Firmas Digitales con Verificación Distribuida en el Modelo de Seguridad Estándar

Javier Herranz, Alexandre Ruiz, Germán Sáez
Dept. de Matemàtica Aplicada IV, Universitat Politècnica de Catalunya
{jherranz,aruiz,germam}@ma4.upc.edu

***Abstract*—Las firmas digitales con verificación distribuida protegen en cierta manera el nivel de anonimato o privacidad del firmante, ya que un subconjunto autorizado de usuarios deben colaboran para verificar la (in)validez de una firma. En trabajos anteriores se propusieron esquemas de este tipo pero que o no alcanzaban el nivel máximo de seguridad o bien lo hacían en el modelo del oráculo aleatorio. Proponemos aquí el primer esquema de firma digital con verificación distribuida que consigue seguridad máxima, en términos de infalsificabilidad y privacidad, y con seguridad demostrable en el modelo de computación estándar.**

***Index Terms*—Firma digital, compartición de secretos, modelo estándar, infalsificabilidad, privacidad**

I. Introducción

En algunas situaciones la propiedad de verificación universal en una firma digital puede ser no deseable, si el firmante desea un cierto nivel de anonimato o de privacidad. Como por ejemplo, en firmas sobre transacciones bancarias, documentos médicos o simplemente diferente información personal. Una posible solución a este problema consiste en exigir la colaboración de varios usuarios en el protocolo de verificación. A este tipo de esquemas los llamaremos esquemas de firma con verificación distribuida; dichos esquemas tienen aplicaciones en situaciones reales como subastas o votaciones electrónicas. Por ejemplo, en una subasta digital, los participantes pueden usar estos esquemas para firmar sus pujas; el proceso de autenticación (mediante la verificación de esas firmas, distribuida entre unas cuantas autoridades) se realizará sólo para la(s) puja(s) más alta(s).

Hay diferentes trabajos en la literatura sobre tipos de firma parecidas, pero tratan el problema desde otra perspectiva. En [11] o [13] se utilizan estructuras de acceso umbral, sin ningún análisis formal de seguridad. En [10] el firmante elige los verificadores y la estructura de acceso justo en el momento de firmar, lo que aumenta el coste computacional y la longitud de las firmas. Las propiedades de seguridad (infalsificabilidad y privacidad) que debe satisfacer un esquema de firma con verificación distribuida se definen de manera formal por primera vez en [8], donde también se propone un esquema concreto. Dicho esquema no alcanza el máximo nivel de seguridad respecto a la propiedad de privacidad, cosa que sí consigue el esquema propuesto en [9]. La seguridad de este último esquema, sin embargo, se demuestra en el modelo (heurístico, no real) del oráculo aleatorio, donde algunas funciones de hash se modelan como funciones completamente aleatorias. En este trabajo proponemos un nuevo esquema de firma con verificación distribuida, que alcanza el nivel máximo de seguridad, y lo hace de manera demostrable en el módelo estandar (es decir, sin la hipótesis del oráculo aleatorio).

La definición de los esquemas de firma con verificación distribuida junto con las dos propiedades de seguridad requeridas se encuentran respectivamente en las Secciones II y III. El diseño del nuevo esquema, que se presenta en la Sección IV, sigue las ideas del esquema de cifrado distribuido de Boneh, Boyen y Halevi [1], combinándolo con dos esquemas de firma digital genéricos. En la Sección V, demostraremos formalmente las propiedades de seguridad, en el modelo estándar, por reducción al problema Bilineal Decisional de Diffie-Hellman y a la seguridad de los esquemas de firma subyacentes.

A. Preliminares

Dado un parámetro de seguridad $\lambda \in \mathbb{N}$, un número primo q de λ bits y un grupo cíclico $\mathbb{G} = \langle g \rangle$ de orden primo q, decimos que el grupo $\mathbb{G}$ es *bilineal* si existe otro grupo $\mathbb{G}_T$ del mismo orden q y una aplicación $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ que satisface las siguientes propiedades:

1) $e(\cdot, \cdot)$ puede ser calculado eficientemente (en tiempo polinomial en λ)
2) $e(g, g)$ es un generador de $\mathbb{G}_T$
3) se verifica $e(g^a, g^b) = e(g, g)^{ab}$, donde $a, b \in \mathbb{Z}_q$.

II. Esquemas de firma con verificación distribuida

Un esquema Σ de firma con verificación distribuida consiste en cuatro protocolos probabilísticos y de tiempo de ejecución polinómico:

1) ***Ini*.** La entrada es un parámetro de seguridad λ. La salida son unos parámetros públicos `params` utilizados en todo el esquema.

$$\Sigma.\mathsf{Ini}(1^\lambda) = \texttt{params}$$

2) ***Gen_Cla*.** Este protocolo utiliza dos algoritmos. El primero corresponde al firmante A que obtendrá un par de claves (sk_A, pk_A), donde sk_A es la clave privada para firmar y pk_A es la correspondiente clave pública. El segundo algoritmo corresponde a un conjunto $\mathcal{B}$ de n verificadores, que tiene asociada una estructura de acceso (monótona creciente) $\Gamma_{\mathcal{B}} \subset 2^{\mathcal{B}}$, que contiene los subconjuntos autorizados a verificar. Estos usuarios obtendrán cierta información privada $\{sk_j\}_{j \in \mathcal{B}}$ que va a ser usada más tarde en el proceso de verificación

distribuida, y cierto valor público $pk_{\mathcal{B}}$ común para el conjunto $\mathcal{B}$. El proceso de generación de claves para el colectivo $\mathcal{B}$ puede ser ejecutado por una tercera autoridad de confianza o de manera conjunta por ellos mismos, usando técnicas conocidas [5].

$$\Sigma.\mathsf{GC}(\mathtt{params}, A, \text{'individual'}) = (sk_A, pk_A)$$

$$\Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}, \Gamma_{\mathcal{B}}, \text{'colectivo'}) = (\{sk_j\}_{j\in\mathcal{B}}, pk_{\mathcal{B}})$$

3) ***Firm.*** Este algoritmo es ejecutado por el firmante A; toma como entrada un mensaje m, su clave privada sk_A y la clave pública asociada a un grupo $\mathcal{B}$ de verificadores, y da como salida una firma σ del mensaje.

$$\Sigma.\mathsf{Firm}(\mathtt{params}, m, pk_{\mathcal{B}}, sk_A) = \sigma$$

4) ***Ver_Dist.*** Dado $B \in \Gamma_{\mathcal{B}}$ un subconjunto autorizado de verificadores, este protocolo toma como entrada un mensaje m, una firma σ, la clave pública pk_A y los fragmentos sk_j de los usuarios $j \in B$. La salida será 1 si σ es una firma válida de m y 0 en el caso contrario.

$$\Sigma.\mathsf{Ver}(\mathtt{params}, m, \sigma, pk_A, B, \{sk_j\}_{j\in B}) = 1 \text{ ó } 0$$

III. Modelo de seguridad

Las propiedades de seguridad que se exigen a un esquema de firma con verificación distribuida son las de *infalsificabilidad* y *privacidad*. En nuestro esquema al adversario se le permite hacer peticiones de firma y verificación para diferentes usuarios, mensajes y firmas.

Además se le permite corromper al mayor número de participantes posibles (con la excepción de los usuarios que sean objetivo de su ataque en cada caso). En particular, la infalsificabilidad se alcanza incluso cuando el adversario conoce toda la información secreta de todos los participantes con la excepción del firmante que quiere atacar. Por otra parte, la privacidad se consigue incluso en el caso que el adversario conozca las claves secretas de todos los posibles firmantes y de un subconjunto no autorizado de verificadores. Este nivel de seguridad recibe en inglés el nombre de *insider security*.

A. Infalsificabilidad

La *infalsificabilidad existencial contra ataques de mensaje escogido* [7] es una noción de seguridad común en los esquemas de firma y en general en cualquier primitiva criptográfica que requiere cierto grado de autenticidad o no repudio. La idea es que cualquier atacante tiene probabilidad despreciable [1] en falsificar una firma válida de un usuario (del cual no conoce su clave secreta), incluso si el atacante puede obtener previamente otros pares (mensaje,firma) válidos, para mensajes y conjuntos de verificadores que él escoge adaptativamente.

Dado un parámetro de seguridad λ, esta propiedad se formaliza con el siguiente juego en el que un retador externo reta a un atacante $\mathcal{A}_{\mathsf{INF}}$ para que intente falsificar una firma válida nueva:

[1]Formalmente, decimos que una función f es *despreciable* (o *negligible*, en inglés) en k si existe un polinomio $p(\cdot)$ y un valor entero positivo k_0 tal que $f(k) \leq 1/p(k)$ para todo $k \geq k_0$. Usualmente, se escribe $f(k) = \mathrm{negl}(k)$ para las funciones f despreciables en k.

1) El retador ejecuta $\mathtt{params} \leftarrow \Sigma.\mathsf{Ini}(1^\lambda)$ y da todos los valores obtenidos junto con una estructura de acceso Γ a $\mathcal{A}_{\mathsf{INF}}$.
2) $\mathcal{A}_{\mathsf{INF}}$ escoge un participante $A^\star$ para ser atacado. El retador ejecuta $(sk_{A^\star}, pk_{A^\star}) \leftarrow \Sigma.\mathsf{GC}(\mathtt{params}, A^\star, \text{'individual'})$, se guarda $sk_{A^\star}$ y le da $pk_{A^\star}$ a $\mathcal{A}_{\mathsf{INF}}$.
3) [Generación de nuevas claves] El atacante puede ejecutar $(sk_A, pk_A) \leftarrow \Sigma.\mathsf{GC}(\mathtt{params}, A, \text{'individual'})$ para firmantes $A \neq A^\star$ de su elección, y también puede ejecutar $\Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}, \Gamma_{\mathcal{B}}, \text{'colectivo'}) = (\{sk_j\}_{j\in\mathcal{B}}, pk_{\mathcal{B}})$ para conjuntos $\mathcal{B}$ de su elección.
4) [Peticiones firma] $\mathcal{A}_{\mathsf{INF}}$ puede escoger, de manera adaptativa, tuplas $(m_\ell, pk_{\mathcal{B}_\ell})$ y enviarlas a un oráculo de firma para el firmante $A^\star$. $\mathcal{A}_{\mathsf{INF}}$ obtiene como respuesta las firmas $\sigma_\ell \leftarrow \Sigma.\mathsf{Firm}(\mathtt{params}, m_\ell, pk_{\mathcal{B}_\ell}, sk_{A^\star})$.
5) [Falsificación] En un cierto momento, $\mathcal{A}_{\mathsf{INF}}$ publica un par $(m^\star, \sigma^\star)$ y una clave $pk_{\mathcal{B}^\star}$ para un conjunto $\mathcal{B}^\star$ y una estructura de acceso $\Gamma_{\mathcal{B}^\star}$. El atacante $\mathcal{A}_{\mathsf{INF}}$ gana el juego si $(m^\star, \sigma^\star) \neq (m_\ell, \sigma_\ell)$, para toda firma obtenida durante el ataque, y además $\Sigma.\mathsf{Ver}(\mathtt{params}, m^\star, \sigma^\star, pk_{A^\star}, B, \{sk_j\}_{j\in B}) = 1$, para algún subconjunto $B \in \Gamma_{\mathcal{B}^\star}$.

La ventaja de un adversario $\mathcal{A}_{\mathsf{INF}}$ en romper la infalsificabilidad de un esquema de firma con verificación distribuida se define como

$$\mathsf{Vent}_{\mathcal{A}_{\mathsf{INF}}}(\lambda) = \Pr[\mathcal{A}_{\mathsf{INF}} \text{ gana el juego}].$$

Definición 1. Un esquema de firma con verificación distribuida Σ es infalsificable si para cualquier adversario $\mathcal{A}_{\mathsf{INF}}$ de tiempo polinómico, el valor $\mathsf{Vent}_{\mathcal{A}_{\mathsf{INF}}}(\lambda)$ es despreciable con respecto al parámetro de seguridad λ.

B. Privacidad

Intuitivamente, en una firma digital con verificación distribuida se requiere que un atacante que corrompa a un subconjunto de usuarios no autorizado, no pueda obtener ninguna información sobre la (in)validez de las firmas calculadas por el usuario A. Para formalizar exactamente qué quiere decir 'no obtener ninguna información', se adapta el concepto de seguridad semántica [6]. De manera informal, dados dos mensajes escogidos por un atacante, y una firma válida para uno de estos mensajes, el adversario no debe ser capaz de distinguir qué mensaje ha sido firmado con probabilidad significativamente mayor que $1/2$ (respuesta aleatoria).

Dado un parámetro de seguridad λ, esta idea intuitiva se formaliza con el siguiente juego de indistinguibilidad donde un atacante $\mathcal{A}_{\mathsf{IND}}$ intenta ganar a un retador externo:

1) El retador ejecuta $\mathtt{params} \leftarrow \Sigma.\mathsf{Ini}(1^\lambda)$ y da todos los valores obtenidos a $\mathcal{A}_{\mathsf{IND}}$.
2) $\mathcal{A}_{\mathsf{IND}}$ escoge un conjunto de verificadores $\mathcal{B}^\star$, una estructura de acceso $\Gamma_{\mathcal{B}^\star} \subset 2^{\mathcal{B}^\star}$ y un subconjunto no autorizado $\widetilde{B} \notin \Gamma_{\mathcal{B}^\star}$, cuyos usuarios puede corromper. El retador ejecuta el protocolo $(\{sk_j\}_{j\in\mathcal{B}^\star}, pk_{\mathcal{B}^\star}) \leftarrow \Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}^\star, \Gamma_{\mathcal{B}^\star}, \text{'colectivo'})$, da al atacante $\mathcal{A}_{\mathsf{IND}}$ los valores $pk_{\mathcal{B}^\star}$ y $\{sk_j\}_{j\in\widetilde{B}}$, y mantiene el resto

de valores sk_j en secreto. Nótese que consideramos adversarios *estáticos* que eligen el subconjunto de usuarios corruptos $\widetilde{B}$ al principio del ataque.

3) [Generación de nuevas claves] El atacante puede ejecutar $(sk_A, pk_A) \leftarrow \Sigma.\mathsf{GC}(\mathtt{params}, A, \text{'individual'})$ para firmantes A de su elección, y también puede ejecutar $\Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}, \Gamma_{\mathcal{B}}, \text{'colectivo'}) = (\{sk_j\}_{j\in\mathcal{B}}, pk_{\mathcal{B}})$ para parejas $(\mathcal{B}, \Gamma_{\mathcal{B}}) \neq (\mathcal{B}^\star, \Gamma_{\mathcal{B}^\star})$ de su elección.
4) [Peticiones verificación] $\mathcal{A}_{\mathsf{IND}}$ escoge diferentes tuplas $(m_\ell, \sigma_\ell, pk_{A_\ell})$ para firmantes A_ℓ de su elección y hace peticiones, de manera adaptativa, a un oráculo de verificación para estas firmas, con conjunto de verificadores $\mathcal{B}^\star$. $\mathcal{A}_{\mathsf{IND}}$ obtiene como respuesta toda la información emitida durante la ejecución del protocolo $\Sigma.\mathsf{Ver}(\mathtt{params}, m_\ell, \sigma_\ell, pk_{A_\ell}, \mathcal{B}^\star, \{sk_j\}_{j\in\mathcal{B}^\star})$.
5) $\mathcal{A}_{\mathsf{IND}}$ escoge dos mensajes m_0, m_1 de la misma longitud y un firmante $A^\star$ con claves $(sk_{A^\star}, pk_{A^\star})$, que $\mathcal{A}_{\mathsf{IND}}$ envía al retador.
6) [Desafío] El retador escoge un bit aleatorio $b \in \{0,1\}$ y ejecuta $\sigma^\star \leftarrow \Sigma.\mathsf{Firm}(\mathtt{params}, m_b, pk_{\mathcal{B}^\star}, sk_{A^\star})$. La firma resultante $\sigma^\star$ se envía a $\mathcal{A}_{\mathsf{IND}}$.
7) [Más peticiones] El paso 4 es repetido, con la restricción que las tuplas $(m_i, \sigma^\star, pk_{A^\star})$ no pueden ser enviadas al oráculo de verificación, para $i = 0, 1$.
8) Finalmente, $\mathcal{A}_{\mathsf{IND}}$ devuelve un bit $b' \in \{0,1\}$.

Decimos que $\mathcal{A}_{\mathsf{IND}}$ gana el juego si $b' = b$. La *ventaja* de un tal adversario (estático) $\mathcal{A}_{\mathsf{IND}}$ en romper la privacidad de un esquema de firma con verificación distribuida se define como

$$\mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda) = |2\Pr[b' = b] - 1| .$$

Definición 2. Un esquema de firma con verificación distribuida Σ satisface la propiedad de privacidad si para cualquier adversario $\mathcal{A}_{\mathsf{IND}}$ de tiempo polinómico, el valor $\mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda)$ es despreciable con respecto al parámetro de seguridad λ.

IV. El esquema propuesto

El diseño de nuestro esquema de firma con verificación distribuida está inspirado en el esquema de cifrado distribuido propuesto por Boneh, Boyen y Halevi [1] cuya seguridad CCA es en el modelo estándar, adaptando además la estrategia propuesta por Canetti, Halevi y Katz en [4]. Para ello, el firmante de nuestro esquema utilizará un esquema de firma Θ individual infalsificable (existencialmente contra ataques de mensaje escogido), seguido de un esquema de firma *de un solo uso* $\tilde{\Theta}$ (en inglés, *one-time*, seguro contra atacantes que pueden hacer como máximo una petición al oráculo de firma).

Detallamos a continuación los protocolos que componen nuestro esquema de firma con verificación distribuida Σ, para un firmante A y un conjunto $\mathcal{B} = \{1, \ldots, n\}$ de n verificadores.

1) ***Ini.*** $\Sigma.\mathsf{Ini}(1^\lambda)$.
Dado un parámetro de seguridad $\lambda \in \mathbb{N}$, se escoge un número primo q tal que $|q| = \lambda$. Se escogen también un grupo cíclico bilineal $\mathbb{G} = \langle g \rangle$ de orden primo q, dos generadores aleatorios $h, g_2 \in \mathbb{G}$ y una aplicación bilineal $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ para un grupo $\mathbb{G}_T$. Posteriormente se escoge un esquema de firma $\Theta = (\Theta.\mathsf{GC}, \Theta.\mathsf{Firm}, \Theta.\mathsf{Ver})$ infalsificable, un esquema de firma de un solo uso $\tilde{\Theta} = (\tilde{\Theta}.\mathsf{GC}, \tilde{\Theta}.\mathsf{Firm}, \tilde{\Theta}.\mathsf{Ver})$ y se publican dos funciones hash $H_0 : \{0,1\}^* \to \mathbb{Z}_q$, $H_1 : \{0,1\}^* \times \mathbb{G}_T \to \{0,1\}^*$. Del protocolo se obtienen los valores $\mathtt{params} = (q, \mathbb{G}, g, h, g_2, \mathbb{G}_T, e, \Theta, \tilde{\Theta}, H_0, H_1)$.
2) ***Gen_Cla.*** $\Sigma.\mathsf{GC}(\mathtt{params}, A, \text{'individual'})$
$\Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}, \Gamma_{\mathcal{B}}, \text{'colectivo'})$.
Para el firmante A, se ejecuta el protocolo de generación de claves $\Theta.\mathsf{GC}$ cuyas claves de firma y verificación resultantes serán la clave secreta y pública respectivamente del usuario A. Es decir, $(x_A, y_A) \leftarrow \Theta.\mathsf{GC}(1^\lambda)$.
Para el colectivo $\mathcal{B}$ de n usuarios se calcula el valor $y_{\mathcal{B}} = g^{\alpha_{\mathcal{B}}}$ para un valor aleatorio $\alpha_{\mathcal{B}} \in \mathbb{Z}_q^*$ que es desconocido para los miembros de $\mathcal{B}$. Cada verificador j de $\mathcal{B}$ recibe un fragmento $\alpha_{\mathcal{B},j}$ de $\alpha_{\mathcal{B}}$, correspondiente a un esquema de compartición de secretos de espacio vectorial [3] para la estructura de acceso $\Gamma_{\mathcal{B}}$. Es decir, para cada conjunto autorizado $B \in \Gamma_{\mathcal{B}}$ existen coeficientes $\{\lambda_j^B\}_{j\in B}$ tales que $\sum_{j\in B} \lambda_j^B \alpha_{\mathcal{B},j} = \alpha_{\mathcal{B}}$. El valor público que se obtiene del protocolo es $y_{\mathcal{B}}$, mientras que cada usuario $j \in \mathcal{B}$ se guarda secretamente su fragmento $x_{\mathcal{B},j} = g_2^{\alpha_{\mathcal{B},j}}$ del secreto $x_{\mathcal{B}} = g_2^{\alpha_{\mathcal{B}}}$.
[Para dotar de robustez al esquema es necesario publicar los valores $D_{\mathcal{B},j} = g^{\alpha_{\mathcal{B},j}}$, para $j = 1, \ldots, n$.]
3) ***Firm.*** $\Sigma.\mathsf{Firm}(\mathtt{params}, m, y_{\mathcal{B}}, x_A)$, con $m \in \{0,1\}^*$.
   a) Ejecuta $\tilde{\Theta}.\mathsf{GC}(1^\lambda) \to (\tilde{\mathsf{sk}}, \tilde{\mathsf{vk}})$ del esquema de firma de un solo uso para obtener un par de claves efímeras de firma y verificación.
   b) Deriva $\mathsf{id} = H_0(\tilde{\mathsf{vk}})$, que puede ser pensado como un elemento de $\mathbb{Z}_q$.
   c) Elige un valor aleatorio $s \in \mathbb{Z}_q^*$
   d) Calcula $C_1 = g^s$, $C_2 = H_1(m, e(y_{\mathcal{B}}, g_2)^s)$ y $C_3 = \left(y_{\mathcal{B}}^{\mathsf{id}} \cdot h\right)^s$.
   e) Ejecuta $\Theta.\mathsf{Firm}(x_A,\ C_1||C_2||C_3||y_A||y_{\mathcal{B}}||\tilde{\mathsf{vk}}) \to \theta$ para obtener la firma θ.
   f) Ejecuta $\tilde{\Theta}.\mathsf{Firm}(\tilde{\mathsf{sk}},\ C_1||C_2||C_3||y_A||y_{\mathcal{B}}||\theta) \to \tilde{\theta}$.
   g) Devuelve la firma $\sigma = (C_1, C_2, C_3, \mathsf{vk}, \theta, \tilde{\theta})$.
4) ***Ver_Dist.*** $\Sigma.\mathsf{Ver}(\mathtt{params}, m, \sigma, y_A, B, \{\alpha_{\mathcal{B},j}\}_{j\in B})$.
Si los participantes de un subconjunto autorizado $B \in \Gamma_{\mathcal{B}}$ quieren cooperar para verificar la firma $\sigma = (C_1, C_2, C_3, \tilde{\mathsf{vk}}, \theta, \tilde{\theta})$ del mensaje m creada por A, ejecutan los pasos siguientes.
   a) Cada verificador $j \in B$ ejecuta $\tilde{\Theta}.\mathsf{Ver}(\tilde{\mathsf{vk}}, C_1||C_2||C_3||y_A||y_{\mathcal{B}}||\theta, \tilde{\theta})$. Si el resultado es 0, difunde $(j, \perp)$. El símbolo $\perp$ denota algún tipo de error, local o global, producido a la hora de verificar.
   b) Cada verificador $j \in B$ ejecuta $\Theta.\mathsf{Ver}(y_A, C_1||C_2||C_3||y_A||y_{\mathcal{B}}||\tilde{\mathsf{vk}}, \theta)$. Si el resultado es 0, dufunde $(j, \perp)$.
   c) Cada verificador $j \in B$ deriva $\mathsf{id} = H_0(\tilde{\mathsf{vk}})$ y comprueba si $e(C_3, g) = e(y_{\mathcal{B}}^{\mathsf{id}} \cdot h, C_1)$. Si la igualdad no se verifica, j difunde $(j, \perp)$.
   d) Cada verificador $j \in B$ elige aleatoriamente $r_j \in \mathbb{Z}_q^*$ y difunde la tupla $(j, \omega_{0,j}, \omega_{1,j})$, donde $\omega_{0,j} =$

$\alpha_{\mathcal{B},j} \cdot (y_{\mathcal{B}}^{\mathsf{id}} \cdot h)^{r_j}$ y $\omega_{1,j} = g^{r_j}$.
[Si se requiere robustez, la validez de esta tupla puede ser verificada públicamente comprobando si $e(\omega_{0,j}, g) = e(D_{\mathcal{B},j}, g_2) \cdot e(y_{\mathcal{B}}^{\mathsf{id}} \cdot h, \omega_{1,j})$.]

e) Si no hay fragmentos válidos provenientes de un subconjunto autorizado $B \in \Gamma_{\mathcal{B}}$, se para y se devuelve $\perp$ como resultado erróneo de verificación. En caso contrario, a partir de las tuplas válidas $\{(j, \omega_{0,j}, \omega_{1,j})\}_{j \in B}$, diferentes de $(j, \perp)$, se calculan los coeficientes $\lambda_j^B \in \mathbb{Z}_q$ definidos por el esquema de compartición de secretos de espacio vectorial.

f) Se calcula $\omega_0 = \prod_{j \in B} \omega_{0,j}^{\lambda_j^B}$ y $\omega_1 = \prod_{j \in B} \omega_{1,j}^{\lambda_j^B}$.
[Nótese que $\omega_0 = x_{\mathcal{B}} \cdot (y_{\mathcal{B}}^{\mathsf{id}} \cdot h)^{\tilde{r}}$ y $\omega_1 = g^{\tilde{r}}$, donde $\tilde{r} = \sum_{j \in B} \lambda_j^B r_j$.]

g) Finalmente, se devuelve 1 si se cumple la igualdad $C_2 = H_1\left(m, \frac{e(C_3, \omega_1)}{e(C_1, \omega_0)}\right)$ y se devuelve 0 en caso contrario.

En [2], [12] podemos encontrar ejemplos de esquemas de firma que pueden usarse para instanciar Θ y $\tilde{\Theta}$ porque cumplen las propiedades de seguridad requeridas en el modelo estándar.

Nótese que el protocolo de verificación distribuida *Ver_Dist* puede ser separado en dos partes: Los pasos de autenticación a)-c) corresponden a la verificación pública (no hay información secreta) y pueden ser ejecutados por cualquier usuario (individualmente). Los pasos d)-g) corresponden al procedimiento de verificación secreta (la información pública y_A del firmante no es utilizada), que requiere la colaboración de usuarios en un conjunto autorizado.

En este sentido, la primera parte puede ser ejecutada por una entidad $T \notin \mathcal{B}$, que rechazaría firmas inválidas y borraría (o guardaría privadamente) la identidad del firmante tras una ejecución correcta. Después de esto, la identidad del firmante permanece oculta durante el resto del proceso de verificación. Esta propiedad puede ser de interés en aplicaciones que requieren cierto nivel de privacidad o anonimidad, como las subastas o las votaciones electrónicas.

V. Análisis de seguridad

En este apartado demostramos que el esquema de firma con verificación distribuida propuesto en la sección anterior satisface las propiedades definidas en la Sección III. Las demostraciones son el modelo estándar, es decir, sin asumir la existencia de oráculos aleatorios.

La seguridad de nuestro esquema está basada por una parte en la seguridad de los esquemas de firma subyacentes Θ y $\tilde{\Theta}$, y por otra parte en la dificultad de resolver el problema *Bilineal Decisional Diffie-Hellman (BDDH)*. El problema BDDH consiste en distinguir tuplas de la forma $(g, g^a, g^b, g^c, e(g,g)^{abc})$ de tuplas de la forma (g, g^a, g^b, g^c, T), para valores aleatorios $a, b, c \in \mathbb{Z}_q^*$ y $T \in \mathbb{G}_T$. Para un algoritmo en tiempo polinómico $\mathcal{A}^{BDDH}$ de este problema, definimos la ventaja $\mathsf{Vent}_{\mathcal{A}^{BDDH}}(\lambda) = |\Pr[\mathcal{A}^{BDDH}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 0] - \Pr[\mathcal{A}^{BDDH}(g, g^a, g^b, g^c, T) = 0]|$. La *hipótesis Bilineal Decisional Diffie-Hellman* asume que el problema BDDH es difícil de resolver, es decir que $\mathsf{Vent}_{\mathcal{A}^{BDDH}}(\lambda)$ es despreciable en λ.

A. Infalsificabilidad

La infalsificabilidad del nuevo esquema se basará en la infalsificabilidad de los esquemas de firma Θ y $\tilde{\Theta}$. La demostración es en el modelo estándar, sin hacer ninguna hipótesis adicional sobre el comportamiento de las funciones de hash.

Teorema 1. Sea $\lambda \in \mathbb{N}$ un parámetro de seguridad. Para cualquier atacante $\mathcal{A}_{\mathsf{INF}}$ contra la infalsificabilidad de nuestro esquema de firma con verificación distribuida, que hace Q peticiones de firma, existe un atacante $\mathcal{F}_\Theta$ contra Θ o un atacante $\mathcal{F}_{\tilde{\Theta}}$ contra $\tilde{\Theta}$, tales que

$$\mathsf{Vent}_{\mathcal{F}_\Theta}(\lambda) + Q \cdot \mathsf{Vent}_{\mathcal{F}_{\tilde{\Theta}}}(\lambda) \geq \mathsf{Vent}_{\mathcal{A}_{\mathsf{INF}}}(\lambda).$$

Demostración. Asumiendo que tenemos un atacante $\mathcal{A}_{\mathsf{INF}}$ que tiene ventaja $\mathsf{Vent}_{\mathcal{A}_{\mathsf{INF}}}(\lambda)$ en romper la infalsificabilidad de nuestro esquema de firma, vamos a construir un atacante $\mathcal{F}_\Theta$ contra el esquema de firma Θ, que va a ir ejecutando a su vez el atacante $\mathcal{A}_{\mathsf{INF}}$ como subrutina, simulando su entorno y respondiendo sus peticiones. $\mathcal{F}_\Theta$ recibe como entrada una clave de verificación vk obtenida al ejecutar $\Theta.\mathsf{GC}(1^\lambda) \rightarrow (\mathsf{sk}, \mathsf{vk})$, y tiene acceso a un oráculo de firma $\Theta.\mathsf{Firm}(\mathsf{sk}, \cdot)$ para mensajes de su elección. El objetivo de $\mathcal{F}_\Theta$ es encontrar una firma válida $(M^\star, \theta^\star)$ tal que $\Theta.\mathsf{Ver}(\mathsf{vk}, M^\star, \theta^\star) \rightarrow 1$.

Inicialización de $\mathcal{A}_{\mathsf{INF}}$. El protocolo $\Sigma.\mathsf{Ini}(1^\lambda)$ es ejecutado por $\mathcal{F}_\Theta$: éste da a $\mathcal{A}_{\mathsf{INF}}$ los valores $\texttt{params} = (q, \mathbb{G}, g, h, g_2, \mathbb{G}_T, e, \Theta, \tilde{\Theta}, H_0, H_1)$. Aquí las funciones hash $H_0 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ y $H_1 : \{0,1\}^* \times \mathbb{G}_T \rightarrow \{0,1\}^*$ son elegidas arbitrariamente por $\mathcal{F}_\Theta$.

Para simular la ejecución del protocolo $\Sigma.\mathsf{GC}(\texttt{params}, A^\star, \text{`individual'})$, para el firmante $A^\star$ escogido por $\mathcal{A}_{\mathsf{INF}}$, el algoritmo $\mathcal{F}_\Theta$ define la clave pública de $A^\star$ como $y_{A^\star} = \mathsf{vk}$ y se la envía a $\mathcal{A}_{\mathsf{INF}}$.

Generación de nuevas claves. El atacante $\mathcal{A}_{\mathsf{INF}}$ puede generar libremente nuevas claves públicas y secretas para otros firmantes $A \neq A^\star$ y para colectivos $(\mathcal{B}, \Gamma_{\mathcal{B}})$ de verificadores de su elección.

Peticiones firma. Cuando $\mathcal{A}_{\mathsf{INF}}$ solicita firmas válidas para mensajes m_ℓ y claves públicas $y_{\mathcal{B}_\ell}$ de su elección, donde el firmante es $A^\star$ y $\mathcal{B}_\ell$ es el colectivo de verificadores, $\mathcal{F}_\Theta$ simula y devuelve firmas σ_ℓ, de la siguiente manera. Primero ejecuta los pasos a), b), c) y d) del protocolo de firma $\Sigma.\mathsf{Firm}(\texttt{params}, m_\ell, y_{\mathcal{B}_\ell}, x_{A^\star})$, obteniendo los valores $\tilde{\mathsf{sk}}_\ell, \tilde{\mathsf{vk}}_\ell, C_{1,\ell}, C_{2,\ell}, C_{3,\ell}$. Después $\mathcal{F}_\Theta$ hace una petición de firma a su oráculo para el mensaje $M_\ell = C_{1,\ell}||C_{2,\ell}||C_{3,\ell}||y_{A^\star}||y_{\mathcal{B}_\ell}||\tilde{\mathsf{vk}}_\ell$ y obtiene como respuesta una firma válida θ_ℓ para el esquema de firma Θ y clave pública $y_{A^\star}$. A continuación $\mathcal{F}_\Theta$ ejecuta $\tilde{\Theta}.\mathsf{Firm}(\tilde{\mathsf{sk}}_\ell, C_{1,\ell}||C_{2,\ell}||C_{3,\ell}||y_{A^\star}||y_{\mathcal{B}_\ell}||\theta_\ell) \rightarrow \tilde{\theta}_\ell$. Finalmente, $\mathcal{F}_\Theta$ devuelve la firma $\sigma_\ell = (C_{1,\ell}, C_{2,\ell}, C_{3,\ell}, \tilde{\mathsf{vk}}_\ell, \theta_\ell, \tilde{\theta}_\ell)$ a $\mathcal{A}_{\mathsf{INF}}$.

Falsificación de Θ. En algún momento $\mathcal{A}_{\mathsf{INF}}$ produce con probabilidad $\varepsilon = \mathsf{Vent}_{\mathcal{A}_{\mathsf{INF}}}(\lambda)$ una clave $y_{\mathcal{B}^\star}$ y

una firma falsificada $(m^\star, \sigma^\star)$ para un conjunto de verificadores $(\mathcal{B}^\star, \Gamma_{\mathcal{B}^\star})$, donde $\sigma^\star = (C_1^\star, C_2^\star, C_3^\star, \tilde{\mathsf{vk}}^\star, \theta^\star, \tilde{\theta}^\star)$ verifica las siguientes dos propiedades. Primero, el par $(m^\star, \sigma^\star)$ debe ser diferente a los obtenidos anteriormente durante las peticiones de firma y segundo, se debe verificar $\Sigma.\mathsf{Ver}(\mathtt{params}, m^\star, \sigma^\star, y_{A^\star}, B, \{\alpha_{\mathcal{B}^\star,j}\}_{j\in B}) = 1$, para algún subconjunto $B \in \Gamma_{\mathcal{B}^\star}$.

Al definir $M^\star = C_1^\star||C_2^\star||C_3^\star||y_{A^\star}||y_{B^\star}||\tilde{\mathsf{vk}}^\star$, podemos distinguir entre dos casos. Primero, con probabilidad ε_1 tenemos que $(M^\star, \theta^\star) \neq (M_\ell, \theta_\ell)$, para todos los mensajes M_ℓ que $\mathcal{F}_\Theta$ ha solicitado a su oráculo de firma. En este caso, $\mathcal{F}_\Theta$ ha obtenido una firma válida $(M^\star, \theta^\star)$ para el esquema Θ y clave pública $y_{A^\star}$. Por tanto, $\varepsilon_1 \leq \mathsf{Vent}_{\mathcal{F}_\Theta}(\lambda)$.

FALSIFICACIÓN DE $\tilde{\Theta}$. Por otro lado, con probabilidad $\varepsilon_2 = \varepsilon - \varepsilon_1$, tenemos que $(M^\star, \theta^\star) = (M_\ell, \theta_\ell)$ para alguno de los Q mensajes M_ℓ que $\mathcal{F}_\Theta$ ha solicitado a su oráculo aleatorio. En este caso, como la falsificación de $\mathcal{A}_{\mathsf{INF}}$ es válida, la única posibilidad es que $\tilde{\theta}^\star \neq \tilde{\theta}_\ell$. De esta manera se puede construir un algoritmo $\mathcal{F}_{\tilde{\Theta}}$ contra la infalsificabilidad del esquema $\tilde{\Theta}$ de firma de un solo uso.

Este algoritmo recibe como entrada una clave de verificación $\tilde{\mathsf{vk}}'$ obtenida al ejecutar $\tilde{\Theta}.\mathsf{GC}(1^\lambda) \rightarrow (\tilde{\mathsf{sk}}', \tilde{\mathsf{vk}}')$, y tiene acceso a un oráculo de firma $\tilde{\Theta}.\mathsf{Firm}(\tilde{\mathsf{sk}}', \cdot)$ para un único mensaje de su elección. $\mathcal{F}_{\tilde{\Theta}}$ escoge una petición de firma válida $\ell \in \{1, \ldots, Q\}$ aleatoriamente, hace una petición a su propio oráculo de firma para obtener la firma correspondiente $\tilde{\theta}_\ell$ de su elección, y utiliza pares de claves $(\tilde{\mathsf{sk}}, \tilde{\mathsf{vk}})$ diferentes, generados por él mismo, para responder al resto de solicitudes de firma de $\mathcal{A}_{\mathsf{INF}}$. Si $\mathcal{F}_{\tilde{\Theta}}$ ha adivinado el índice ℓ correcto (lo que ocurre con probabilidad $1/Q$) entonces este segundo tipo de falsificación de $\mathcal{A}_{\mathsf{INF}}$ en el que $(M^\star, \theta^\star) = (m_\ell, \theta_\ell)$ implica directamente una falsificación válida de $\mathcal{F}_{\tilde{\Theta}}$ contra el esquema de firma $\tilde{\Theta}$. Por tanto, obtenemos $\mathsf{Vent}_{\mathcal{F}_{\tilde{\Theta}}}(\lambda) \geq \varepsilon_2/Q$.

La probabilidad de éxito es pues $\mathsf{Vent}_{\mathcal{A}_{\mathsf{INF}}}(\lambda) = \varepsilon = \varepsilon_1 + \varepsilon_2 \leq \mathsf{Vent}_{\mathcal{F}_\Theta}(\lambda) + Q \cdot \mathsf{Vent}_{\mathcal{F}_{\tilde{\Theta}}}(\lambda)$. Si asumimos que ambos esquemas Θ y $\tilde{\Theta}$ son infalsificables, entonces las respectivas ventajas $\mathsf{Vent}_{\mathcal{F}_\Theta}(\lambda)$ y $\mathsf{Vent}_{\mathcal{F}_{\tilde{\Theta}}}(\lambda)$ son negligibles. Consecuentemente, también lo es $\mathsf{Vent}_{\mathcal{A}_{\mathsf{INF}}}(\lambda)$, y podemos concluir que el nuevo esquema de firma con verificación distribuida Σ es infalsificable. □

B. Privacidad

En el siguiente teorema demostraremos que nuestro esquema de firma con verificación distribuida satisface la propiedad de privacidad, por reducción al problema decisional BDDH en grupos $\mathbb{G}, \mathbb{G}_T$ y a la seguridad del esquema de firma de un solo uso $\tilde{\Theta}$. La demostración es en el modelo estándar.

Teorema 2. Sea $\lambda \in \mathbb{N}$ un parámetro de seguridad. Para cualquier atacante $\mathcal{A}_{\mathsf{IND}}$ contra la privacidad de nuestro esquema de firma con verificación distribuida, existe un solucionador $\mathcal{A}^{BDDH}$ del problema Bilineal Decisional de Diffie-Hellman o un algoritmo $F_{\tilde{\Theta}}$ contra $\tilde{\Theta}$, tales que

$$\mathsf{Vent}_{\mathcal{A}^{BDDH}}(\lambda) \; + \; \mathsf{Vent}_{\mathcal{F}_{\tilde{\Theta}}}(\lambda) \; \geq \; \mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda).$$

Demostración. Asumiendo que tenemos un atacante $\mathcal{A}_{\mathsf{IND}}$ que tiene ventaja $\mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda)$ en romper la privacidad de nuestro esquema de firma, vamos a construir un algoritmo $\mathcal{A}^{BDDH}$ que irá usando a su vez a $\mathcal{A}_{\mathsf{IND}}$ como subrutina, para resolver el problema BDDH en los grupos $\mathbb{G}, \mathbb{G}_T$.

El algoritmo $\mathcal{A}^{BDDH}$ recibe como entrada un grupo cíclico $\mathbb{G} = \langle g \rangle$ de orden primo q junto con una tupla (g^a, g^b, g^c, R), donde R es $e(g,g)^{abc}$ o un elemento aleatorio de $\mathbb{G}_T$. El objetivo de $\mathcal{A}^{BDDH}$ es distinguir entre estos dos casos.

INICIALIZACIÓN DE $\mathcal{A}_{\mathsf{IND}}$. El adversario $\mathcal{A}_{\mathsf{IND}}$ escoge un conjunto de verificadores $\mathcal{B}^\star = \{1, \ldots, n\}$, una estructura de acceso $\Gamma_{\mathcal{B}^\star} \subset 2^{\mathcal{B}^\star}$ y un subconjunto de participantes corruptos $\widetilde{B} \notin \Gamma_{\mathcal{B}^\star}$.

$\mathcal{A}^{BDDH}$ simula una ejecución del protocolo $\Sigma.\mathsf{Ini}(1^\lambda)$: $\mathcal{A}^{BDDH}$ ejecuta el protocolo de generación de claves del esquema de firma $\tilde{\Theta}$, obteniendo $\tilde{\Theta}.\mathsf{GC}(1^\lambda) \rightarrow (\tilde{\mathsf{sk}}^\star, \tilde{\mathsf{vk}}^\star)$. Entonces $\mathcal{A}^{BDDH}$ elige dos funciones hash $H_0 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ y $H_1 : \{0,1\}^* \times \mathbb{G}_T \rightarrow \{0,1\}^*$ arbitrarias y un esquema de firma Θ. Se calcula el valor $\mathsf{id}^\star = H_0(\tilde{\mathsf{vk}}^\star)$. $\mathcal{A}^{BDDH}$ define $g_2 = g^a$, elige un valor aleatorio $\gamma \in \mathbb{Z}_p^*$ y define $h = (g^b)^{-\mathsf{id}^\star} \cdot g^\gamma$. $\mathcal{A}^{BDDH}$ inicializa $\mathcal{A}_{\mathsf{IND}}$ dándole los valores $\mathtt{params} = (q, \mathbb{G}, g, h, g_2, \mathbb{G}_T, e, \Theta, \tilde{\Theta}, H_0, H_1)$.

La ejecución del protocolo $\Sigma.\mathsf{GC}(\mathtt{params}, \mathcal{B}^\star, \Gamma_{\mathcal{B}^\star}, \text{`colectivo'})$ es simulada por $\mathcal{A}^{BDDH}$ de la siguiente manera. Define la clave pública de $\mathcal{B}^\star$ como $y_{\mathcal{B}^\star} = g^b$, que es enviada a $\mathcal{A}_{\mathsf{IND}}$. Esto significa que la clave secreta $\alpha_{\mathcal{B}^\star}$ asociada es implícitamente b. Los fragmentos de los participantes corruptos, $\{x_{\mathcal{B}^\star,j}\}_{j\in\widetilde{B}}$ son calculados primero eligiendo aleatoria e independientemente valores $\alpha_{\mathcal{B}^\star,j} \in \mathbb{Z}_q$ y luego calculando $x_{\mathcal{B}^\star,j} = g_2^{\alpha_{\mathcal{B}^\star,j}}$. Estos fragmentos son también dados a $\mathcal{A}_{\mathsf{IND}}$. Denotaremos en adelante $\psi : \mathcal{B}^\star \rightarrow (\mathbb{Z}_q)^{|\widetilde{B}|+1}$ a un esquema de compartición de secretos que satisface $\psi(0) = b = \alpha_{B^\star}$ y $\psi(j) = \alpha_{B^\star,j}$, para $j \in \widetilde{B}$.

Usando interpolación en el exponente y los valores $y_{\mathcal{B}^\star} = g^{\alpha_{\mathcal{B}^\star}} = g^b$ y $\{\alpha_{\mathcal{B}^\star,j}\}_{j\in\widetilde{B}}$, se pueden obtener todos los valores $D_{\mathcal{B}^\star,j} = g^{\alpha_{\mathcal{B}^\star,j}}$ (si se requiere robustez), para todos los miembros $j \in \mathcal{B}^\star$, corruptos o no.

GENERACIÓN DE NUEVAS CLAVES. El atacante $\mathcal{A}_{\mathsf{IND}}$ puede generar libremente nuevas claves públicas y secretas para firmantes A y para colectivos $(\mathcal{B}, \Gamma_{\mathcal{B}}) \neq (\mathcal{B}^\star, \Gamma_{\mathcal{B}^\star})$ de verificadores de su elección.

PETICIONES VERIFICACIÓN. $\mathcal{A}_{\mathsf{IND}}$ solicita una petición de verificación $(m_\ell, \sigma_\ell, y_{A_\ell})$ para firmantes A_ℓ y conjunto de verificadores $\mathcal{B}^\star$, con $\sigma_\ell = (C_{1,\ell}, C_{2,\ell}, C_{3,\ell}, \tilde{\mathsf{vk}}_\ell, \theta_\ell, \tilde{\theta}_\ell)$. Si $\tilde{\mathsf{vk}}_\ell = \tilde{\mathsf{vk}}^\star$ y $\tilde{\Theta}.\mathsf{Ver}(\tilde{\mathsf{vk}},\ C_{1,\ell}||C_{2,\ell}||C_{3,\ell}||y_A||y_{\mathcal{B}}||\theta_\ell\ , \tilde{\theta}_\ell) \rightarrow 1$, entonces $\mathcal{A}^{BDDH}$ aborta y devuelve un bit aleatorio. En caso contrario, $\mathcal{A}^{BDDH}$ ejecuta los pasos a)-c), que son verificaciones públicas, del protocolo de verificación.

Si $(m_\ell, \sigma_\ell, y_{A_\ell})$ es una firma válida y $\mathcal{A}^{BDDH}$ no ha abortado, tenemos que $\tilde{\mathsf{vk}}_\ell \neq \tilde{\mathsf{vk}}^\star$ y $\mathsf{id}_\ell = H_0(\tilde{\mathsf{vk}}_\ell)$. Como se supone que la función hash es resistente a colisiones, obtenemos $\mathsf{id}_\ell \neq \mathsf{id}^\star$. Ahora $\mathcal{A}^{BDDH}$ debe simular los valores que son emitidos en la ejecución del resto del protocolo. Para ello construye tuplas consistentes $(j, \omega_{0\ell,j}, \omega_{1\ell,j})$ para cualquier usuario $j \in \mathcal{B}^\star$, donde $\omega_{0\ell,j} = \alpha_{B^\star,j} \cdot (y_{\mathcal{B}^\star}^{\mathsf{id}_\ell} \cdot h)^{r_{\ell,j}}$ y $\omega_{1\ell,j} = g^{r_{\ell,j}}$ para un valor uniformemente aleatorio $r_{\ell,j} \in \mathbb{Z}_q$.

Para los participantes corruptos $j \in \widetilde{B}$, $\mathcal{A}^{BDDH}$ puede calcular estos valores fácilmente ya que conoce $\{\alpha_{B^\star,j}\}_{j\in\widetilde{B}}$.

Para los usuarios no corruptos $i \in \mathcal{B}^\star \backslash \widetilde{B}$, $\mathcal{A}^{BDDH}$ utiliza los coeficientes públicos dados por $\psi(i) = \lambda_0\psi(0) + \sum_{j\in\widetilde{B}} \lambda_j\psi(j)$. De esta manera, $\mathcal{A}^{BDDH}$ puede usar en el exponente las relaciones lineales entre fragmentos y secretos, determinadas por el esquema para compartir secretos que realice $\Gamma_{\mathcal{B}^\star}$. Ahora, el algoritmo $\mathcal{A}^{BDDH}$ elige $\tilde{r}_{\ell,i} \in \mathbb{Z}_q$ aleatoriamente y define $\omega_{0\ell,i} = g_2^{\frac{-\gamma\lambda_0}{\mathsf{id}_\ell - \mathsf{id}^\star}} \cdot (y_{\mathcal{B}^\star}^{\mathsf{id}_\ell} \cdot h)^{\tilde{r}_{\ell,i}} \cdot \prod_{j\in\widetilde{B}} \alpha_{B^\star,j}^{\lambda_j}$ y $\omega_{1\ell,i} = g_2^{\frac{-\lambda_0}{\mathsf{id}_\ell - \mathsf{id}^\star}} \cdot g^{\tilde{r}_{\ell,i}}$. Se puede comprobar fácilmente que estos dos valores $(\omega_{0\ell,i}, \omega_{1\ell,i})$ tienen la forma $\omega_{0\ell,i} = g_2^{\psi(i)} \cdot (y_{\mathcal{B}^\star}^{\mathsf{id}_\ell} \cdot h)^{r_{\ell,i}} = \alpha_{\mathcal{B}^\star,i} \cdot (y_{\mathcal{B}^\star}^{\mathsf{id}_\ell} \cdot h)^{r_{\ell,i}}$ y $\omega_{1\ell,i} = g^{r_{\ell,i}}$, donde $r_{\ell,i} = \tilde{r}_{\ell,i} - \frac{a\lambda_0}{\mathsf{id}_\ell - \mathsf{id}^\star}$ está definido implícitamente como un valor aleatorio distribuido uniformemente en $\mathbb{Z}_q$.

Resumiendo, $\mathcal{A}^{BDDH}$ puede simular tuplas válidas $(j, \omega_{0\ell,j}, \omega_{1\ell,j})$ para cualquier usuario $j \in \mathcal{B}^\star$. Finalmente, el resto del proceso de verificación puede ser completado fácilmente por $\mathcal{A}^{BDDH}$, que verifica $H_1\left(m_\ell, \frac{e(C_{3,\ell},\omega_{1\ell})}{e(C_{1,\ell},\omega_{0\ell})}\right) = C_{2,\ell}$. Según el resultado de esa verificación, $\mathcal{A}^{BDDH}$ devuelve la salida 1 (firma válida) o 0 (firma no válida), junto con todos los valores obtenidos en el proceso a $\mathcal{A}_{\mathsf{IND}}$.

DESAFÍO. En un momento dado, $\mathcal{A}_{\mathsf{IND}}$ escoge y publica dos mensajes m_0, m_1 de la misma longitud, junto con un firmante $A^\star$ con claves $(x_{A^\star}, y_{A^\star})$. Para que $\mathcal{A}_{\mathsf{IND}}$ reciba una firma $\sigma^\star$ como repuesta, el algoritmo $\mathcal{A}^{BDDH}$ elige un bit aleatorio $d \in \{0,1\}$ y procede de la siguiente manera:

1) Define $C_1^\star = g^c$, $C_2^\star = H_1(m_d, R)$ y $C_3^\star = (g^c)^\gamma = (y_{\mathcal{B}^\star}^{\mathsf{id}^\star} \cdot h)^c$.
   Nótese que $(C_1^\star, C_2^\star, C_3^\star)$ es una firma consistente de m_b para la identidad $\mathsf{id}^\star$ cuando $R = e(g,g)^{abc}$. Por otra parte, cuando R es un valor aleatorio en $\mathbb{G}_T$, la distribución de $(C_1^\star, C_2^\star, C_3^\star)$ es independiente del bit d, puesto que las distribuciones $H_1(m_0, R)$ y $H_1(m_1, R)$ son indistinguibles si la función de hash H_1 tiene las propiedades usuales de seguridad.
2) Ejecuta $\Theta.\mathsf{Firm}(x_{A^\star},\ C_1^\star||C_2^\star||C_3^\star||y_{A^\star}||y_{\mathcal{B}^\star}||\tilde{\mathsf{vk}}^\star) \to \theta^\star$.
3) Ejecuta $\tilde{\Theta}.\mathsf{Firm}(\tilde{\mathsf{sk}}^\star,\ C_1^\star||C_2^\star||C_3^\star||y_{A^\star}||y_{\mathcal{B}^\star}||\theta^\star) \to \tilde{\theta}^\star$.
4) Envía la firma final $\sigma^\star = (C_1^\star, C_2^\star, C_3^\star, \tilde{\mathsf{vk}}^\star, \theta^\star, \tilde{\theta}^\star)$ al atacante $\mathcal{A}_{\mathsf{IND}}$.

MÁS PETICIONES. El atacante $\mathcal{A}_{\mathsf{IND}}$ puede hacer más peticiones de verificación $\sigma = (C_1, C_2, C_3, \tilde{\mathsf{vk}}, \theta, \tilde{\theta})$ para el colectivo de verificadores $\mathcal{B}^\star$ siempre que las peticiones no hayan sido hechas con anterioridad; es decir, $\sigma \neq \sigma^\star$ para claves $y_A \neq y_{A^\star}$. Si $\tilde{\mathsf{vk}} \neq \tilde{\mathsf{vk}}^\star$, entonces estas peticiones son respondidas de igual manera a como ha sido descrito anteriormente. En caso contrario, $\tilde{\mathsf{vk}} = \tilde{\mathsf{vk}}^\star$ y $\tilde{\Theta}.\mathsf{Ver}(\tilde{\mathsf{vk}},\ C_1||C_2||C_3||y_A||y_{\mathcal{B}}||\theta\ ,\tilde{\theta}) \to 1$, el algoritmo $\mathcal{A}^{BDDH}$ aborta y devuelve un bit aleatorio.

ANÁLISIS FINAL. $\mathcal{A}_{\mathsf{IND}}$ devuelve un bit d'. Si $d' = d$ entonces $\mathcal{A}^{BDDH}$ devuelve 0 como la solución correcta del problema BDDH. Si $d' \neq d$ entonces $\mathcal{A}^{BDDH}$ devuelve 1.

Denotamos δ a la probabilidad que $\mathcal{A}_{\mathsf{IND}}$ haga una petición de verificación para la firma válida $\sigma = (C_1, C_2, C_3, \tilde{\mathsf{vk}}, \theta, \tilde{\theta})$ tal que $\tilde{\mathsf{vk}} = \tilde{\mathsf{vk}}^\star$. En otras palabras, δ es la probabilidad que $\mathcal{A}^{BDDH}$ aborte antes que $\mathcal{A}_{\mathsf{IND}}$ devuelva el bit d'. Usando un argumento similar a la demostración de infalsificabilidad, construimos un algoritmo $\mathcal{F}_{\tilde{\Theta}}$ contra la infalsificabilidad del esquema de firma de un solo uso $\tilde{\Theta}$: $\mathcal{F}_{\tilde{\Theta}}$ recibe como entrada $\tilde{\mathsf{vk}}^\star$, y el único acceso al oráculo de firma es usado para calcular la firma en el desafío. Cualquier petición de verificación válida que $\mathcal{A}_{\mathsf{IND}}$ solicite para $\tilde{\mathsf{vk}} = \tilde{\mathsf{vk}}^\star$ lleva a una falsificación válida del esquema de firma $\tilde{\Theta}$. Por tanto, $\delta \leq \mathsf{Vent}_{\mathcal{F}_{\tilde{\Theta}}}(\lambda)$.

Finalmente, vamos a calcular la probabilidad que $\mathcal{A}^{BDDH}$ devuelva la solución correcta del problema BDDH, es decir, que devuelva 0 en los dos casos posibles. Cuando $R = e(g,g)^{abc}$, la firma calculada en el desafío es consistente y obtenemos $\Pr[\mathcal{A}^{BDDH}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 0] = \delta \cdot \frac{1}{2} + (1-\delta) \cdot \left(\mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda) + \frac{1}{2}\right)$.

En caso contrario, cuando $R = T$ es un elemento aleatorio de $\mathbb{G}_T$, la firma calculada es independiente de d. Así que la probabilidad que $d' = d$ es $1/2$, y obtenemos $\Pr[\mathcal{A}^{BDDH}(g, g^a, g^b, g^c, T) = 0] = \delta \cdot \frac{1}{2} + (1-\delta) \cdot \frac{1}{2}$.

De esta manera, obtenemos que $\mathsf{Vent}_{\mathcal{A}^{BDDH}}(\lambda) = |\Pr[\mathcal{A}^{BDDH}(g, g^a, g^b, g^c, e(g,g)^{abc}) = 0] - \Pr[\mathcal{A}^{BDDH}(g, g^a, g^b, g^c, T) = 0]| = (1-\delta) \cdot \mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda) = \mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda) - \delta \cdot \mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda)$.

Juntando las relaciones anteriores, obtenemos la desigualdad buscada: $\mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda) = \mathsf{Vent}_{\mathcal{A}^{BDDH}}(\lambda) + \delta \cdot \mathsf{Vent}_{\mathcal{A}_{\mathsf{IND}}}(\lambda) \leq \mathsf{Vent}_{\mathcal{A}^{BDDH}}(\lambda) + \delta \leq \mathsf{Vent}_{\mathcal{A}^{BDDH}}(\lambda) + \mathsf{Vent}_{\mathcal{F}_{\tilde{\Theta}}}(\lambda)$. □

REFERENCES

[1] D. Boneh, X. Boyen, S. Halevi, "Chosen ciphertext secure public key threshold encryption without random oracles". *Proceedings of CT-RSA'06*, LNCS **3860**, Springer-Verlag, pp. 226–243, 2006.

[2] D. Boneh, E. Shen, B. Waters, "Strongly unforgeable signatures based on Computational Diffie-Hellman". *Proceedings of PKC'06*, LNCS **3958**, Springer-Verlag, pp. 229–240, 2006.

[3] E.F. Brickell, "Some ideal secret sharing schemes". *Journal of Combinatorial Mathematics and Combinatorial Computing*, **9**, pp. 105–113, 1989.

[4] R. Canetti, S. Halevi, J. Katz "Chosen-ciphertext security from identity-based encryption". *Proceedings of Eurocrypt'04*, LNCS **3027**, Springer-Verlag, pp. 207–222, 2004.

[5] R. Gennaro, S. Jarecki, H.Krawczyk, T. Rabin, "Secure distributed key generation for Discrete-Log based cryptosystems". *Journal of Cryptology*, vol. **4** (1), Springer-Verlag, pp. 51–83, 2007.

[6] S. Goldwasser, S. Micali, "Probabilistic encryption". *Journal of Computer and System Sciences*, **28**, pp. 270–299, 1984.

[7] S. Goldwasser, S. Micali, R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks". *SIAM J. of Computing* **17** (2), pp. 281–308, 1988.

[8] J. Herranz, A. Ruiz, G. Sáez, "Esquemas de firma digital con verificación distribuida". *Actas de la X Reunión Española de Criptología y Seguridad de la Información, RECSI'08*, pp. 209–216, 2008.

[9] J. Herranz, A. Ruiz, G. Sáez, "Máxima seguridad para firmas digitales con verificación distribuida". *Actas de la XI Reunión Española de Criptología y Seguridad de la Información, RECSI'10*, pp. 97–103, 2010.

[10] F. Laguillaumie, D. Vergnaud, "Multi-designated verifiers signatures". *Proceedings of ICICS'04*, LNCS **3269**, Springer-Verlag, pp. 495–507, 2004.

[11] C.H.Lim, P.J.Lee, "Directed signatures and applications to threshold cryptosystems". *Workshop on Security Protocols*, LNCS **1189**, Springer-Verlag, pp. 131–138, 1997.

[12] P. Mohassel, "One-time signatures and chameleon hash functions". *Proceedings of SAC'10*, LNCS **6544**, Springer-Verlag, pp. 302–319, 2011.

[13] H. Petersen, M. Michels, "On signature schemes with threshold verification detecting malicious verifiers". *Security Protocols Workshop*, LNCS **1361**, Springer-Verlag, pp. 67–78, 1997.