Manejo de Colisiones en un Protocolo Non Interactive Dining Cryptographers

Jeroen van de Graaf[1], Germán Montejano[2], Pablo García[3]

[1] Departamento de Ciência da Computação - Universidade Federal de Minas Gerais - Av. Antonio Carlos, 6627 – 31270-010 - Belo Horizonte – Minas Gerais - Brasil
jvdg@dcc.ufmg.br
http://www.dcc.ufmg.br/jvdg
[2] Departamento de Informática - Universidad Nacional de San Luis Ejército de los Andes 950 – (5700) San Luis – San Luis – Argentina
gmonte@unsl.edu.ar
http://www.unsl.edu.ar
[3] Departamento de Matemática - Universidad Nacional de La Pampa Av. Uruguay 151- (6300) Santa Rosa - La Pampa - Argentina
pablogarcia@exactas.unlpam.edu.ar
http:/www.exactas.unlpam.edu.ar

Resumen Dining Cryptographers es un protocolo criptográfico presentado por Chaum en [2] cuya característica más notable es la de proveer nivel incondicional de seguridad para la privacidad de los mensajes publicados por un grupo de usuarios de una red. El protocolo exige la concurrencia online de los participantes. Sin embargo, existen múltiples situaciones prácticas en las que esta condición no necesariamente se cumple. Particularmente, voto electrónico. En consecuencia, y atendiendo a la riqueza del protocolo, van de Graaf presenta en [14] una derivación que denominó Non Interactive Dining Cryptographers (NIDC) que intenta explotar todo el potencial de la propuesta original, intentando cubrir un rango mayor de problemas a los que el protocolo pueda aplicarse. El presente trabajo tiene por objetivo central proponer técnicas concretas de optimización del espacio destinado al almacenamiento de los sufragios en un modelo NIDC.

Keywords: Colisión, slot, seguridad incondicional, anonimato, voto electrónico, Dining Cryptographers, Birthday Paradox, Occupancy Problem.

1. Introducción y motivación

La posibilidad de que un modelo de voto electrónico sea exitoso está directamente relacionada con las ventajas concretas que el mismo pueda brindar con respecto al esquema manual tradicional. En ningún caso, reemplazar un modelo por otro que resulte menos funcional puede resultar de interés.

Particularmente, van de Graaf en [14] indica que la mayoría de los esquemas de voto electrónico implementan seguridad incondicional para proteger el proceso, pero seguridad computacional en lo referido a la privacidad. Afirma además

que tal orden debe invertirse, dado que el proceso de votación se desarrolla en un tiempo finito y que, una vez finalizado, los resultados se publican. El anonimato en cambio, debería ser protegido indefinidamente.

Por lo tanto, este trabajo busca analizar protocolos que provean tales niveles de seguridad. Dining Cryptographers tiene esa propiedad. En particular, una derivación propuesta por van de Graaf (Non Interactive Dining Cryptographers), resulta aplicable a voto electrónico por no exigir la concurrencia en el tiempo de todos los participantes. Específicamente, se analiza la problemática de la pérdida de sufragios en tal esquema y se propone una variante que optimiza la utilización del espacio de almacenamiento de los mismos, minimizando la probabilidad de pérdida de votos.

2. Dining Cryptographers

En [2], Chaum presenta un modelo de comunicación con características extraordinarias, a la que llamó "Dining Cryptographers". El protocolo presenta niveles de seguridad incondicional, en lo referente al anonimato asociado a la emisión de una información determinada, a través de canales públicos.

El problema inicial se presenta basado en tres participantes que sólo intercambian un bit de información, pero se generaliza a cualquier cantidad de participantes y a cualquier volumen de información, de manera natural y sin complicaciones de importancia. El modelo 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:

$$P(Cara) = P(Ceca) = \frac{1}{2}$$

Para comprender la causa subyacente por la que el anonimato queda asegurado, es necesario colocarse en el lugar de un comensal que no haya abonado la cuenta. Obviamente, el caso en el que el pagador sea externo garantiza el anonimato en base a los supuestos del protocolo. Por lo tanto, se debe analizar el caso en que el pagador pertenezca al grupo de criptógrafos. Esto solamente puede presentar dos alternativas:

Las dos monedas que él ve son iguales. En este caso, necesariamente, uno de los demás participantes indicó "iguales" y el otro "diferentes". Por lo tanto, si la moneda que él no pudo observar fuese igual a las que vió, aquel participante que declaró "diferentes" es el pagador. Y, por el contrario, si la moneda desconocida fuera diferente de las que él pudo ver, el pagador es quien expresó "iguales". Sin embargo, ambos estados son equiprobables. Por lo tanto, no es posible obtener información relacionada con la identidad del pagador.

Las dos monedas que él observa son distintas. En tal caso, es inevitable que los otros dos criptógrafos expresen resultados coincidentes. Si ambos dicen "diferentes", el pagador será el participante que se encuentre más cerca de la moneda que coincide con el resultado de la moneda oculta. Y si ambos proclaman "iguales", el pagador es quien se encuentra más cerca de la moneda que difiere del valor de la moneda oculta. Tal como en el caso 1, ambos escenarios son equiprobables. En consecuencia, no se puede obtener ninguna información que delate la identidad del pagador.

Para confirmar la seguridad del protocolo podemos definir la noción de *Vista (View):*

Una vista es una variable aleatoria basada en el conjunto de información con la que cuenta un participante determinado al finalizar el proceso.

Concretamente, la vista de un participante estará conformada por:

- Sus entradas.
- Sus bits aleatorios.
- Todos los mensajes enviados y recibidos.

Si podemos probar que la vista de la que dispone un usuario determinado no permite determinar las elecciones realizadas por los demás participantes en ningún caso, el anonimato queda garantizado. En particular, para analizar el protocolo Dining Cryptographers, podemos distinguir los siguientes elementos y analizar cuáles de ellos están visibles para cada participante:

Las monedas: de acuerdo a la mecánica del modelo DC, cada participante ve su propia moneda y la de su vecino de la izquierda. Denominaremos:

$$r_i \in \{Cara, Ceca\}$$

al valor obtenido en la acción de lanzar la moneda i.

La información de inversión: Este elemento será $m_i \in \{True, False\}$. Si m_i es $True$, implica que el participante pagó la cuenta y, por lo tanto, miente al expresar el resultado que observa de la comparación de las dos monedas que puede observar. Un valor $False$ implica lo contrario.

La información de comparación de dos monedas: Para este dato se utilizará $x_i \in \{Iguales,\ Diferentes\}$. Obviamente un valor iguales implica que el participante i declara que los valores de ambas monedas que puede ver son coincidentes.

Por lo tanto, puede analizarse en esos términos la seguridad del protocolo. Bastará con analizar las vistas que un participante determinado dispone ante todos los casos posibles, dado que la simetría garantiza que las conclusiones pueden generalizarse para todos los participantes.

Se observa entonces las vistas que el participante P_1 podría tener a disposición. Los casos son los siguientes:

El pagador es externo. En este caso, el anonimato está garantizado.
P_1 **es el pagador.** Está situación también es trivial.
P_2 **es el pagador.** La vista que P_1 tiene a disposición es la siguiente:

$$V_1 = (r_1, -, r_3, m_1, -, -, x_1, x_2, x_3)$$

P_3 **es el pagador.** Ante esta situación, P_1 ve:

$$V_1 = (r_1, -, r_3, m_1, -, -, x_1, x_2, x_3)$$

Queda claro que las vistas de los dos últimos casos tienen la misma distribución. Concretamente, los valores de r_1 ,r_3, m_1 y x_1 serán exactamente iguales en ambos casos. x_2 y x_3 , en cambio, presentarán valores opuestos dependiendo de quien haya pagado la cuenta. Sin embargo, eso no le entrega información adicional a P_1 , porque tales valores dependen de r_2 , valor que él no conoce y que presenta equiprobabilidad de tomar cualquiera de los dos valores posibles.

Por lo tanto, si alguno de sus compañeros abonó la cena, P_1 no puede distinguir quién fue, dado que:

$$P(P_2(\ pagó)) = P(P_3(\ pagó)) = \tfrac{1}{2}$$

Queda demostrada la seguridad del protocolo original de Chaum, contando con que los tres criptógrafos se comportarán de manera honesta. Obviamente, es casi imposible que todos los participantes mantengan tal conducta, sobre todo cuando los intereses en juego son importantes como en el caso de voto electrónico. Es necesario implementar herramientas externas para que el protocolo sea capaz de detectar y administrar intentos fraudulentos.

3. Non Interactive Dining Cryptographers

Por lo expuesto en la sección anterior, el protocolo Dining Cryptographers reviste enorme interés en aplicaciones criptográficas. Sin embargo, la versión original presenta la limitación de exigir la concurrencia en el tiempo de todos los participantes. 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.

El protocolo original de [7] se basa en *Mix Nets*. En cambio, en [14] se utiliza *Non Interactive Dining Cryptographers.*. Se describe a continuación dicho esquema. Se deja constancia, sin embargo, que es un objetivo futuro reemplazar el mismo (que se considera ineficiente por la gran cantidad de operaciones a nivel de bit), por otro, basado en logaritmos discretos y commitments de Pedersen.

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 cada mensaje.

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.

Para pasar a una descripción detallada de NIDC, vale mencionar que la única restricción inevitable pasa por la longitud de los mensajes involucrados, que deberán ser todos de la misma longitud. Esta condición es obligatoria y tiene que ver con la interrelación entre los mensajes y con algunos significados semánticos específicos relacionados con la "posición" que una información determinada ocupa.

Aparecen entonces los flujos de información del tipo "desafío y respuesta". En efecto, tras la publicación del commitment se produce un esquema donde los participantes puedan "desafiar" a los efectos de verificar que el mensaje que se desea publicar no contradice el commitment previo. El modelo descripto se basa en la heurística de Feige-Shamir ([4]): con posterioridad al commitment, se implementa un modelo que permita a los participantes verificar que el mensaje es coherente con el commitment inicial. Si la implementación es apropiada, el modelo se comporta de manera segura y, simultáneamente, es una manera apropiada de reemplazar la falta de concurrencia temporal.

Concretamente, NIDC implementa una estrategia basada en BCX (Bit Commitments con XOR). La solución adoptada consiste en un protocolo integrado por commitments basados en funciones de hash. El modelo exacto es descripto en detalle en [14]. También es posible encontrarlo en la literatura por su abreviatura (BCX). La idea es representar cada BCX como un vector de pares de "bit commitments" simple, tal que si a cada par se le aplica un XOR, el resultado obtenido es el valor del bit comprometido. La técnica, entonces, habilita la posibilidad de desafíos sobre una mitad del bit commitment, pero sin revelar su valor.

Vamos a ejemplificar. Supongamos que se desea representar un valor de $x = 1$, lo hacemos a través de un vector $\overline{X}$, que podría incluir los siguientes BCX:

$$(0,1) - (1,0) - (0,1) - (0,1) - (0,1)$$

Obviamente, el conjunto de commitments elegidos podría haber sido otro, siempre que se verifique la condición de que cada uno de los BCX sea $\overline{01}$ o $\overline{10}$. También es importante aclarar que el número de commitments implementados (cinco, en este caso) es solamente a los efectos de ejemplificar. Concretamente, el

número resulta bajo si se desea un nivel de seguridad significativo. Más adelante se analizará este punto con más detalle

Por tratarse de commitments, el objetivo es demostrar la igualdad entre dos elementos, sin dar a conocer los contenidos. Luego supongamos un vector $\overline{Y}$, que también corresponde a un valor $y = 1$ pero se representa por un conjunto de BCX diferente:

$$(1,0) - (1,0) - (0,1) - (1,0) - (0,1)$$

El Cuadro 1 muestra la mecánica del protocolo:

	Emisor	**Receptor**
Genera $\overline{X'}$, con $j \in Z^+$ bit commitments para $\overline{X}$		
Genera $\overline{Y'}$, con j bit commitments para $\overline{Y}$		
Para cada par de commitments $(\overline{X'}_l, \overline{Y'}_l)$, con $(l \in 1..j)$		
Calcula $i \in \{0,1\}$		
0 indica que los commitments son iguales		
1 indica que los commitments son diferentes		
	$i \longrightarrow$	
		Elige aleatoriamente $g \in \{0,1\}$
	$\longleftarrow g$	
$Si\ g = 0 \Rightarrow r$ =igualdad entre bits izquierdos		
$Si\ g = 1 \Rightarrow r$ =igualdad entre bits derechos		
	$r \longrightarrow$	
		Verificación correcta $\Rightarrow m = OK$
		Verificación incorrecta $\Rightarrow m = CANCEL$
	$\longleftarrow m$	

Cuadro 1: Protocolo basado en BCX

Basado en ese modelo general, van de Graaf propone en [48], un protocolo que permite comprometer el valor de cada bit del mensaje original, manteniendo la privacidad de manera incondicional, que se basa en los siguientes pasos:

- Se genera una permutación de el vector de BCX.
- Se compromete el orden exacto de dicha permutación. Esto significa que se compromete el conjunto de sustituciones exactas del vector original en el vector permutado.
- El desafiante podrá elegir entre dos retos, de manera aleatoria:

1. Si opta por el primer desafío, el retador exige la apertura de la permutación. Al hacerlo, el emisor demuestra la igualdad de los BCX que no contienen mensajes.
2. Si se elige el segundo desafío, lo que se exige es la apertura de todos los BCX, excepto el que contiene el mensaje. Es claro que todos deben dar cero.

Es sencillo advertir que el protocolo no revela el contenido. En el caso de bit commitments y XOR, cada fraude individual tendrá una probabilidad igual a $\frac{1}{2}$ 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 $(\frac{1}{2})^d$, donde $d \in Z^+$ es la cantidad de pares que se implementen.

4. Colisiones y Birthday Paradox

En un protocolo NIDC, el anonimato queda garantizado por la elección aleatoria de la ubicación de un sufragio. En consecuencia, es posible que dos o más votantes elijan el mismo slot. Tal situación se denomina *colisión* y da lugar a la pérdida de todos los votos coincidentes. Si se implementa un vector simple para el almacenamiento, aparece como referencia indicutible Birthday Paradox:

En un grupo de 23 personas la probabilidad de que dos cumplan años el mismo día es cercana a $\frac{1}{2}$.

Obviamente, tal afirmación es desfavorable, dado que, la proporción de slots vacíos es alta y la probabilidad de que no se produzcan colisiones muestra valores muy alejados de algo que pueda resultar aceptable en un esquema de voto electrónico. Por otro lado, $N = 23$ es un valor mucho más pequeño que aquellos que puedan resultar de interés para voto electrónico. Por ejemplo, un recinto en Brasil es de alrededor de 500 votantes. La relación entre el número de votantes y la cantidad de slots, mateniendo constante la probabilidad de colisión, se expresa con la fórmula:

$$N \approx 1{,}17\sqrt{T}$$

Se busca, entonces, un enfoque alternativo que permita optimizar el almacenamiento destinado a los sufragios. Lo primero que se observa es que aún en el mejor caso (las 23 personas cumplen años en fechas diferentes), la cantidad de slots que no serán utilizados es 342, entonces tenemos aproximadamente un 6,3 % de slots ocupados y un 93,7 % de slots vacíos. En consecuencia, por cada slot que contiene un sufragio, hay más de 15 que no reciben votos.

Aparece entonces la idea de buscar una manera de aprovechar de manera más eficiente tal almacenamiento. La propuesta consiste en dividir la totalidad

de slots en Q>1 canales paralelos y depositar una ocurrencia de cada sufragio en cada uno de los canales. El disparador de tal hipótesis es una propiedad de los sucesos independientes:

$$P(A \cap B) = P(A)P(B)$$

Es evidente que un voto se perderá solamente si colisiona en todos los canales. Si bien el número de colisiones va a aumentar, dado que cada canal tendrá una medida menor que el vector único, es posible que se obtenga una optimización basada en las réplicas. En las dos secciones siguientes se analiza el problema desde dos perspectivas diferentes: aproximaciones teóricas y simulaciones.

5. Aproximación teórica

Sean N, T ,S y ε, tal que:
$N = \#$ votos.
$T = \#$ slots totales.
$S = \#$ slots en cada canal.
$\varepsilon = \frac{N}{S}$.

Inicialmente se analiza la situación si se implementa un único vector, donde $S = T$. En [16] se presentan varias estrategias basadas en analizar la distribución de probabilidades. Por ejemplo, Feller en [5] propone una aproximación basada en una distribución de Poisson, que se puede resolver en general aplicando la aproximación de Stirling para el cálculo de los factoriales.

En este documento se propone otro enfoque, que resulta más simple que el anterior por calcular solamente valores esperados en lugar de distribución de probabilidades.

Para un voto determinado, la probabilidad de que el mismo caiga en el slot 1 es $p = \frac{1}{S}$. En consecuencia, la probabilidad de que no caiga en el slot 1 es $q = 1 - p = (1 - \frac{1}{S})$

Generalizando a N votos, obtenemos una distribución binomial con parámetros N y P. Sea:

$X_k =$"Se almacenan exactamente k votos en el slot 1", con $k \in Z^+$

$$P(X_k) = \binom{N}{k} p^k q^{N-k}$$
$$P(X_k) = \binom{N}{k} (\tfrac{1}{S})^k (1 - \tfrac{1}{S})^{N-k}$$

Teniendo en cuenta que:

$$\lim_{x \to \infty} (1 + \tfrac{1}{n})^n = e$$

Se puede afirmar que:

$$P(X_0) = (1 - \frac{1}{S})^N \approx e^{-\varepsilon}$$
$$P(X_1) = N\frac{1}{S}(1 - \frac{1}{S})^{N-1} \approx \varepsilon e^{-\varepsilon}$$
$$P(X_2) = \frac{N(N-1)}{2}(\frac{1}{S})^2(1 - \frac{1}{S})^{N-2} \approx \frac{1}{2}\varepsilon^2 e^{-\varepsilon}$$

Estas probabilidades también representan el número esperado de votos en el slot 1. Por lo tanto, es posible hallar la frecuencia esperada, que nos expresa el porcentaje.

Tomando en cuenta que $\lim_{x\to\infty}(1 + \frac{1}{n})^n = e$, para $N = S = 1000$, $\varepsilon = 1$. Por lo tanto:

$$P(X_0) = P(X_1) = e^{-1} \approx 0{,}3678$$

Análogamente, para $N = 500, S = 1000$, tenemos $\varepsilon = \frac{1}{2}$, en cuyo caso:

$$P(X_0) = e^{-\frac{1}{2}} \approx 0{,}6065$$
$$P(X_1) = \frac{1}{2}e^{-\frac{1}{2}} \approx 0{,}3032$$

Sea $E(k) = \#$ slots esperado que contengan k votos. Su valor se obtiene de la siguiente manera:

$$E(k) = S\ p(X_k)$$

Esto coincide con la aproximación de Poisson enunciada en [5]:

$$E(Poisson(\lambda)] = \lambda$$
$$\lambda[X_0] = Se^{-\frac{N}{S}}$$

Para $k = 1$, tenemos:

$$\lambda[X_1] = \frac{(ne^{\frac{-r}{n}})}{k!}(\frac{r}{n})^k = \frac{Se^{(-\varepsilon)}}{k!}\varepsilon = S\varepsilon e^{-\varepsilon}$$

Para $k = 2$:

$$\lambda[X_2] = \frac{ne^{(\frac{-r}{n})}}{k!}(\frac{r}{n})^k = \frac{Se^{-\varepsilon}}{2}\varepsilon^2 = S\varepsilon^2 e^{-\varepsilon}$$

La aproximación de Poisson mejora su calidad cuando S y N tienden a infinito. La fórmula anterior se relaciona con S. Resulta más interesante obtener una relación con la cantidad de votos exitosos, es decir que para $k = 1$, se divide $E[1]$ por N y obtenemos:

$$\frac{S\varepsilon e^{-\varepsilon}}{N} = e^{-\varepsilon}.$$

De esa manera obtenemos una fórmula para calcular el valor esperado de la variable *Porcentaje de Votos Perdidos*.

Es posible generalizar el enfoque a Q canales, con $Q > 1$. Por ejemplo, para $S = N = 1000$:

$$s[1] = 1000\ e^{-1} \approx 368$$

En consecuencia, para $Q = 1$:

$$p(voto\ exitoso) \approx 0{,}36 \text{ y } p(voto\ perdido) \approx 1 - 0{,}36 = 0{,}64$$

Para $Q = 2$, un voto se pierde si colisiona en los dos canales:

$$(voto\ exitoso) = 1 - 0{,}39 \approx 0{,}61 \text{ y } P(voto\ perdido) = 0{,}64^2 \approx 0{,}39$$

El mismo esquema se generaliza $\forall Q > 2$.

6. Simulaciones

De manera complementaria a los desarrollos teóricos, como parte del proyecto se implementa un simulador de actos eleccionarios. El mismo tiene dos objetivos claros:

1. Confirmar de manera práctica los resultados matemáticos. Las fórmulas expuestas en la sección anterior tienen su correlato en los valores obtenidos por simulación.
2. Permitir una observación más clara de las variables involucradas, a los efectos de confirmar o descartar ideas surgidas de manera intuitiva.

El simulador recibe las siguientes entradas:

- $N = \#$ votos.
- $T = \#$ total de slots.
- $C : \#$ canales paralelos.
- $R : \#$ procesos eleccionarios que se simulan en la presente sesión del simulador.

Y entrega los siguientes resultados como salida:

Votos efectivos acumulados (*vea*) : indica la cantidad exacta de votos que no se perdieron, teniendo en cuenta la totalidad de las corridas.

Votos perdidos acumulados (*vpa*): expresa la cantidad de votos perdidos en total. Obviamente:

$$NR = vea\ vpa$$

Cantidad de corridas con Pérdidas (*cccp*).
Cantidad de corridas sin pérdidas (*ccsp*). Es evidente que:

$$R = cccp + ccsp$$

Media del generador aleatorio (*mga*): este valor se obtiene a los efectos de verificar la calidad de la muestra generada.

Porcentaje de Votos Perdidos (*pvp*): referido a la totalidad de simulaciones de una sesión determinada.

Mejor caso (*mc*): indica cuál es el menor número de votos perdidos en algún acto eleccionario, considerando todas las corridas.

Peor caso (*pc*): idem al anterior para indicar el mayor número de votos perdidos en alguna corrida.

Para todas las simulaciones realizadas, dos parámetros fueron mantenidos constantes:

- Cantidad de votantes: 480.
- Cantidad de corridas: 1.000.000.

Alos efectos de ilustrar el funcionamiento del simulador, se muestran los resultados obtenidos en un par de sesiones.

En el primer caso, se implementaron 4800 slots y se vigila la variable "cantidad de corridas sin pérdidas". El Cuadro 2 muestra los resultados obtenidos con la implementación de 1, 2, 4, 5, 6, 8, 10 y 12 canales paralelos. En la misma puede observarse que el valor óptimo se obtiene con la utilización de 6 u 8 canales.

Cantidad de Canales	**Porcentaje de Votos Perdidos**
1	0,0913168
2	0,0304556
3	0,0156735
4	0,0103669
5	0,0083182
6	0,0073571
8	0,00732933
10	0,0089224
12	0,0121763

Cuadro 2: Cantidad de Corridas con Pérdidas (4800 slots)

La mejora progresiva que se obtiene aumentando desde 1 a 6 canales es producto de la propiedad mencionada previamente relacionada con la independencia de los sucesos. Tal optimización se pierde cuando se utilizan más de 8 canales porque la cantidad de colisiones en cada canal unitario aumentan significativamente porque el tamaño de cada vector unitario es igual a la cantidad de votantes en el caso de utilizar 10 canales e incluso menor si se usan 12 canales, en cuyo caso cada canal tendrá sólo 400 slots, para 480 votantes.

El segundo ejemplo es similar, pero se implementan 9600 slots. Como muestra el Cuadro 3, la eficiencia crece de manera continua con el agregado de canales. Por ejemplo, con un sólo canal, sobre un millón de corridas, no se perdieron votos solamente en 5. En cambio, con 12 canales, se registaron exactamente 971901 corridas sin pérdidas.

Cantidad de Canales	**Porcentaje de Votos Perdidos**
1	0,0474395
2	0,00850993
3	0,00246987
4	0,000961267
5	0,000461467
6	0,000259
8	0,0001178
10	7,19E-005
12	5,94E-005

Cuadro 3: Cantidad de Corridas con Pérdidas (9600 slots)

Obviamente, los resultados mejoran si se aumenta el número total de slots. Sin embargo, el punto que se desea resaltar es la optimización obtenida, al incorporar más canales paralelos, con un mismo número total de slots.

7. Conclusiónes e problemas abiertos

La primera conclusión de este trabajo es que es posible optimizar el espacio destinado al almacenamiento de sufragios mediante la utilización de canales paralelos. Concretamente, la variable *pvp* (porcentaje de votos perdidos), puede ser estimada aplicando la fórmula:

$$pvp = (1 - e^{\frac{-N}{S}})^Q$$

Si se elijen N y S constantes, el valor óptimo de Q se obtiene con la expresión:

$$Q = (ln(2))\frac{T}{N}$$

Efectivamente, para valores mayores de Q, la optimización decae por un aumento significativo de las colisiones en cada canal, por aproximarse demasiado los valores de S y N.

A futuro se planea investigar la manera exacta en que se comporta una variante: implementar un número de canales Q, pero replicar los sufragios una cantidad de veces Q', con $Q' < Q$. Se modificará el código del simulador para permitir esa variante y en caso de obtener resultados favorables, se analizará su comportamiento matemático.

Se busca también relacionar las conclusiones obtenidas sobre la variable *pvp* (porcentaje de votos perdidos) al comportamiento de *cccp* (cantidad de corridas con pérdidas). Se considera que esta última magnitud resulta más significativa en términos de evaluar la calidad de un protocolo de este tipo, dado que resulta de máximo interés indicar, frente a un acto eleccionario, la probabilidad exacta de que en el mismo no se pierdan votos.

Otro objetivo de esta línea de investigación es determinar si es posible utilizar herramientas para recuperación de colisiones y el alcance de las mismas. Por ejemplo, si en cada réplica se almacena información que indique donde se almacenaron las demás copias, un sufragio que haya colisionado en todos los canales podría recuperarse mediante una operación XOR si en alguna de esas colisiones coincide con un sufragio que tiene alguna instancia válida. Se desea investigar el alcance exacto de tales herramientas y ver si es posible encontrar alternativas para colisiones múltiples.

8. Trabajos Relacionados

El presente trabajo se deriva fundamentalmente de [14], donde se deja para investigaciones futuras la obtención de optimizaciones en dos puntos muy concretos: manejo de colisiones y administración de maniobras fraudulentas. Al momento de presentar este documento, el primer punto se encuentra avanzado y se exponen los resultados obtenidos. Para el segundo tópico, se halla en desarrollo un protocolo para reemplazar el expuesto en la sección 3, por uno basado en logaritmos discretos y commitments de Pedersen. Se considera factible que la nueva propuesta mantenga los niveles de seguridad con una mayor eficiencia.

Es bueno mencionar que lo expuesto en la secciones 4 y 5, contradice el supuesto original de [14], demostrando que dividir T en Q slots es sustancialmente mejor que replicar Q veces cada sufragio en el mismo canal.

Otros documentos relacionados con el presente trabajo son [1] y [8], que proponen técnicas concretas que apuntan a la detección de fraudes y que muestran similitudes con el protocolo en elaboración. El problema de las colisiones, en cambio, no es abordado en ninguna de las dos referencias.

Referencias

1. Bos J.: "Practical Privacy"- Technische Universiteit Eindhoven, ISBN: 90-6196-405-9. 1992.
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/volume4-number1/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". Memorias del XV Workshop de Investigadores en Ciencias de la Computación 2013 (WICC 2013). Ps. 749 - 753. ISBN: 9789872817961