Optimización de un Protocolo Non-Interactive Dining Cryptographers

Jeroen van de Graaf
Universidade Federal de Minas Gerais (UFMG) – Brasil

Germán Montejano
Universidad Nacional de San Luis (UNSL) - Argentina

Pablo García
Universidad Nacional de La Pampa (UNLPam) - Argentina

Abstract

Existen múltiples aplicaciones, como por ejemplo, voto electrónico, que exigen seguridad incondicional para el anonimato, de manera tal que la identidad de un votante no pueda ser descubierta, independientemente del tiempo y los recursos con los que pueda contar un criptoanalista. Dining Chryptographers es un protocolo que asegura tal condición. Una derivación posterior, Non-Interactive Dining Cryptographers (NIDC), relaja la condición de concurrencia online exigida en la versión original. Esto amplía notablemente el conjunto de situaciones a las que puede aplicarse el protocolo, porque son múltiples los problemas prácticos que exigen anonimato incondicional pero que no verifican la asistencia simultánea de todos los participantes.
El presente trabajo propone una optimización en el protocolo de comunicación, manteniendo el nivel requerido de seguridad en lo que respecta a la administración de maniobras fraudulentas.

Palabras Clave:

Seguridad Incondicional, Anonimato, Voto Electrónico, Non Interactive Dining Cryptographers, Fraude, Logaritmos discretos, Bit Commitment, Homomorfismo, Prover, Verifier.

Introducción

Muchos esquemas de voto electrónico actuales proveen seguridad incondicional en lo referido al proceso específico de votación, otorgando simultáneamente un nivel computacional para el mantenimiento del anonimato de los votantes. Tal situación parece incorrecta, dado que el anonimato debe garantizarse indefinidamente, mientras que el proceso eleccionario sólo debe ser protegido durante un período limitado de tiempo, en lo referido a las maniobras tendientes a adulterar los resultados del comicio, dado que, una vez finalizado el mismo, los resultados serán públicos.
En consecuencia, crece el interés por aquellos protocolos que provean niveles de seguridad acorde a lo expresado. Dining Cryptographers (Chaum, [2]) cumple exactamente con esos requerimientos.
Si se trata de una aplicación de voto electrónico, resulta de interés una derivación (van de Graaf, [14]) llamada Non-Interactive Dining Cryptographers (NIDC) que incorpora la valiosa característica de no exigir la concurrencia en el tiempo de todos los participantes.

El esquema original implementa una solución que satisface las exigencias de seguridad planteadas, pero que resulta ineficiente. El modelo se basa en la utilización de BCX (Bit Commitments con XOR) que derivan en una cantidad significativa de operaciones a nivel de bit.
Se propone una variante que usa las propiedades de los logaritmos discretos y los Commitments de Pedersen. La nueva propuesta mantiene la seguridad en el nivel exigido, pero con una importante optimización en términos de las operaciones involucradas.
Dining Cryptographers puede describirse de la siguiente manera:

"Tres criptógrafos comparten una cena en un restaurant. Al llegar el momento de pagar, el mozo les indica que la adición ya ha sido abonada y, que quién lo hizo, no desea que se conozca su identidad. Los criptógrafos desean saber si alguno de los comensales fue quien realizó el pago, o si la misma fue abonada por alguien externo al grupo. Ellos desean saber solamente eso: si pagó alguno de ellos o no. En caso de un pagador externo, el anonimato está garantizado, pero si fuese un integrante del grupo, los demás respetan el derecho a invitar y no desean saber la identidad del pagador. "

Planteado de esta manera, la solución que encuentran es la siguiente:

"Cada uno de los comensales lanza una moneda al aire. Observa el resultado obtenido y lo comparte con su vecino de la izquierda. Luego, cada uno de ellos ve exactamente dos monedas, la propia y la del vecino que comparte con él. Finalmente, cada uno debe indicar si las dos monedas que pudo observar son "iguales" o "diferentes", con la condición de que si alguno de ellos abonó la adición, debe mentir con respecto a su afirmación."

En las condiciones descriptas, si el número de criptógrafos que proclama "diferentes" es impar, el pagador se encuentra en el grupo de comensales. Un número par, en cambio, indica que el pagador es externo al grupo. Se considera que las monedas utilizadas otorgan un resultado auténticamente aleatorio con respecto al experimento "lanzar la moneda al aire", de manera que:

$$P(\text{Cara}) = P(\text{Ceca}) = ½$$

En [17], se demuestra la seguridad del protocolo original de Chaum, contando con que los tres criptógrafos se comportarán de manera honesta. Obviamente, no es realista suponer que todos los participantes mantengan tal conducta, sobre todo cuando los intereses en juego son importantes como en el caso de voto electrónico. Es imprescindible, en consecuencia, implementar herramientas externas para que el protocolo sea capaz de detectar y administrar intentos fraudulentos.
La versión original de Dining Cryptographers presenta la limitación de exigir la concurrencia en el tiempo de todos los participantes. Sin embargo, existen muchas aplicaciones que exigen anonimato incondicional, pero que muestran carácterísticas asincrónicas.
En [14], van de Graaf propone una metodología que otorga seguridad incondicional al anonimato sin exigir la concurrencia temporal de todos los participantes. Para ello, combina el concepto de Firmas Ciegas con una derivación del protocolo de Chaum que denominó Non-Interactive Dining Cryptographers (NIDC).
El concepto de Firma Ciega implica cualquier técnica por la cuál el votante obtenga un voto válido de parte de las autoridades del proceso eleccionario. En particular, podría utilizarse el protocolo de Fujioka, Okamoto y Ohta [7]. Dicho protocolo permite al votante comunicarse con las autoridades para enviar un voto ciego. Éstas responden firmando (de manera ciega) el voto y reenviándoselo al votante. Cabe tener en cuenta que el proceso es perfecto y que todas las opciones

son matematicamente equiprobables, razón por la cuál las autoridades no pueden deducir ninguna información relacionada con las elecciones el votante.
Obviamente, las autoridades deben dejar constancia de cada voto, para que ningún votante pueda reincidir. De la misma manera, ambas partes involucradas deberán fimar sus mensajes y mantener registros de los mismos a los efectos de resolver cualquier diferencia posterior. Un ejemplo del concepto de firmas ciegas se describe en [7], aunque implementado sobre Mix Nets. Sin embargo, se aplica sin inconvenientes a un esquema NIDC.
Conceptualmente, resulta bastante simple describir el modelo NIDC. La diferencia con el modelo original de Chaum radica en que no es necesario que los participantes se encuentren online simultáneamente. Por tratarse de un protocolo sin retroalimentación, aparecen algunas características novedosas. Sin embargo, su comportamiento no difiere demasiado del protocolo original. Se puede describir en tres pasos:

1. En una fase preliminar, cada par de participantes intercambia bits aleatorios.
2. Basándose en los bits aleatorios y la entrada de las partes, se publica un mensaje.
3. Todos los mensajes se combinan, de tal manera que se cancelan todos los bits aleatorios y lo único que permanece son las entradas de todos los participantes.

Es bueno recordar a esta altura, que el anonimato es garantizado directamente por Dining Cryptographers; los detalles de ese punto son explicados en detalle en [2]. En consecuencia, si se prepara un protocolo que permita distinguir mensajes, los mismos serán interpretados evitando la posibilidad de conocer la autoría de los mismos.
En el caso de un modelo asíncrono como el que se describe, el nivel de redundancia es significativo. Esto se debe a que hace falta realizar múltiples verificaciones tendientes a evitar que un participante deshonesto arruine voluntariamente el proceso. Del mismo modo, la protección de la información circulante sólo debe soportar el lapso de tiempo que corresponda al proceso. Por ejemplo, si se trata de un proceso de E-Voting, todas las firmas se publicarán una vez cerrada la elección, haciendo pública esa información en pos de aumentar la transparencia del procedimiento.
En [17] se ha descripto detalladamente el modelo basado en BCX que aparece en [14] para administrar maniobras fraudulentas en un esquema NIDC. Allí se demuestra que la seguridad puede llevarse hasta el nivel que se desee. En efecto, por tratarse de Bit Commitments, cada fraude individual tendrá una probabilidad igual a ½ de ser descubierto. Por lo tanto, para llevar la seguridad hasta el nivel deseado, se deberá implementar la cantidad apropiada de pares. El comportamiento, desde el punto de vista probabilístico corresponde a los sucesos independientes, es decir que la probabilidad de que un fraude no sea detectado disminuye al aumentar los pares. Concretamente, la probabilidad de cometer fraude y que el mismo no sea detectado es $(½)^d$, donde $d \in Z^+$ es la cantidad de pares que se implementen.
El modelo, descripto detalladamente en [17], proporciona resultados satisfactorios en lo referido a seguridad, dado que puede llevarse hasta el nivel que se desee. Sin embargo, resulta ineficiente por la cantidad de operaciones que deben implementarse a nivel de bit.

Optimización del esquema NIDC

Se propone entonces un nuevo modelo que reemplace la utilización de BCX por otros recursos que permitan mantener el nivel de seguridad pero ofreciendo una eficiencia mayor. Concretamente, se utilizan dos elementos:

- Logaritmos discretos: Los logaritmos discretos resultan de sumo interés en temas criptográficos porque cumplen con una premisa fundamental: su aplicación garantiza un encriptado sencillo y un desencriptado muy simple si se cuenta con la clave; al mismo tiempo es muy difícil de descifrar si no se cuenta con la clave. A la fecha no existen algoritmos conocidos que puedan calcular en tiempo aceptable este tipo de algoritmos. Esto constituye su fortaleza frente a cualquier ataque de un criptoanalista.

- Commitments de Pedersen: En criptografía, un esquema basado en commitments permite a alguien comprometer un valor sin develarlo, para que sea dado a conocer en el momento apropiado. Lo importante del enfoque es que garantiza una doble seguridad: el emisor no devela el valor hasta el momento indicado, pero el receptor cuenta con la garantía de que quién publicó la información no podrá modificarlo posteriormente. En consecuencia, los commitments son apropiados para situaciones en las que dos partes necesitan garantías para una situación de ese estilo. El esquema es ampliamente utilizado en protocolos criptográficos, como por ejemplo, Zero Knowledge Proofs (ZKP), Coin Flipping y Secure Computation. En particular, los commitments de Pedersen presentan otra característica muy favorable: son homomórficos. Esto les otorga un interés especial en temas criptográficos.

Ambos conceptos se describen con detalle en [13]. En [15] se demuestran algunas propiedades de los Commitments de Pedersen, que resultan de interés para el presente análisis:

- Medida constante de los commitments, independientemente del valor a comprometer.
- Seguridad incondicional respecto del anonimato.
- Nivel de seguridad deseado con respecto al commitment.
- Garantía de que quien realiza commitments puede demostrar a cualquier participante de que dos instancias corresponden a un mismo valor con pruebas sólidas, que además podrán repetirse un número arbitrario de veces.
- Posibilidad de abrir un subconjunto arbitrario de commitments, sin revelar ninguna información relacionada con aquellos que no han sido abiertos.

El nuevo modelo se implementa en los siguientes pasos:

1. Cada par de participantes (p_i y p_j) intercambia bits aleatorios, formando r_{ij}. Este valor se compromete ante los demás participantes. Este paso se corresponde exactamente con la técnica del esquema Dining Cryptographers de Chaum ([2]), que en su versión original consiste en compartir los demás participantes el valor (cara o ceca) que se obtuvo al lanzar una moneda al aire. El enfoque se generaliza con facilidad desde tres criptógrafos que comparten información de un bit a cualquier cantidad de participantes intercambiando datos de cualquier tamaño, aunque todos de la misma dimensión.
2. Cada participante p_i , elige un slot al azar. La aleatoriedad de la elección es condición imprescindible para asegurar el anonimato.
3. Cada p_i construye m_i, conteniendo un mensaje, en el slot elegido y 0 en los demás. Este esquema es explicado en detalle en [14]. La idea es

que al terminar el proceso, los slots elegidos por cada usuario contendrán los respectivos datos y los que no resultaran seleccionados, contendrán solamente ceros. Como fue mencionado anteriormente, la aleatoriedad hace que exista la posibilidad de colisiones, las cuales derivan en la pérdida de todos los datos que coinciden en un slot determinado. En [16] y [17] se proponen técnicas para mantener la probabilidad de colisión tan baja como se desee, las cuales son totalmente compatibles con esta propuesta.

4. Se divide m_i en bloques x_i . Cada bloque será un slot del mensaje construido en el paso 3. Es aconsejable que la medida de los slots sea razonablemente mayor que el mensaje que se desea enviar, para disponer de una cantidad de almacenamiento para utilizar como redundancia orientada a elementos de que garanticen la autenticidad. También es aconsejable que dicho tamaño sea potencia de dos, para aprovechar de manera óptima el espacio.
5. Se publica un conjunto de generadores $g_0 \ldots g_l$. El valor l representa la cantidad de slots que se implementan.
6. Cada pi construye:

$$h = (g_0)^k (g_1)^{x1} \ldots (g_l)^{xl}$$

Este punto representa un commitment del mensaje creado, en el cuál k es la clave. Tal commitment debe verificar dos condiciones:

Es imprescindible demostrar que h es un commitment que corresponde a un mensaje bien formado de acuerdo a las reglas antes mencionadas, es decir que contiene ceros en todos los slots, excepto en uno, donde irá su voto.

Esto, obviamente, debe realizarse sin denunciar la ubicación, es decir el slot en donde el usuario va a escribir el dato que desea publicar, ni el contenido del mismo.

Debe demostrarse que existe relación lineal entre lo comprometido y la contribución final que se va a realizar. Es imprescindible que, luego de demostrar que h corresponde a un mensaje bien formado, se demuestre también de manera fehaciente que la contribución final se corresponde con lo comprometido.

La demostración de la primera de las dos condiciones mencionadas consiste en la aplicación de un subprotocolo específico, el cuál es una generalización del que se propone en la sección 3 de [14].

Concretamente, los pasos a seguir para resolver este punto en particular, son los siguientes:

- Realizar una permutación h' del commitment h, en la que se altere el orden de los slots.
- Comprometer dicha permutación. Esto significa que se comprometerá de manera indiscutible en qué orden de la nueva permutación se encuentra cada uno de los bloques de h.
- Se permitirán dos tipo de retos, que el desafiante elegirá de manera aleatoria. Si elige el primer tipo, *prover* abrirá todos los commitments, menos aquellos que incluyen el mensaje. Obviamente, el resultado de los mismos debe ser 0 en todos los casos. Pero si *verifier* elige el segundo tipo de reto, lo que se abre es la permutación completa.

Es fácil observar la equivalencia entre lo expuesto y el esquema presentado en [14]. En cualquier caso, queda claro que lo comprometido corresponde a un mensaje bien formado.

Con respecto a la segunda condición, la misma se verifica durante la aplicación de los siguientes pasos del proceso.

7. Cada p_i computa su contribución final:

$$c_i = m_i + \sum r_{ij}$$

8. Se divide la contribución en bloques w_i, equivalentes a los x_i, en el sentido de que cada uno tiene el tamaño de un slot. La diferencia es que estos nuevos bloques w_i corresponden a un slot de la contribucion definitiva c_i.
9. Se realiza un nuevo commitment a, que hace referencia a la contribución final:

$$a = g_0^s g_1^{w1} \ldots g^{wl}$$

Cada participante podrá realizar la comprobación de que este nuevo commitment se corresponde fielmente con lo comprometido en h. Para ello podrá desafiar con tantos valores v_i como se desee. En todos los ca-sos, quien publica responderá con:

$$b = g_0^{vi\,r+s}\, g_0^{vi\,x1+w1} \ldots\ g_0^{vi\,xl+wl}$$

Verifier podrá controlar que:

$$b = h^c\, a$$

Si la probabilidad de fraude en el i-ésimo caso es p_i, la probabilidad de engañar n desafíos será p_i^n,*en* base a propiedades de los sucesos independientes, las cuales se desarrollan en profundidad en [34].
Entonces, la probabilidad de fraude se puede hacer tan pequeña como se desee. Cabe recordar que la probabilidad p_i va a ser mucho más pequeña que el ½ propuesto en el modelo original de [14], el cuál es un valor que se origina en la propia operación de XOR. Por lo tanto, para alcanzar niveles satisfactorios de seguridad, el número de desafios necesario será relativamente bajo.
Más concretamente, la probabilidad del evento X=”falla un commitment” es:

$$P(X) = 1/q$$

Donde q es el mayor valor que pueda tomar el commitment en cuestión (orden del grupo utilizado) y, en consecuencia, un número primo grande. Por lo tanto, para la mayoría de las aplicaciones habituales, la utilización de un único reto dará niveles suficientes de seguridad. Sin embargo, si en alguna ocasión fuera necesario un nivel de seguridad extrema, el producto de dos o más valores tan cercanos a cero es capaz de proporcionar cualquier nivel de seguridad que pueda exigirse.

10. El resultado de la elección se calcula como el producto de todas las contribuciones.
11. Al resultado final se le aplica XOR de todos los r_{ij}.

Resultados

La seguridad del esquema puede probarse desde un análisis de las *vistas (views)* que cada participante dispone.
Definimos como *view* a una variable aleatoria que describe la información con que cada participante cuenta al finalizar el proceso, es decir cuando el recuento final se da a conocer.
Luego, si podemos mostrar que todos los participantes dispondrán de información que resulta insuficiente para reconstruir el resto de los datos, aún cuando existan conspiraciones, el esquema debe considerarse seguro.
Se busca entonces que dado el resultado final, que finalmente será publicado, ningún participante cuente con elementos que le permitan inferir la identidad del emisor de un dato determinado. Para que los ejemplos resulten más claros vamos a ejemplificar esto aplicándolo a voto electrónico. La generalización a cualquier otra aplicación es inmediata.
El protocolo deberá, entonces, cumplir con una serie de condiciones:

1. Aparece la necesaria restricción de que cada participante sólo puede emitir un voto.
2. Si se aplica un protocolo que garantice anonimato, las entradas de todos los participantes se conocen,

pero debe protegerse la identidad del emisor.

3. Las conspiraciones son posibles. Significa que 2 o más participantes acuerden compartir su información a los efectos de deducir el resto de los votos. Obviamente, el análisis de tales conspiraciones tiene como elemento fundamental el número de conspiradores:

- Si la totalidad de los participantes conspira, el análisis no tiene sentido.
- Si el grupo de conspiradores incluye a todos menos uno, la deducción es trivial.
- Cuando hay dos participantes honestos, si ambos hicieron la misma elección, no hay forma de protegerlos. Si, en cambio, sus opciones fueron diferentes, no debe poder deducirse la elección de cada uno de ellos.
- En el caso de que existan tres o más participantes honestos, se generaliza la idea anterior: la existencia de un voto diferente de los demás debe garantizar el anonimato de todos ellos.

Se deben distinguir dos situaciones diferentes:

- El proceso se lleva a cabo con la participación de un grupo reducido de autoridades. Es el modo más aplicable a voto electrónico. En tal caso, cada votante sólo debe intercambiar mensajes con dichas autoridades.
- No se implementa la figura de las autoridades. Es una generalización del caso anterior. En este caso, cada participante cumple la función de autoridad de todos los votos, menos del propio.

Se pone énfasis en el modelo que incluya autoridades, dado que es el modelo más habitual para voto electrónico. Sin embargo el esquema es fácilmente generalizable a un modelo que no las implemente.
Definimos:

- P_i: Participante i-ésimo. Para el caso de un esquema de voto electrónico, es un votante.
- A_i: Autoridad i-ésima. Para el caso de un esquema de voto electrónico, es aquella persona designada con funciones de control.

La tabla 1 muestra, de manera esquemática, el intercambio de información entre un participante y una autoridad. Obviamente, se generaliza con facilidad, resaltando que, en este caso, cada P_i tiene comunicación con todas las autoridades A_i, pero no con los demás participantes.
La seguridad del esquema puede confirmarse observando las vistas de los involucrados en el proceso:
Cada votante tendrá a la vista los siguientes datos de las autoridades:

1. Grupo de bits aleatorios r_j.
2. Reto x.
3. Retos v_i. Podrían ser n, con $n \in Z^+$. Cuanto mayor sea el valor de n, mayor es el nivel de seguridad obtenido.
4. Respuesta b.

Los datos que observa un votante son exclusivamente desafíos más el intercambio inicial de bits aleatorios necesario para comenzar el proceso de intercambio.
Cada autoridad tendrá una vista consistente en los siguientes datos provenientes del votante:

1. Grupo de bits aleatorio r_{ij}.
2. Commitment h.
3. Respuesta y al desafío x.
4. Commitment a.
5. Respuesta b a los desafíos v_i.

Los datos que expone un participante a una autoridad son commitments y respuestas a desafíos más el necesario intercambio

Tabla 1: Descripción del Protocolo de Comunicación

Votante		**Autoridad**
Bits aleatorios r_i	→	
	←	Bits aleatorios r_j
Construye m_i con el mensaje en la posición elegida y cero en el resto. Lo divide en bloques del tamaño de un slot.		
Construye $h = (g_0)^k (g_1)^{x1} ... (g_l)^{xl}$		
Construye h', permutación (por bloques) de h		
h'	→	
		Elige aleatoriamente $x \in \{0, 1\}$
	←	x
Genera y: Si (x=0) ⇒ y = Abrir compromisos Si (x=1) ⇒ y = Exhibir permutación		
y	→	
		Verifición correcta ⇒ o = OK Verificación incorrecta ⇒ o = CANCEL
	←	o
Calcula la contribución definitiva $c_i = mi + \sum rij$		
Calcula $a = g_0^s g_1^{w1} ... g^{wl}$ (compromiso de la contribución definitiva)		
a	→	
		Elige aleatoriamente $v_i \in Z^+$
	←	v_i
Calcula $b = g_0^{vi\, r + s} g_0^{vi\, x1 + w1} ... g_0^{vi\, xl + wl}$		
b	→	
		$(b = h^c\, a) \Rightarrow y$ = OK $(b \neq h^c\, a) \Rightarrow y$ = CANCEL
	←	y

inicial de bits aleatorios. La seguridad relacionada con tal información puede llevarse al nivel que se desee. La publicación final de los resultados no agrega información que pueda comprometer el anonimato. En consecuencia, queda demostrado que el proceso es confiable.
Cabe mencionar que el protocolo se basa en lo expuesto en la sección 2.4.3 de [1], cuya seguridad es demostrada en la sección 2.4.8 del mismo documento.

Análisis de la optimización

El modelo presentado en el presente trabajo busca reemplazar el esquema original de NIDC, basado en BCX, por otro que mantenga niveles incondicionales de seguridad, con una mayor eficiencia.
El esquema original implementa una cantidad $c \in Z^+$ de *commitments* por cada bit del mensaje; cada uno de ellos, al comprometer un valor binario, presenta una probabilidad ½ de fallar. Por lo tanto, llevar la seguridad a un nivel razonable en tal esquema exige un número relativamente alto de commitments.
Concretamente, si se exigiera un nivel de seguridad mayor que 10^{-6}, la cantidad de commitments necesarios (por cada bit) sería de 20. Y, por supuesto el número de bits en este esquema resulta relativamente muy elevado por las siguientes razones:

- El tamaño de los slots: como ya fue dicho, es aconsejable que esta medida sea bastante mayor que el mensaje claro que se desea enviar, a los efectos de permitir la utilización de redundancia tendiente a garantizar la autenticidad de la información contenida.
- La cantidad de slots: para minimizar las colisiones, la cantidad de slots que se van a utilizar debe ser significativamente mayor que el número de participantes. Más allá de las técnicas presentadas en [16] y [17], que claramente optimizan la utilización del espacio destinado a tal efecto, el número final de slots a implementar es significativamente mayor que el número de participantes.

Para ejemplificar, definimos los siguientes parámetros:

- s: Cantidad de slots implementados.
- t: Tamaño, en bits, de cada slot unitario.
- c: Número de commitments que se implementarán para cada bit del mensaje a publicar.

Luego, si llamamos x al número total de compromisos implementados, obtenemos:

$$x = s\ t\ c$$

Para un ejemplo que utilice:

- $s = 9600$ slots.
- $t = 256$ bits.
- $c = 20$ commitments por bit.

La cantidad de commitments implementados sería de 49.152.000, para cada mensaje que se desee publicar.
En el nuevo esquema, en cambio, el número de commitments disminuye significativamente. De hecho, la implementación de una única ocurrencia de cada uno de los desafíos implementados otorga niveles de seguridad altos. Y cada desafío que se agregue aumenta significativamente tales niveles, pudiéndose alcanzar cualquier nivel que sea exigido, con un número pequeño de commitments.

Conclusión

Reemplazar el esquema original de NIDC basado en la utilización de BCX por otro cuyo fundamento sean las propiedades de los logaritmos discretos permite obtener una eficiencia mayor en cuanto a las operaciones implementadas. La motivación

principal de tal mejora tiene que ver con la semántica de cada commitment individual: En el esquema original, cada BCX presenta una probabilidad ½ de ser engañado. Para obtener un nivel deseado de seguridad *ns*, deben implementarse *d* BCX para cada bit de información, de manera que se verifique:

$$ns < ½^d$$

En la nueva propuesta, cada commitment tiene una probabilidad de $1/n$ de fraude, con $n \in Z^+$ un primo grande. En consecuencia, cada commitment tiene una probabilidad muy pequeña de ser descubierto. En este caso, la incorporación de un nuevo commitment aumenta significativamente el nivel de seguridad, porque se trata de sucesos independientes cuyas probabilidades son muy bajas.
Por otro lado, es importante observar que los commitments no se aplican a nivel de bit, como en [14], sino sobre la totalidad del mensaje. Resulta evidente, en consecuencia, la optimización obtenida, en términos de las operaciones que deben realizarse.

Agradecimientos

El agradecimiento permanente de los autores para los docentes de la Universidade Federal de Minas Gerais que con suma generosidad dieron un apoyo logístico fundamental durante la estadía de Pablo García en Belo Horizonte, en el marco del convenio UNSL – UFMG para el desarrollo de una tesis cotutelada:

- Dr. José Monteiro da Mata
- Dr. José Marcos Silva Nogueira
- Dr. Carlos Camarao
- Dr. Luiz Chaimowicz
- Dr. Wagner Meira Jr
- Dra. Giselle Lobo Papa
- Dra. Jussara Almeida

Referencias

1. Brands S.: Rethinking Public Key Infrastructures and Digital Certificates". MIT Press, 2000.
2. Chaum D.: ”The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability”. Journal of Criptology. 1988.
3. Chaum D., Damgård I. van de Graaf J.: ”Multiparty computation ensuring privacy of each party’s input and correctness of the result”. Advances in Cryptology: Proc. Crypto ’87 (Santa Barbara, CA, August1987), pp. 87-119.
4. Feige U., Fiat A., Shamir A.: ”Zero-Knowledge Proofs of Identity”. Journal of Cryptology. 1988.
5. Feller W.: ”An Introduction to Probability Theory and its Applications”. Volúmen I. Tercera Edición. John Wiley and Sons. New York, 1957.
6. Flajolet P., Gardy D., Thimonier L.: ’Birthday paradox, coupon collectors, caching algorithms and self-organizing search’. Discrete Applied Mathematics 39, ps. 207-223. North-Holland. 1992.
7. Fujioka A., Okamoto T., Ohta K.: ”A Practical Secret Voting Scheme for Large Scale Elections”. AUSCRYPT 1992. LNCS, Vol. 718. Páginas 244 a 251. Springer Heidelberg. 1993.
8. Golle P., Juels A.: "Dining Cryptographers Revisited". In J. Cachin and J. Camenisch, eds., Eurocrypt ’04, pp. 456-473. Springer-Verlag, 2004. LNCS no. 3027.
9. Kizza J.: ”Feige-Fiat-Shamir ZKP Scheme Revisited”. Journal of Computing and ICT Research, Vol. 4, No. 1,pp. 9-19. http://www.ijcir.org/volume4number1/article2.pdf.
10. Lucena López M.: ”Criptografía y Seguridad en Ordenadores”. Tercera Edición. Kriptópolis. 2004.
11. Mao W.: ”Modern Cryptography: Theory and Practice”. Prentice Hall – ISBN: 978-0132887410. 2003.
12. Menezes A., van Oorschot P. and Vanstone S.: ”Handbook of Applied Cryptography”. CRC Press. ISBN: 0-8493-8523-7. 1996.
13. Trappe W., Washington L.: ”Introduction to Criptography with Coding Theory”. Prentice Hall. ISBN: 0-13-061814-4. 2002.
14. van de Graaf J.: ”Anonymous One Time Broadcast Using Non Interactive Dining Cryptographer Nets with Applications to Voting”. Publicado en: ”Towards Trustworthy Elections”. Ps. 231-241. Springer-Verlag Berlin, Heidelberg. ISBN: 978-3-642-12979-7. 2010.
15. van de Graaf J.: ”Voting with Unconditionally Privacy: CFSY for Booth Voting”. IACR Cryptology ePrint Archive. Ps. 574-579. 2009.
16. van de Graaf J., Montejano G., García P.: “Optimización de un esquema “Ocuppancy Problem” orientado a E – Voting”. Memo-

rias del XV Workshop de Investigadores en Ciencias de la Computación 2013 (WICC 2013). ISBN: 9789872817961. Ps. 749 – 753.
17. van de Graaf J., Montejano G., García P.: “Manejo de Colisiones en un Protocolo Dining Cryptographers”. Anales del Workshop de Seguridad Informática (WSegI, ISSN: 2313-9110) de las 42 Jornadas Argentinas de Informática e Investigación Operativa (JAIIO 2013, ISSN: 1850-2776). Ps. 35 a 50.

Datos de Contacto:

Jeroen van de Graaf. Departamento de Ciência da Computação – Universidade Federal de Minas Gerais. Av. Antônio Carlos 6627 - Prédio do ICEx – (31270-010) Belo Horizonte - Minas Gerais - Brasil
E-mail: jvdg.ufmg@gmail.com

Germán Montejano. Facultad de Ciencias Físico Matemáticas y Naturales - Universidad Nacional de San Luis. Avenida Ejército de Los Andes 950 - (5700) San Luis – San Luis – Argentina.
E-mail:gmonte@unsl.edu.ar

Pablo García. Facultad de Ciencias Exactas y Naturales – Universidad Nacional de La Pampa. Uruguay 151 – (6300) Santa Rosa - La Pampa – Argentina.
E-mail:pablogarcia@exactas.unlpam.edu.ar