Cifrado Basado en Identidades Simétrico

Javier Herranz
Dept. Matemàtiques, Universitat Politècnica de Catalunya
C. Jordi Girona 1-3, Barcelona, 08034
javier.herranz@upc.edu

***Resumen*—El cifrado basado en identidades (IBE) es un paradigma muy popular y útil, con aplicaciones tanto teóricas como prácticas. Desafortunadamente, diseñar esquemas de cifrado basado en identidades seguros es una tarea difícil: existen algunos resultados de imposibilidad, y actualmente los únicos esquemas IBE eficientes seguros utilizan *bilinear pairings* o *lattices*.**

La situación cambia si alguno de los requisitos del esquema IBE se suaviza. Un primer ejemplo es el caso de esquemas IBE con colusión acotada, en los que se conoce de antemano una cota del número de peticiones de clave secreta que el adversario puede hacer. En este trabajo consideramos otra manera diferente de relajar la noción estándar de IBE: consideramos cifrado basado en identidades simétrico (sIBE), en el que la misma clave (máster) que se usa para generar claves secretas de usuario debe ser utilizada para cifrar los mensajes. Después de motivar la noción de sIBE con algunas aplicaciones en sistemas reales, mostramos que se pueden construir esquemas sIBE seguros de una manera bastante sencilla, en el modelo del oráculo aleatorio, a partir de un sistema seguro de cifrado simétrico. Por tanto, los resultados de imposibilidad conocidos para IBE estándar no aplican a sIBE.

Palabras clave: cifrado basado en identidades, criptografía simétrica, modelo del oráculo aleatorio.

I. Introducción

La criptografía basada en identidades fue propuesta por Shamir [1] como alternativa al paradigma clásico de criptografía de clave pública. La idea era evitar la necesidad de certificados digitales que vinculasen una clave pública con la identidad del usuario correspondiente. En un sistema de cifrado basado en identidades (IBE, de sus iniciales en inglés), hay una entidad máster que usa sus claves de máster para generar claves secretas de usuario, $\mathsf{sk}_{\mathsf{id}}$, para cualquier usuario que lo solicite, presentando su identidad id. Para cifrar un mensaje a un usuario con identidad id, sólo hace falta conocer id (que actúa como clave pública, en cierta manera). El cifrado resultante sólo podrá ser descifrado por aquél que conozca $\mathsf{sk}_{\mathsf{id}}$, cosa que en teoría incluye al usuario con identidad id (pero también a la entidad máster). Además de esa aplicación a evitar los certificados digitales, los sistemas IBE han encontrado otras aplicaciones, tanto a nivel teórico como práctico.

Tuvieron que pasar 17 años hasta que se propuso el primer sistema IBE con seguridad demostrable [2], en el que se usaban emparejamientos bilineales (*bilinear pairings*), una herramienta algebraica definida en ciertas curvas elípticas. Todos los intentos de diseñar sistemas IBE seguros y razonablemente eficientes (en particular, en los que los parámetros del sistema no dependan del número de posibles identidades) basados en herramientas criptográficas de clave pública clásica (como RSA o bien criptografía basada en la dificultad del logaritmo discreto en grupos cíclicos) han sido infructuosos. De hecho, existen resultados de imposibilidad que cierran la puerta a diseñar sistemas IBE usando (de manera *black-box*) una permutación unidireccional con trampa (como RSA) o a partir de la dificultad del problema Decisional de Diffie-Hellman (DDH) [3], [4]. Para sortear estos resultados de imposibilidad, por ejemplo en el caso de grupos cíclicos $\mathbb{G}$ clásicos donde el problema del logaritmo discreto es difícil (sin emparejamientos bilineales), hay dos posibilidades:

(a) Intentar diseñar un sistema IBE estándar, pero sin usar las operaciones en el grupo $\mathbb{G}$ de manera *black-box*, sino usando el circuito que calcula esas operaciones (en particular, las exponenciaciones). Esta estrategia se ha seguido en [5], pero el sistema IBE resultante no es nada eficiente.

(b) Relajar alguno de los requisitos del sistema IBE que se quiere diseñar, pensando en que ese sistema IBE no tendrá todas las propiedades posibles, pero tal vez sí las suficientes para algunas aplicaciones reales interesantes.

En este trabajo nos centramos en la opción (b). No somos los primeros en hacerlo. Una posibilidad ya analizada es la de considerar sistemas IBE en los que se conoce con anterioridad el número máximo q_k de peticiones de clave secreta de usuario que puede realizar un adversario que intente atacar el sistema IBE. Esto puede ser una opción realista en situaciones en las que el sistema IBE se va a implementar en empresas o instituciones moderadamente pequeñas. Si permitimos que el tamaño de los parámetros del sistema IBE (por ejemplo los parámetros públicos pms o las claves de usuario) pueda depender linealmente de esa cota q_k, entonces se pueden diseñar sistemas IBE *con colusión acotada* basados en cualquier sistema de cifrado de clave pública con seguridad semántica (como ElGamal) [6], en particular basándonos en la dificultad del problema DDH en grupos cíclicos sin emparejamientos bilineales. Por tanto, los resultados de imposibilidad [3], [4] no aplican a los sistemas IBE con colusión acotada.

Aquí consideramos otra posible manera de alterar (relajar) las propiedades de un sistema IBE: imponemos que sea necesario conocer la clave secreta de máster para poder cifrar un mensaje. A los sistemas resultantes los llamaremos sistemas de cifrado basado en identidades simétricos (usaremos las siglas sIBE). En otras palabras, en un sistema sIBE la misma entidad (máster) que cifra / esconde una cierta información es la que puede luego generar claves de usuario para que otros usuarios puedan descifrar y obtener la información en claro.

I-A. Posibles Aplicaciones de sIBE

En general, la noción de cifrado basado en identidades simétrico tiene aplicabilidad en cualquier escenario en que la misma persona / entidad que posee una cierta información sensible (que guarda cifrada por ejemplo en la Nube) quiere

después dar acceso (gratis o previo pago) a diferentes partes de esa información, a diferentes usuarios. Aquí van algunos ejemplos concretos:

- Delegación a diferentes dispositivos. Un usuario puede cifrar toda su información sensible en la nube, usando diferentes identidades, como id_1 =móvil, id_2 =portátil, id_3 =tableta, id_4 =PC_trabajo, id_5 =PC_casa, para diferentes partes de la información, en función de los dispositivos que cree que van a tener que acceder a esos datos en el futuro.
- Venta de contenidos digitales. Una plataforma de distribución de contenido digital (series, películas...) puede tener todo ese contenido cifrado en la Nube, bajo diferentes identidades. Por ejemplo una película puede estar cifrada dos veces, con las identidades "películas" y "total", de manera que cuando un usuario se suscribe a la plataforma, en función de su tipo de suscripción se le asignan diferentes claves secretas de usuario. Si paga una cuota para ver sólo películas y series, recibirá dos claves secretas, para "películas" y "series". Si un usuario paga la cuota máxima, recibirá la clave secreta para la identidad "total".
- Revelación gradual de secretos. Supongamos que una persona o entidad, que llamaremos Sn, consigue de alguna manera una gran cantidad de información sensible y compremetedora para uno o varios países. Sn decide compartir esa información con todo el mundo, pero de una manera gradual: divide la información en paquetes, y decide que revelará un paquete cada mes. Como Sn sospecha que los servicios secretos de los países implicados están intentando hacerse con esa información, lo que hace es cifrar toda la información y publicarla en Internet; cada bloque lo cifra con una identidad temporal, del tipo "octubre_2020". Lo único que tiene que guardar secretamente Sn (tal vez con varias copias de seguridad repartidas entre dispositivos y personas de su confianza) es la clave secreta de máster, msk. Cuando llegue cada mes, por ejemplo octubre de 2020, Sn usa msk para calcular la clave secreta $\mathsf{sk}_{\mathsf{id}}$ correspondiente a id ="octubre_2020" y envía $\mathsf{sk}_{\mathsf{id}}$ a todos los medios de comunicación del mundo.

I-B. Nuestra Contribución

La principal contribución del trabajo, además de introducir y definir la noción de sIBE y su seguridad, es la construcción genérica de un sistema sIBE seguro usando como ingredientes un sistema de cifrado simétrico y una función de hash (que en la demostración de seguridad, en el *random oracle model*, se modela como una función perfectamente aleatoria). Eso quiere decir que se pueden diseñar sistemas sIBE concretos y seguros de manera muy eficiente, utilizando básicamente funciones de hash y otras técnicas de criptografía simétrica, mucho más eficaces que las de criptografía asimétrica. A nivel de seguridad teórica, nuestra construcción también implica que los resultados de imposibilidad conocidos para sistemas IBE estándar no aplican a sistemas IBE simétricos.

II. Definiciones

II-A. Cifrado Simétrico

Un sistema de cifrado simétrico SKE consiste en tres algoritmos probabilísticos que requieren tiempo polinómico:

- $\mathsf{SKE.KG}(1^\lambda)$. El algoritmo setup toma como entrada el parámetro de seguridad λ, y da como salida unos parámetros públicos pms y una clave secreta k. Por un lado, pms contienen la descripción de los espacios de mensajes $\mathcal{M}$, de claves $\mathcal{K}$, de textos cifrados $\mathcal{C}$, y tal vez otros parámetros. Los parámetros pms son entrada (que no explicitaremos) del resto de algoritmos de SKE. Por otro lado, la clave secreta k se escoge de manera aleatoria y uniforme en el espacio $\mathcal{K}$.
- $\mathsf{SKE.Enc}(m, k)$. El algoritmo de cifrado toma como entradas un mensaje m y la clave secreta k, y da como salida un texto cifrado C.
- $\mathsf{SKE.Dec}(C, k)$. El algoritmo de descifrado toma como entradas un texto cifrado C y la clave secreta k, y da como salida un mensaje $\tilde{m}$.

Para que un esquema SKE funcione correctamente, requerimos que, para toda ejecución $(\mathsf{pms}, k) \leftarrow \mathsf{SKE}.KG(1^\lambda)$ que dé lugar a un espacio de mensajes $\mathcal{M}$, para todo mensaje $m \in \mathcal{M}$ y toda ejecución $C \leftarrow \mathsf{SKE.Enc}(m, k)$, se cumpla $m \leftarrow \mathsf{SKE.Dec}(C, k)$.

La seguridad requerida a un esquema de cifrado simétrico es la siguiente: un adversario no debe obtener ninguna información sobre el mensaje escondido en un texto cifrado C^*, incluso si el adversario sabe que el mensaje sólo puede ser uno de dos mensajes posibles (escogidos por él), e incluso si el adversario puede pedir que se cifren mensajes de su elección, durante el ataque. La seguridad resultante se conoce como indistinguibilidad de mensajes frente a ataques de mensaje en claro escogido (seguridad IND-CPA, de las siglas en inglés).

Formalmente, esta noción de seguridad se define mediante el siguiente experimento $\mathbf{Exp}^{\mathsf{SKE}}_{\mathcal{B}}(\lambda)$, en el que participa un adversario $\mathcal{B}$, que actúa en dos fases (el parámetro st_1 se usa para la información adicional que $\mathcal{B}$ podría pasarse a si mismo de la primera a la segunda fase).

1. $b \xleftarrow{R} \{0, 1\}$ se escoge aleatoriamente,
2. Se ejecuta $(\mathsf{pms}, k^*) \leftarrow \mathsf{SKE.KG}(1^\lambda)$,
3. $(m_0, m_1, st_1) \leftarrow \mathcal{B}^{\mathsf{SKE.Enc}(\cdot, k^*)}(\mathsf{pms})$,
4. $C^* \leftarrow \mathsf{SKE.Enc}(m_b, k^*)$,
5. $b' \leftarrow \mathcal{B}^{\mathsf{SKE.Enc}(\cdot, k^*)}(C^*, \mathsf{pms}, st_1)$,
6. Devuelve 1 si $b' = b$ y $m_0 \neq m_1$.

La ventaja de $\mathcal{B}$ en romper la seguridad IND-CPA del esquema SKE se define como

$$\mathsf{Adv}^{\mathsf{SKE}}_{\mathcal{B}}(\lambda) = \left| \Pr[\mathbf{Exp}^{\mathsf{SKE}}_{\mathcal{B}}(\lambda) = 1] - \frac{1}{2} \right|.$$

Definición 1. *Un esquema de cifrado simétrico SKE es IND-CPA seguro si, para todo adversario $\mathcal{B}$ con tiempo de ejecución polinómico en λ, se cumple que la ventaja $\mathsf{Adv}^{\mathsf{SKE}}_{\mathcal{B}}(\lambda)$ es una función negligible en λ.*

II-B. Cifrado Basado en Identidades Simétrico

Un sistema de cifrado basado en identidades simétrico sIBE consiste en cuatro algoritmos probabilísticos que requieren tiempo polinómico:

- $\mathsf{sIBE.Setup}(1^\lambda)$. El algoritmo setup toma como entrada el parámetro de seguridad λ, y da como salida unos parámetros públicos pms y una clave secreta de máster, msk. Como antes, pms será una entrada del resto de los algoritmos, que no explicitamos.
- $\mathsf{sIBE.KG}(\mathsf{msk}, \mathsf{id})$. El algoritmo de generación de claves toma como entradas la clave secreta de máster y una identidad id, y da como salida una clave secreta de usuario, $\mathsf{sk}_{\mathsf{id}}$.
- $\mathsf{sIBE.Enc}(m, \mathsf{msk}, \mathsf{id})$. El algoritmo de cifrado toma como entradas un mensaje m, la clave secreta de máster y una identidad id, y da como salida un texto cifrado C.
- $\mathsf{sIBE.Dec}(C, \mathsf{sk}_{\mathsf{id}})$. El algoritmo de descifrado toma como entradas un texto cifrado C y una clave de usuario $\mathsf{sk}_{\mathsf{id}}$, y da como salida un mensaje $\tilde{m}$.

Para que un esquema sIBE funcione correctamente, requerimos que, para toda identidad $\mathsf{id} \in \mathcal{ID}$, todo mensaje m y toda ejecución de los algoritmos $(\mathsf{pms}, \mathsf{msk}) \leftarrow \mathsf{sIBE.Setup}(1^\lambda)$, $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{sIBE.KG}(\mathsf{msk}, \mathsf{id})$, $C \leftarrow \mathsf{sIBE.Enc}(m, \mathsf{msk}, \mathsf{id})$, se cumpla $m \leftarrow \mathsf{sIBE.Dec}(C, \mathsf{sk}_{\mathsf{id}})$.

La seguridad requerida a un esquema sIBE será la siguiente (es la combinación natural entre la seguridad requerida a un esquema de cifrado simétrico y la requerida a un esquema de cifrado basado en identidades): un adversario no debe obtener ninguna información sobre el mensaje escondido en un texto cifrado C^* dirigido a un usuario con identidad id^*, incluso si el adversario sabe que el mensaje sólo puede ser uno de dos mensajes posibles (escogidos por él), e incluso si el adversario puede pedir que se cifren mensajes de su elección y puede pedir un número polinómico de claves secretas de usuario, obviamente para identidades diferentes a id^*. La seguridad resultante la denotaremos seguridad IND-CPA.

Formalmente, esta noción de seguridad se define mediante el siguiente experimento $\mathbf{Exp}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$, en el que participa un adversario $\mathcal{A}$.

1. $b \xleftarrow{R} \{0,1\}$ se escoge aleatoriamente,
2. $(\mathsf{pms}, \mathsf{msk}) \leftarrow \mathsf{sIBE.Setup}(1^\lambda)$,
3. $(m_0, m_1, \mathsf{id}^*, st_1) \leftarrow \mathcal{A}^{\mathsf{sIBE.Enc}(\cdot,\mathsf{msk},\cdot),\mathsf{sIBE.KG}(\mathsf{msk},\cdot)}(\mathsf{pms})$
4. $C^* \leftarrow \mathsf{sIBE.Enc}(m_b, \mathsf{msk}, \mathsf{id}^*)$,
5. $b' \leftarrow \mathcal{A}^{\mathsf{sIBE.Enc}(\cdot,\mathsf{msk},\cdot),\mathsf{sIBE.KG}(\mathsf{msk},\cdot)}(C^*, \mathsf{pms}, st_1)$,
6. Devuelve 1 si $b' = b$, $m_0 \neq m_1$ y $\mathcal{A}$ no ha hecho ninguna petición de clave de usuario para id^*.

La ventaja de $\mathcal{A}$ en romper la seguridad IND-CPA del esquema sIBE se define como

$$\mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda) = \left| \Pr[\mathbf{Exp}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|.$$

Definición 2. *Un esquema de cifrado basado en identidades simétrico sIBE es IND-CPA seguro si, para todo adversario $\mathcal{A}$ con tiempo de ejecución polinómico en λ, se cumple que la ventaja $\mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$ es una función negligible en λ.*

III. Construyendo sIBE a partir de Cifrado Simétrico

Sea $\mathsf{SKE} = (\mathsf{SKE.KG}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ un esquema de cifrado simétrico. Construimos a continuación un esquema de cifrado basado en identidades, sIBE, que utiliza los algoritmos de SKE; denotamos como $\mathcal{ID}$ el espacio de identidades admitidas por el esquema sIBE, cuyos algoritmos funcionan de la manera siguiente.

- $\mathsf{sIBE.Setup}(1^\lambda)$. Se ejecuta en primer lugar el protocolo $(\mathsf{pms}_{\mathsf{SKE}}, k) \leftarrow \mathsf{SKE}.KG(1^\lambda)$. Recordemos que $\mathsf{pms}_{\mathsf{SKE}}$ incluye los espacios de mensajes $\mathcal{M}$, claves $\mathcal{K}$ y textos cifrados C de esa ejecución de SKE. Tomamos una función de hash $H : \{0,1\}^\lambda \times \mathcal{ID} \to \mathcal{K}$. Los parámetros públicos de sIBE son $\mathsf{pms} = (\mathsf{pms}_{\mathsf{SKE}}, H)$. La clave secreta de máster $\mathsf{msk} \in \{0,1\}^\lambda$ se genera de manera aleatoria uniforme.
- $\mathsf{sIBE.KG}(\mathsf{msk}, \mathsf{id})$. La clave secreta de usuario se define como $\mathsf{sk}_{\mathsf{id}} = H(\mathsf{msk}, \mathsf{id}) \in \mathcal{K}$.
- $\mathsf{sIBE.Enc}(m, \mathsf{msk}, \mathsf{id})$. Para cifrar el mensaje m para la identidad id, primero se usa la clave secreta de máster msk para calcular la clave de cifrado simétrico $\mathsf{sk}_{\mathsf{id}} = H(\mathsf{msk}, \mathsf{id})$ y después se cifra el mensaje como $C \leftarrow \mathsf{SKE.Enc}(m, \mathsf{sk}_{\mathsf{id}}) \in \mathcal{C}$.
- $\mathsf{sIBE.Dec}(C, \mathsf{sk}_{\mathsf{id}})$. La salida de este algoritmo es la salida que se obtenga al ejecutar $\tilde{m} \leftarrow \mathsf{SKE.Dec}(C, \mathsf{sk}_{\mathsf{id}})$.

Es evidente que la construcción funciona de manera correcta, si suponemos que así lo hace el esquema SKE. En lo referente a la seguridad, vamos a demostrar el siguiente teorema.

Teorema 1. *En el modelo del oráculo aleatorio para la función de hash H, si el esquema SKE es IND-CPA seguro, entonces el esquema sIBE que acabamos de construir también es IND-CPA seguro.*

Concretamente, para todo adversario $\mathcal{A}$ que ataca la seguridad IND-CPA del esquema sIBE con ventaja $\mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$, que haga q_H llamadas al oráculo aleatorio, q_e llamadas al oráculo de cifrado y q_k peticiones de clave secreta de usuario, podemos construir un adversario $\mathcal{B}$ que ataca la seguridad IND-CPA del esquema SKE con ventaja

$$\mathsf{Adv}^{\mathsf{SKE}}_{\mathcal{B}}(\lambda) \;\geq\; \left(1 - \frac{q_H}{2^\lambda}\right) \cdot \frac{1}{q_e + q_k + 1} \cdot \mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$$

Demostración. Sea $\mathcal{A}$ un adversario que ejecuta el experimento $\mathbf{Exp}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$, con acceso adicional a peticiones de evaluación de la función de hash H, y obtiene ventaja $\mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$. Construimos a continuación un adversario $\mathcal{B}$ que ejecuta el experimento $\mathbf{Exp}^{\mathsf{SKE}}_{\mathcal{B}}(\lambda)$.

En el primer paso de ese experimento, se escoge un bit aleatorio $b \xleftarrow{R} \{0,1\}$. En el segundo paso, se ejecuta $(\mathsf{pms}_{\mathsf{SKE}}, k^*) \leftarrow \mathsf{SKE.KG}(1^\lambda)$, y los parámetros públicos $\mathsf{pms}_{\mathsf{SKE}}$ se entregan al adversario $\mathcal{B}$. En este momento, $\mathcal{B}$ inicia la ejecución del experimento $\mathbf{Exp}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$, teniendo como adversario a $\mathcal{A}$, a quién le envía los parámetros públicos $\mathsf{pms} = (\mathsf{pms}_{\mathsf{SKE}}, H_{r.o.})$.

Aquí $H_{r.o.}$ quiere decir que no se incluye ninguna descripción de la función de hash H, sino que al ser la demostración en el modelo del oráculo aleatorio, lo que estamos suponiendo es que $H : \{0,1\}^\lambda \times \mathcal{ID} \to \mathcal{K}$ se comporta igual que una función elegida uniforme y aleatoriamente entre todas las funciones de $\{0,1\}^\lambda \times \mathcal{ID}$ a $\mathcal{K}$. Como es imposible describir una función aleatoria en una cantidad de memoria polinómica en λ, lo que se hace en la demostración es que $\mathcal{B}$ va a mantener en una tabla TAB_H algunas relaciones entrada/salida para la función H, y usará esa tabla para responder las q_H posibles peticiones de $\mathcal{A}$ de evaluaciones de H en entradas $(a, \mathsf{id}) \in \{0,1\}^\lambda \times \mathcal{ID}$.

Recordemos que $\mathcal{A}$, además de las peticiones de evaluación de H, puede pedir cifrados para mensajes e identidades de su

elección, y puede pedir claves secretas para identidades de su elección. $\mathcal{B}$ escoge de manera uniforme y aleatoria $\mathsf{msk} \in \{0,1\}^\lambda$ y creará entradas en la tabla TAB_H de dos maneras diferentes:

- la primera, cuando $\mathcal{A}$ haga una petición de evaluación de H en una entrada (a, id): (i) si $a = \mathsf{msk}$, entonces $\mathcal{B}$ acaba el experimento y devuelve un bit b' aleatorio; (ii) si $a \neq \mathsf{msk}$, entonces $\mathcal{B}$ escoge un valor $k \xleftarrow{R} \mathcal{K}$ aleatorio y uniforme, devuelve k a $\mathcal{A}$ y añade la entrada (a, id, k) a la tabla TAB_H. Lógicamente, si (a, id) ya estaba en la tabla, se devuelve el mismo valor k. La probabilidad de que pase (i) en toda la ejecución de $\mathbf{Exp}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$ es $\frac{q_H}{2^\lambda}$, negligible en el parámetro de seguridad λ.
- la segunda manera en que $\mathcal{B}$ actualiza la tabla TAB_H se da en los otros dos tipos de peticiones que puede hacer $\mathcal{A}$: de clave secreta $\mathsf{sk}_{\mathsf{id}}$ y de cifrado para parejas (m, id). En estas peticiones pueden aparecer como mucho $q_e + q_k$ identidades. Si añadimos la identidad id^* que $\mathcal{A}$ pedirá para el reto, en el paso 3 del experimento $\mathbf{Exp}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$, tenemos un total máximo de $L = q_e + q_k + 1$ identidades. Nuestro adversario $\mathcal{B}$ escoge aleatoriamente un índice $j \xleftarrow{R} \{1, 2, \ldots, L\}$. Ahora, para la i-ésima identidad id_i que aparece en una de estas peticiones de clave secreta o de cifrado, donde $i \in \{1, 2, \ldots, q_e + q_k\}$:
  - Si $i < j$ o bien $i > j$ y además $\mathsf{id}_i \neq \mathsf{id}_j$, entonces $\mathcal{B}$ escoge aleatoriamente $k_i \xleftarrow{R} \mathcal{K}$ y añade la entrada $(\mathsf{msk}, \mathsf{id}_i, k_i)$ en TAB_H. Si la petición era de clave secreta, se devuelve $\mathsf{sk}_{\mathsf{id}_i} = k_i$ a $\mathcal{A}$; si la petición (m, id_i) era de cifrado, se devuelve $C = \mathsf{SKE.Enc}(m, k_i)$.
  - Si $i = j$, enconces implícitamente $\mathcal{B}$ define la relación $H(\mathsf{msk}, \mathsf{id}_j) = k^*$. Si era una petición de clave secreta, $\mathcal{B}$ acaba el experimento y devuelve un bit b' aleatorio. Si era una petición (m, id_j) de cifrado, $\mathcal{B}$ envía el mensaje m a su propio oráculo de cifrado, dentro de su experimento $\mathbf{Exp}^{\mathsf{SKE}}_{\mathcal{B}}(\lambda)$. El resultado que obtiene es $C \leftarrow \mathsf{SKE.Enc}(m, k^*)$, y $\mathcal{B}$ reenvía a $\mathcal{A}$ ese texto cifrado. Las posibles peticiones posteriores de este segundo tipo en las que aparezca la identidad id_j se contestarán así.

Siempre que se haga una petición para una identidad ya solicitada anteriormente, la respuesta tiene que ser coherente con la información guardada en la tabla.

En algún momento de $\mathbf{Exp}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$, el adversario $\mathcal{A}$ escoge una identidad id^* y dos mensajes $m^{(0)}, m^{(1)}$. Si $\mathsf{id}^* \neq \mathsf{id}_j$, enconces $\mathcal{B}$ acaba el experimento y devuelve un bit b' aleatorio. Si $\mathsf{id}^* = \mathsf{id}_j$, entonces $\mathcal{B}$ envía los mismos mensajes $m^{(0)}, m^{(1)}$ como paso 3 del experimento $\mathbf{Exp}^{\mathsf{SKE}}_{\mathcal{B}}(\lambda)$.

A partir de aquí, $\mathcal{B}$ sigue contestando el resto de peticiones de $\mathcal{A}$, y en algún momento $\mathcal{A}$ devuelve un bit b'. Nuestro adversario $\mathcal{B}$ devuelve el mismo bit b'.

Observamos que si $\mathcal{B}$ ha tenido suerte y $\mathsf{id}_j = \mathsf{id}^*$, cosa que pasa con probabilidad al menos $\frac{1}{q_e + q_k + 1}$, entonces por definición de adversario exitoso $\mathcal{A}$, se cumpliría que $\mathcal{A}$ no ha hecho una petición de clave secreta para $\mathsf{id}^* = \mathsf{id}_j$. En ese caso, la probabilidad de que $b' = b$ es $\frac{1}{2} + \mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)$. En el resto de casos (en los que $\mathcal{B}$ ha parado el experimento y ha devuelto un bit aleatorio b'), tendremos que la probabilidad que $b' = b$ es exactamente $\frac{1}{2}$. Por tanto, con probabilidad

$$\rho \geq \left(1 - \frac{q_H}{2^\lambda}\right) \cdot \frac{1}{q_e + q_k + 1}$$

tenemos que $\mathcal{B}$ llega al final del experimento, y con probabilidad $1 - \rho$ tenemos que $\mathcal{B}$ acaba el experimento antes, con una salida aleatoria.

La probabilidad total que el bit b' devuelto por $\mathcal{B}$ cumpla $b' = b$ es

$$(1-\rho) \cdot \frac{1}{2} \; + \; \rho \cdot \left(\frac{1}{2} + \mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda)\right) =$$

$$= \frac{1}{2} + \rho \cdot \mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda).$$

Por tanto, tenemos que

$$\mathsf{Adv}^{\mathsf{SKE}}_{\mathcal{B}}(\lambda) = \rho \cdot \mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda) \geq$$

$$\geq \left(1 - \frac{q_H}{2^\lambda}\right) \cdot \frac{1}{q_e + q_k + 1} \cdot \mathsf{Adv}^{\mathsf{sIBE}}_{\mathcal{A}}(\lambda),$$

como queríamos demostrar. □

IV. Conclusiones y Trabajo Futuro

La noción de cifrado basado en identidades ha sido extendida a otras nociones aún más difíciles de conseguir. Entre ellas, la que tal vez ha atraído más atención debido a sus potenciales aplicaciones en sistemas reales es la de cifrado basado en atributos [7], [8]. Puesto que la existencia de un sistema de cifrado basado en atributos (ABE) seguro implica la existencia de un sistema IBE seguro [9], los resultados de imposibilidad para IBE [3], [4] implican que es imposible diseñar un sistema ABE seguro y eficiente que funcione en grupos cíclicos con logaritmo discreto difícil, sin emparejamientos bilineales.

Igual que hemos hecho en este trabajo, se podrían considerar sistemas ABE simétricos: estos sistemas tendrían básicamente las mismas aplicaciones prácticas que los sistemas IBE simétricos, y además otras aplicaciones más teóricas: la transformación de [10] que transforma un sistema ABE en un sistema de delegación verificable de computación de funciones funcionaría de hecho tomando como punto de partida un sistema ABE simétrico.

Desafortunadamente, las ideas de la construcción de sIBE que hemos dado en este trabajo no llevan a un sistema ABE simétrico eficiente, puesto que habría que generar una clave simétrica para cada posible subconjuto de atributos; en consecuencia, el tamaño de las claves de usuario y de los textos cifrados dependería exponencialmente del tamaño del conjunto total de atributos del sistema. Por tanto, el problema de diseñar sistemas ABE simétricos eficientes y que funcionen en un escenario criptográfico clásico (como RSA o DDH) sigue abierto y será tema de nuestro trabajo futuro.

Agradecimientos

El trabajo del autor está parcialmente subvencionado por el Ministerio de Ciencia e Innovación, mediante el proyecto PID2019-109379RB-I00.

REFERENCIAS

[1] A. Shamir. Identity-based cryptosystems and signature schemes. *Proc. of Crypto'84*, LNCS **196**, Springer-Verlag, pp. 47–53 (1984)

[2] D. Boneh, M.K. Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, vol. **32 (3)**, pp. 586–615 (2003)

[3] D. Boneh, P.A. Papakonstantinou, C. Rackoff, Y. Vahlis, B. Waters. On the impossibility of basing identity based encryption on trapdoor permutations. *Proc. of FOCS'08*, ACM Press, pp. 283–292 (2008)

[4] P.A. Papakonstantinou, C. Rackoff, Y. Vahlis. How powerful are the DDH hard groups? *Technical report* (2012), available at `https://eprint.iacr.org/2012/653`

[5] N. Döttling, S. Garg. Identity-based encryption from the Diffie-Hellman Assumption. *Proc. of Crypto'17*, LNCS **10401**, Springer-Verlag, pp. 537–569 (2017)

[6] S. Tessaro, D.A. Wilson. Bounded-collusion identity-based encryption from semantically-secure public-key encryption: Generic constructions with short ciphertexts. *Proc. of PKC'14*, LNCS **8383**, Springer-Verlag, pp. 257–274 (2014)

[7] V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. *Proc. of Computer and Communications Security, CCS'06*, ACM Press, pp. 89–98 (2006)

[8] J. Bethencourt, A. Sahai, B. Waters. Ciphertext-policy attribute-based encryption. *Proc. of IEEE Symposium on Security and Privacy*, IEEE Society Press, pp. 321–334 (2007)

[9] J. Herranz. Attribute-based encryption implies identity-based encryption. *Applicable Algebra in Engineering, Communication and Computing*, Vol. **27 (1)**, pp. 17–57 (2016)

[10] B. Parno, M. Raykova, V. Vaikuntanathan. How to delegate and verify in public: verifiable computation from attribute-based encryption. *Proc. of TCC'12*, LNCS **7194**, Springer-Verlag, pp. 422–439 (2012)