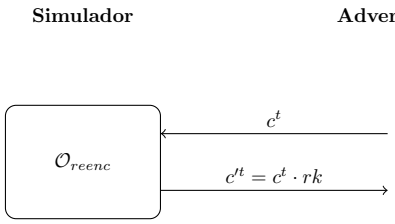
La estrategia utilizada para demostrar la segunda separación es similar, excepto que ahora la clave de recifrado obtenida permite recifrar hacia un usuario corrupto; consecuentemente, no es necesario utilizar el oráculo de descifrado al final. Por lo tanto, la Separación 2 es $\mathsf{IND\text{-}CCA}_{2,0} \not\Longrightarrow \mathsf{IND\text{-}CCA}_{0,1}$, ya que se sigue necesitando el oráculo de recifrado en algún momento para obtener la clave de recifrado.

2.2. Atacando un esquema que no garantiza la privacidad de las claves de recifrado

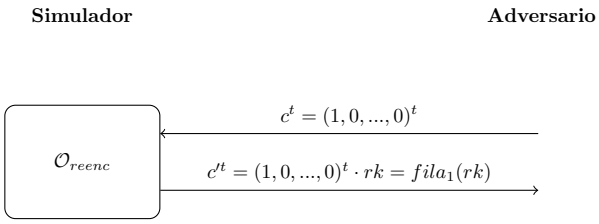
La importancia de la propiedad de privacidad de las claves de recifrado y de estudiar correctamente los modelos de ataque queda de manifiesto con una vulnerabilidad que detectamos en el esquema de Kirshanova [2], presentado en PKC 2014, y que logra supuestamente "seguridad CCA1". Al analizar su prueba de seguridad, observamos que no confiere una capacidad adecuada de recifrado al adversario, correspondiente a un modelo de ataque CCA1. Nuestro estudio revela que en realidad este esquema es de tipo $\mathsf{IND\text{-}CCA}_{1,0}$, ya que aunque permite un oráculo de descifrado en la primera fase, el esquema es vulnerable si se introduce un oráculo de recifrado. El motivo es que no cumple la propiedad de privacidad de las claves de recifrado, por lo que se puede usar un ataque similar a los descritos en las separaciones anteriores.

El esquema de Kirshanova es uno de los primeros esquemas de recifrado basados en retículos. Es una extensión del cifrado de clave pública de Micciancio

[1] En las definiciones de seguridad habituales para el recifrado delegado, es necesario modelar la existencia de diversos usuarios, los cuales pueden ser *honestos* (el adversario solo conoce su clave pública) o *corruptos* (el adversario también dispone de su clave privada). Ver [5] para más detalle al respecto.



(a) El oráculo de recifrado



(b) Vector unitario como entrada

Figura 3: Filtrando la clave de recifrado en el esquema de Kirshanova

y Peikert [6]. Por cuestiones de espacio, no describiremos el esquema; puede encontrarse una descripción simplificada en [5].

La idea básica del ataque es la siguiente: la clave de recifrado es una matriz rk, y la operación de recifrado consiste únicamente en la multiplicación de un vector fila, que contiene el texto cifrado, con dicha matriz, tal como muestra la Figura 3a. Por lo tanto, si usamos como vector de entrada un vector unitario (o más específicamente, los vectores de la base canónica de dimensión correspondiente), obtendremos una de las filas de dicha matriz, como se ve en la Figura 3b. Repitiendo este proceso por cada fila de la matriz rk, podemos recuperarla por completo.

Es posible, además, refinar este ataque. En lugar de usar vectores unitarios como entrada, podemos usar vectores aleatorios, de tal forma que la agrupación de estos vectores dé lugar a una matriz invertible P. Si agrupamos las respuestas en forma de matriz obtendremos, por tanto, $P \cdot rk$; el siguiente paso es multiplicar por la izquierda esta matriz con P^{-1} para recuperar la clave de recifrado rk. Una vez se ha filtrado la clave de recifrado, podemos continuar con la estrategia descrita para la Separación 2, de forma que el adversario gane el juego.

3. Aplicabilidad de transformaciones genéricas para seguridad CCA

Una vez explicadas las sutilezas que aparecen en las nociones de seguridad, en esta sección nos centramos en técnicas genéricas concretas que nos permitan mejorar la seguridad de un esquema de recifrado delegado. Sería deseable que, al igual que ya ocurre en la criptografía de clave pública, lograr esquemas con una noción fuerte de seguridad (por ejemplo, $\mathsf{IND\text{-}CCA}_{2,2}$), pudieran aprovecharse esquemas existentes a los que se les aplicara algún método genérico para tratar de mejorar la noción de seguridad. Para el caso de la criptografía de clave pública, existen transformaciones genéricas, como la Fujisaki-Okamoto [7] y REACT [8]. Sin embargo, este no es el caso del recifrado delegado. En este trabajo, estudiamos la adaptación de estas transformaciones al contexto del recifrado delegado y encontramos resultados tanto positivos como negativos.

3.1. La transformación Fujisaki-Okamoto original

Fujisaki y Okamoto proponen en [7] una transformación genérica para lograr seguridad IND-CCA en el modelo del oráculo aleatorio a partir de un esquema de clave pública que logre seguridad OW-CPA, una débil noción de seguridad. Para ello, la transformación integra también un esquema de cifrado simétrico y funciones hash. La transformación Fujisaki-Okamoto, que denotaremos como Hyb, asume un cifrado de clave pública PKE, un cifrado simétrico Sym, y funciones hash H y G. Asumiremos también que la función de cifrado en PKE, a parte de tomar una clave pública y un mensaje como entrada, también toma un parámetro adicional para añadir aleatoriedad.

Para cifrar un mensaje m, la transformación toma primero un elemento aleatorio σ del espacio de mensajes de PKE. El mensaje m es cifrado con Sym usando $G(\sigma)$ como clave, lo que produce $c = \mathsf{Sym.Enc}(G(\sigma), m)$. A continuación, el término σ se cifra con PKE, tomando $H(\sigma, c)$ como la aleatoriedad de entrada. Por tanto, el texto cifrado correspondiente a m es la tupla:

$$\mathsf{Hyb.Enc}(pk, m) = (\mathsf{PKE.Enc}(pk, \sigma, H(\sigma, c)), c)$$

Para descifrar una de estas tuplas (e, c), la transformación sigue el procedimiento inverso: descifra σ a partir de e, y computa $G(\sigma)$ para obtener la clave de descifrado de c, lo que le permite calcular el mensaje original m. Una sutileza de esta transformación es que después de descifrar el mensaje m, es necesario recalcular $e = \mathsf{PKE.Enc}(pk, \sigma, H(\sigma, c))$ para verificar que es igual que el original. Si no lo es, el texto cifrado se considera como no válido.

Es precisamente este último paso el motivo por el cual la transformación falla si se aplica directamente con un esquema de recifrado delegado: la función de recifrado cambia los textos cifrados de forma que esta validación falla inevitablemente, concretamente al alterar la aleatoriedad original. Por ejemplo, el segundo esquema de Aono et al. [9] falla precisamente por no considerar esta sutileza.

3.2. Extendiendo la transformación Fujisaki-Okamoto

El problema anterior nos lleva a preguntarnos si existen esquemas de recifrado en donde la función de recifrado no altere la aleatoriedad original. En este trabajo caracterizamos una propiedad que captura esta noción, llamada *sustitución perfecta de claves*, y la utilizamos como una de las condiciones para aplicar exitosamente la transformación Fujisaki-Okamoto al recifrado delegado.

Informalmente, un esquema de recifrado con sustitución perfecta de claves se caracteriza porque el resultado de la función de recifrado desde una clave a otra es idéntico al que ocurriría si se cifrara directamente con la segunda clave y usando la misma aleatoriedad. En otras palabras, el recifrado "sustituye" limpiamente una clave por otra, sin que se vea afectada la aleatoriedad original. La siguiente definición captura esta noción de manera formal:

$$\mathsf{ReEnc}(rk_{i\to j}, \mathsf{Enc}(pk_i, m, r)) = \mathsf{Enc}(pk_j, m, r)$$

Los esquemas de recifrado BBS [10] y Canetti-Hohenberger [3] son ejemplos que cumplen esta propiedad. Por ejemplo, en el esquema BBS, los textos cifrados son de la forma $(pk^r, g^r \cdot m)$, para un exponente aleatorio r y claves públicas de la forma $pk = g^a$. Las claves de recifrado son cocientes entre exponentes, de la forma $rk = y/x$. Se puede comprobar que:

$$\begin{aligned}
&\mathsf{ReEnc}(rk_{A\to B}, \mathsf{Enc}(pk_A, m, r)) \\
&\quad = \mathsf{ReEnc}(\frac{b}{a}, ((g^a)^r, g^r \cdot m)) \\
&\quad = ((g^{ar})^{\frac{b}{a}}, g^r \cdot m) \\
&\quad = (g^{br}, g^r \cdot m) = \mathsf{Enc}(pk_B, m, r)
\end{aligned}$$

Puede verse que el texto cifrado original $(g^{ar}, g^r \cdot m)$ se transforma limpiamente en $(g^{br}, g^r \cdot m)$.

Una vez asumimos que el esquema de recifrado delegado cumple esta propiedad, el problema que aparecía en el paso de validación al intentar aplicar la transformación Fujisaki-Okamoto desaparece. La extensión de esta transformación al recifrado delegado es idéntica en cuanto a los algoritmos de generación de claves, cifrado y descifrado; queda, por tanto, definir las funciones de recifrado y generación de claves de recifrado:

- $\mathsf{Hyb.ReKeyGen}(pk_i, sk_i, pk_j, sk_j) \to rk_{i\to j}$. Tomando como entrada dos pares de claves públicas y privadas, correspondientes a los usuarios i y j, este algoritmo devuelve $rk_{i\to j} \leftarrow \mathsf{PRE.ReKeyGen}(pk_i, sk_i, pk_j, sk_j)$.
- $\mathsf{Hyb.ReEnc}(rk_{i\to j}, (e_i, c_i)) \to (e_j, c_j)$. Tomando como entrada una clave de recifrado $rk_{i\to j}$ y un texto cifrado (e_i, c_i), el algoritmo de recifrado devuelve el texto cifrado $(\mathsf{PRE.ReEnc}(rk_{i\to j}, e_i), c_i)$.

Esta extensión cumple con las condiciones de corrección esperadas, además de que su seguridad puede demostrarse [11]. Los problemas mencionados en

la sección anterior con la definición del oráculo de recifrado en el modelo del oráculo aleatorio implican que es necesario asumir que el esquema original tiene seguridad $\mathsf{IND\text{-}CCA}_{0,1}$ (noción inmediatamente superior a IND-CPA) y que el esquema que se obtiene tras la transformación es de tipo $\mathsf{IND\text{-}CCA}_{2,1}$ (noción inmediatamente inferior a IND-CCA2).

Esta misma aproximación podría aplicase para extender otras transformaciones genéricas similares, como REACT [8] y GEM [12].

3.3. Imposibilidad de aplicación directa

Como resultado negativo de esta investigación, mostramos que la aplicación directa de estas transformaciones implica fallos sutiles en algunas demostraciones de seguridad. En la versión completa de este trabajo [11], detectamos fallos en una docena de esquemas, entre los que destacamos [13, 14, 15, 16].

Para comprender el fallo es necesario conocer cómo se hace la prueba de seguridad en la transformación Fujisaki-Okamoto original. Como el objetivo de esta transformación es lograr seguridad CCA, es necesario definir un oráculo de descifrado. En la prueba, este oráculo hace uso de las tablas asociadas a los oráculos aleatorios, que contienen un registro con las entradas y salidas de estas funciones. Asumiendo acceso a dichas tablas, es posible construir un oráculo de descifrado que no requiera la correspondiente clave de descifrado.

Esta estrategia, que funciona muy bien para el oráculo de descifrado, no puede usarse para el oráculo de recifrado. El problema consiste en que la demostración difiere de la ejecución real del esquema para cierto tipo de consultas de recifrado, lo que invalida las demostraciones. Este es precisamente el fallo común en los once esquemas anteriores.

El problema se basa en que en la función de cifrado los valores aleatorios se obtienen al llamar a una función hash con un valor que se mantiene secreto (por ejemplo, el propio mensaje original). Durante el descifrado, estos valores se pueden obtener de nuevo, por lo que es posible recuperar los valores aleatorios para realizar el paso de validación. Sin embargo, esto no es así en el recifrado, ya que los valores secretos no se desvelan en este proceso, por lo que no es posible verificar aquí si el texto cifrado se generó correctamente (es decir, usando las funciones hash de manera adecuada) o no. En las demostraciones asociadas a los esquemas fallidos, si un adversario hace una consulta a un oráculo de recifrado usando un texto cifrado no válido (por ejemplo, sin usar las funciones hash), entonces su oráculo de recifrado rechaza el texto cifrado al considerarlo inválido, ya que no hay un registro correspondiente en las tablas de los oráculos aleatorios. No obstante, en una ejecución real, no es posible comprobar esto, por lo que el recifrado va a funcionar normalmente. Consecuentemente, estas demostraciones de seguridad no son correctas al no corresponderse con el comportamiento real de los esquemas.

4. Conclusiones y trabajo futuro

Este artículo se ha centrado, principalmente, en el estudio del recifrado delegado. La motivación inicial surge de la problemática de la compartición segura de datos, una funcionalidad especialmente relevante hoy en día, ya que es muy habitual tener información almacenada en la nube. El recifrado delegado constituye una elegante forma de mantener los datos cifrados en todo momento, permitiendo al proveedor de almacenamiento compartir la información sin necesidad de acceder a los datos.

En este artículo describimos diversos retos de investigación asociados a los esquemas de recifrado delegado, en particular aquellos que surgen del análisis de su seguridad. A nivel más fundamental, identificamos una familia paramétrica de modelos de ataque para el recifrado delegado, que considera separadamente la capacidad de descifrar de la de recifrar. Esto nos permite definir, a su vez, una familia de nociones de seguridad, cuyas relaciones analizamos. De especial interés es el estudio de las separaciones entre nociones, que explotamos mediante el análisis de una propiedad deseable en el recifrado delegado: la privacidad de las claves de recifrado. Descubrimos que los esquemas que no cumplen esta propiedad no pueden lograr una noción de seguridad significativa, y lo ilustramos al mostrar un ataque a un esquema reciente.

Aparte de la investigación sobre modelos de ataque, que se sitúa en el plano de los fundamentos del concepto de seguridad, realizamos un estudio sobre transformaciones genéricas para incrementar la seguridad, más cercano a las construcciones concretas. A este respecto definimos una extensión de la transformación Fujisaki-Okamoto e identificamos las condiciones que permiten aplicarla. La aplicación de esta y otras transformaciones genéricas no es obvia, ya que la funcionalidad de recifrado interfiere con los mecanismos de comprobación de validez que son inherentes a estas transformaciones. Como prueba de ello, mostramos hasta 12 esquemas de recifrado delegado que son incorrectos a causa de esta problemática.

Como trabajo futuro, hay diversas líneas de actuación con problemáticas interesantes. Por ejemplo, no descartamos que el estudio de relaciones entre los modelos de ataque y nociones de seguridad puede ampliarse con nuevas implicaciones y separaciones. Respecto a la contribución sobre transformaciones genéricas, una línea futura de trabajo consiste en estimar cuantitativamente el nivel de seguridad obtenido, de forma que se puedan inferir conjuntos de parámetros concretos. Aparte de esto, un problema ambicioso es desarrollar transformaciones genéricas en el modelo estándar.

Referencias

[1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security*, 9(1):1–30, 2006.

[2] Elena Kirshanova. Proxy re-encryption from lattices. In *Public-Key Cryptography–PKC 2014*, pages 77–94. Springer, 2014.

[3] R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 185–194. ACM, 2007.

[4] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology—CRYPTO'98*, pages 26–45. Springer, 1998.

[5] David Nuñez, Isaac Agudo, and Javier Lopez. A parametric family of attack models for proxy re-encryption. In *Proceedings of the 28th IEEE Computer Security Foundations Symposium*, CSF'15, pages 290–301. IEEE Computer Society, 2015.

[6] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology–EUROCRYPT 2012*, pages 700–718. Springer, 2012.

[7] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of cryptology*, 26(1):80–101, 2013.

[8] Tatsuaki Okamoto and David Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In *Topics in Cryptology—CT-RSA 2001*, pages 159–174. Springer, 2001.

[9] Yoshinori Aono, Xavier Boyen, Le Trieu Phong, and Lihua Wang. Key-private proxy re-encryption under LWE. In *Progress in Cryptology–INDOCRYPT 2013*, pages 1–18. Springer, 2013.

[10] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. *Advances in Cryptology—EUROCRYPT'98*, pages 127–144, 1998.

[11] David Nuñez, Isaac Agudo, and Javier Lopez. On the application of generic CCA-secure transformations to proxy re-encryption. *Security and Communication Networks*, 2016.

[12] Jean-Sébastien Coron, Helena Handschuh, Marc Joye, Pascal Paillier, David Pointcheval, and Christophe Tymen. GEM: A generic chosen-ciphertext secure encryption method. In *Topics in Cryptology—CT-RSA 2002*, pages 263–276. Springer, 2002.

[13] Jian Weng, Robert H Deng, Xuhua Ding, Cheng-Kang Chu, and Junzuo Lai. Conditional proxy re-encryption secure against chosen-ciphertext attack. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 322–332. ACM, 2009.

[14] J. Shao and Z. Cao. CCA-secure proxy re-encryption without pairings. In *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC'09*, pages 357–376. Springer-Verlag, 2009.

[15] Jian Weng, Robert H Deng, Shengli Liu, and Kefei Chen. Chosen-ciphertext secure bidirectional proxy re-encryption schemes without pairings. *Information Sciences*, 180(24):5077–5089, 2010.

[16] Sherman SM Chow, Jian Weng, Yanjiang Yang, and Robert H Deng. Efficient unidirectional proxy re-encryption. In *Progress in Cryptology–AFRICACRYPT 2010*, pages 316–332. Springer, 2010.