Codificación de Sufragios con Detección de Colisiones en NIDC con Canales Paralelos de Slots

Pablo Marcelo García
Departamento de Matemática
FCEyN – UNLPam
Santa Rosa – La Pampa – Argentina
pablogarcia@exactas.unlpam.edu.ar

Silvia Gabriel Bast
Departamento de Matemática
FCEyN – UNLPam
Santa Rosa – La Pampa – Argentina
silviabast@exactas.unlpam.edu.ar

Germán Antonio Montejano
Departamento de Informática
FCFMyN – UNSL
San Luis – San Luis – Argentina
gmonte@unsl.edu.ar

Estela Marisa Fritz
Departamento de Matemática
FCEyN – UNLPam
Santa Rosa – La Pampa – Argentina
fritzem@exactas.unlpam.edu.ar

Resumen

El presente documento propone una codificación a nivel de bits para optimizar el comportamiento de una técnica de almacenamiento para datos anónimos basada en la implementación de canales paralelos.

Si bien originalmente el modelo fue diseñado para su aplicación en productos de voto electrónico, es posible generalizarlo a cualquier problema práctico que requiera anonimato y exija niveles muy altos de seguridad respecto de la pérdida de información.

El esquema propuesto permite detectar las colisiones y, en consecuencia, minimizar las pérdidas de información que las mismas generan, llevándolas al valor que se exija. Simultáneamente, se hace un análisis probabilístico para cuantificar la probabilidad de una interpretación incorrecta de un sufragio determinado.

Introducción

En el marco de una Línea de Investigación que comienza en 2013 y que se presenta formalmente en [1], se estudian temas relacionados con ciberseguridad. Dentro de ese ámbito, se han realizado una serie de avances relacionadas con la problemática del voto electrónico.

En ese sentido, se postula que en un sistema de E – Voting es necesario otorgar seguridad incondicional a la privacidad del votante ([2]). Esta conclusión surge de observar que el anonimato debe ser protegido de manera indefinida, mientras que la integridad de los datos debe resguardarse por un tiempo finito. Esta afirmación se opone a los modelos basados en Mix Net ([3]), que protegen incondicionalmente el desarrollo de la votación, pero proveen seguridad computacional para el anonimato.

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

En consecuencia, resultan de interés aquellos protocolos que proporcionen tal nivel de seguridad. En particular, el protocolo Dining Cryptographers, detalladamente descripto en [4], constituye una propuesta muy ingeniosa y responde a los requerimientos mencionados. Se enfoca el análisis en un derivado del mismo, denominado Non Interactive Dining Cryptographers (NIDC, [5]), que agrega al modelo original la posibilidad de que no todos los participantes se encuentren simultáneamente online, condición que aparece en muchos problemas reales.

Non Interactive Dining Cryptographers (NIDC)

NIDC se deriva de Dining Cryptographers, que es un protocolo que presenta niveles de seguridad incondicional para el anonimato asociado a la emisión de una información determinada, a través de canales públicos, que 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 cuenta fue abonada por alguien que no pertenece al grupo. Ellos pretenden 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.

La demostración formal de que el esquema es seguro se expone por primera vez en [6] y se formaliza con mayor precisión en [7], utilizando el concepto de Vista (View):

Una vista (View) 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:

- Las entradas que él mismo aporta al proceso.
- Sus conjuntos de bits aleatorios.
- Todos los mensajes que haya enviado o recibido.

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, 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 definirse en esos términos la seguridad del protocolo.

Será suficiente con analizar las vistas que un participante determinado dispone, ante todos los casos posibles, dado que la simetría garantiza que las conclusiones pueden generalizarse.

Se observan entonces las vistas que el participante P1 podría tener a disposición.

Los casos son los siguientes:

- El pagador es externo. En este caso, el anonimato está garantizado.

- *P1* es el pagador. Está situación también es trivial.

- *P2* es el pagador. La vista que *P1* tiene a disposición es la siguiente:

  $$V1 = (r_1; -; r_3; m_1; -; -; x_1; x_2; x_3)$$

- *P3* es el pagador. Ante esta situación, *P1* ve:

  $$V1 = (r_1; -; r_3; m_1; -; -; x_1; x_2; x_3)$$

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 *P1*, porque tales valores dependen de r_2, valor que él no conoce y que presenta equiprobabilidad de tomar cualquiera de los dos valores posibles.

Se define:

P_i = *"El criptógrafo i-ésimo abonó la cuenta"*

Si alguno de sus compañeros abonó la cena, P1 no puede distinguir quién fue, dado que:

$$Pr(P2) = Pr(P3)$$

En consecuencia, Dining Cryptographers reviste gran interés en aplicaciones criptográficas. Sin embargo, presenta la limitación de exigir la concurrencia en el tiempo de todos los participantes. Existen muchas aplicaciones que requieren anonimato incondicional, pero que muestran características asincrónicas. En consecuencia, surge una metodología que mantiene los niveles de seguridad del esquema original pero no exige la concurrencia temporal de todos los participantes Tal esquema se conoce como Non – Interactive Dining Cryptographers (NIDC) ([5]), que combina el concepto de Firmas Ciegas descripto detalladamente en [8] con una derivación del protocolo de Chaum que se presenta en [4]:

Por tratarse de un protocolo sin retroalimentación, presenta algunas características novedosas. 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.

El anonimato es garantizado directamente por Dining Cryptographers de acuerdo al análisis de las vistas (views) realizado previamente. En consecuencia, si se prepara un protocolo que permita distinguir mensajes, los mismos serán interpretados evitando la posibilidad de conocer la autoría de cada uno de ellos.

La protección de la información circulante sólo debe soportar el lapso de tiempo que corresponda al proceso de votación. Todas las firmas se darán a conocer una vez cerrada la elección. Hacer pública esa información deriva en un aumento significativo de la transparencia del procedimiento.

Aparecen entonces los flujos de información del tipo desafío y respuesta, que permiten verificar que el mensaje que se desea publicar no contradice ninguna afirmación previa. Esto se implementa mediante la utilización de un esquema commitment.

En criptografía, se denomina “commitment” a cualquier técnica que pueda aplicarse para “comprometer” una información de manera que, al finalizar el proceso, pueda verificarse que la misma no fue modificada.

Existen muchas modalidades de implementación. En particular, NIDC se basa en la heurística de Feige-Shamir [9], con posterioridad al commitment, se implementa un esquema que permite a los participantes controlar que el mensaje es coherente con los valores comprometidos inicialmente. Si la implementación es apropiada, el modelo se comporta de manera segura sin la necesidad 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. 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.

La manera exacta en que se implementa la administración de BCX se describe detalladamente en [5]. En [7] se presenta un protocolo alternativo que mantiene los niveles de seguridad originales pero que resulta mucho más eficiente en términos de las operaciones involucradas. Esa propuesta se basa en propiedades de los logaritmos discretos.

En este trabajo en particular, en cambio, el interés se enfoca en la seguridad de los sufragios. En consecuencia, se analiza en la próxima sección todo lo relacionado con la probabilidad de perder votos en un esquema NIDC.

Almacenamiento de los Sufragios en NIDC

NIDC propone almacenar los sufragios en un vector unidimensional de slots. El anonimato se garantiza por la elección totalmente aleatoria de la posición en la que un voto se almacenar**á**. Tal aleatoriedad trae como consecuencia que la probabilidad de colisión, es decir, que dos o más votos elijan la misma posición, sea positiva. Inicialmente, una colisión deriva en la pérdida de todos los sufragios coincidentes.

Implementar un vector unidimensional nos ubica en el caso descripto por Birthday Paradox ([10]):

“En un grupo de 23 personas, la probabilidad de que haya al menos 2 que cumplan años el mismo da es muy cercana a ½.”

Tal afirmación es muy poco auspiciosa a los efectos de esta investigación. El tamaño del vector asociado resulta relativamente grande con respecto a la muestra y, sin embargo, la seguridad asociada está muy lejos de niveles que pudieran ser aceptables en una aplicación práctica. No tiene sentido un sistema de e-Voting en el que la probabilidad de que se pierda al menos un voto se aproxime a $\frac{1}{2}$.

Gráficamente, el esquema original se muestra en la Figura1.

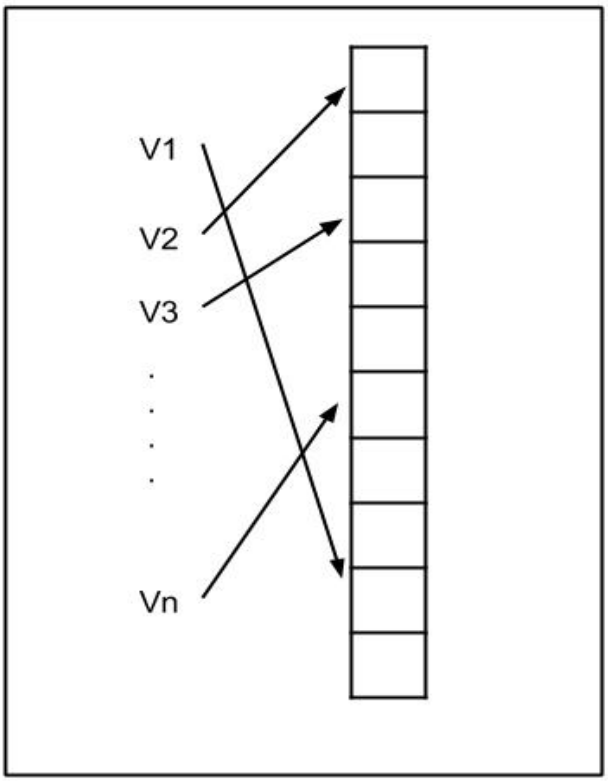

Figura 1: Esquema original de NIDC

La tendencia en el comportamiento de un esquema equivalente a Birthday Paradox se mantiene en valores poco interesantes para cualquier valor que tomen los parámetros.

Para obtener niveles de seguridad que resulten aceptables, la cantidad de posiciones que debe tener el arreglo es muy significativa.

Por ejemplo, en el caso típico (365 días y 23 personas), la probabilidad de colisión es cercana a ½, aún cuando existen, al menos, 342 fechas en las que nadie cumple años. Es dable pensar en un esquema que utilice esa redundancia de manera más eficiente.

Por todo lo expuesto en la introducción, resulta de interés analizar en profundidad la posibilidad de encontrar métodos alternativos que optimicen la utilización del almacenamiento asociado, dado que el esquema de vector simple parece exigir una cantidad significativa de espacio en proporción a los datos efectivos. En [11] se propone una nueva técnica de almacenamiento, que se muestra en la Figura 2.

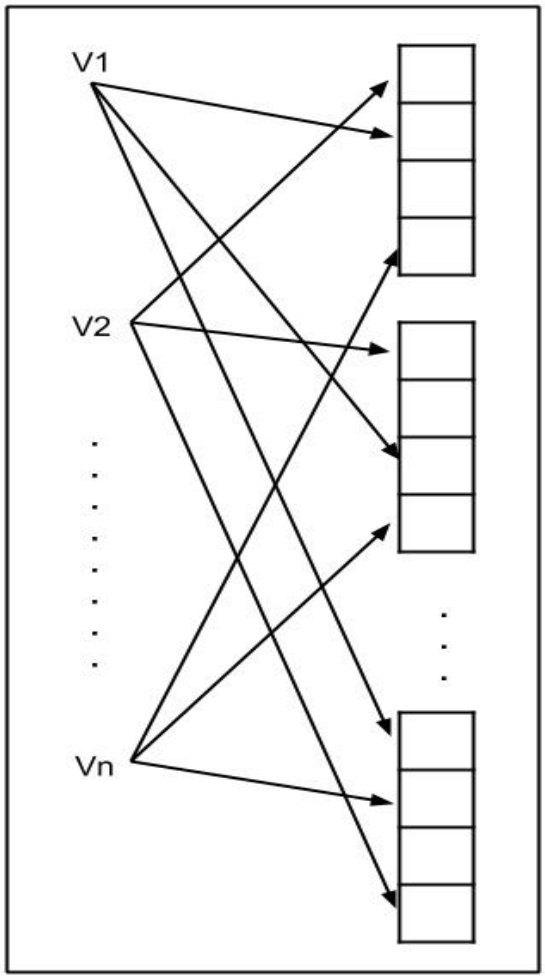

Figura 2: Nuevo Esquema propuesto para NIDC

En [12] se presenta un enfoque alternativo, específicamente relacionado con NIDC, que propone trabajar con varias redes NIDC dispuestas en serie o en paralelo. En este documento, en cambio. se avanza sobre lo expuesto en [13], donde se presentan una serie de ecuaciones que describen el comportamiento del modelo, en base a los siguientes parámetros:

T: # total de slots a implementar.

S: # slots en cada canal individual.

Q: # canales paralelos.

N: # votantes.

V_i: i-ésimo sufragio.

R_{ij}: Suceso que indica que V_i ocupa el j- ésimo slot.

C_{ijk}: Evento que ocurre cuando V_i colisiona con V_j en el canal k.

B_{ij}: Suceso en el que V_i se pierde en el canal j.

A_i: Evento que significa que V_i se pierda en todos los canales.

X: Suceso que expresa que ningún voto se pierde simultáneamente en todos los canales.

L: # sufragios que se pierden en todos los canales simultáneamente.

I: Evento que indica que se produce al menos una colisión múltiple.

En este caso, la ecuación que resulta de interés es la siguiente:

$$Pr(X) > 1 - (\frac{1}{S}(N-1)^Q)^N \quad (1)$$

La fórmula anterior es de sumo interés, dado que permite asegurarle a un usuario del sistema de voto electrónico que la probabilidad de que no se pierda ningún voto es mayor que Y, el cual puede llevarse a cualquier valor deseado manipulando los parámetros del sistema.

En las dos secciones siguientes se profundiza el estudio sobre dos puntos que permiten mayor precisión en el análisis de las colisiones

Proporción de Colisiones múltiples

La fórmula (1) se deduce en base a la suposición de que la proporción de colisiones múltiples es despreciable en comparación con la cantidad de colisiones simples. Concretamente:

Si se considera un canal único, es decir:

$Q = 1$:

Si V_1 se almacena en el slot 1, la probabilidad de que colisione con V_2, esto es, que ambos sufragios sean asignados a dicho slot es:

$$\Pr(C_{121}|R_{11)} = \frac{1}{S}\frac{1}{S} = \frac{1}{S^2} \quad (2)$$

Entonces, la probabilidad de que V_1 y V_2 colisionen en cualquier slot puede expresarse como sigue:

$$\Pr(C_{121}) = \frac{1}{S^2}S = \frac{1}{S} \quad (3)$$

$$\Pr(C_{121}) = \Pr(C_{1j1})\ \forall j \epsilon \{3..N\} \quad (4)$$

Se define:

I="Se producen colisiones múltiples"

En consecuencia, la probabilidad de que V_1 se pierda en el único canal implementado está dada por:

$$\Pr(B_{11}) = \frac{1}{S}(N-1) - Pr(I) \quad (5)$$

De la ecuación previa puede derivarse otra, que es más apropiada en este caso. La misma toma el camino contrario, basándose en la posibilidad de que V_1 no se pierda.

$$Pr(\overline{B_{11}}) = 1 - \frac{1}{S}(N-1) + Pr(I) \quad (6)$$

El valor de *Pr(I)* es positivo. Si se toma como hipótesis que el valor de *Pr(I)* es bajo es posible encontrar una cota para el valor de $Pr(\overline{B_{11}})$.

$$Pr(\overline{B_{11}}) > 1 - \frac{1}{S}(N-1) \quad (7)$$

La ecuación (7) indica la probabilidad de que un voto determinado se pierda en un canal único. Al agregar más canales, totalizando Q, se obtiene:

$$Pr(A_i) = \Pr(B_{i1})^Q > \left(\frac{1}{S}(N-1)\right)^Q \ \forall i \in \{1 \dots N\} \quad (8)$$

Finalmente, se obtiene una cota apropiada para la probabilidad del evento:

X="ningún voto se pierde simultáneamente en todos los canales"

Mediante la aplicación de la ecuación siguiente:

$$\Pr(X) = \Pr(\overline{A_1}) \cap \Pr(\overline{A_2}) \cap \dots \cap \Pr(\overline{A_N}) \quad (9)$$

Pero:

$$\Pr(\overline{A_1}) = \Pr(\overline{A_2}) = \cdots \Pr(\overline{A_N}) \quad (10)$$

Además, por tratarse de eventos independientes:

$$\Pr(X) = Pr(\overline{A_1})^N \qquad (11)$$

O sea:

$$\Pr(X) = 1 - Pr(A_1)^N \qquad (12)$$

Lo cual nos lleva a la fórmula definitiva para la cota propuesta:

$$\Pr(X) > 1 - ((\tfrac{1}{S}(N-1))^Q)^N \qquad (13)$$

De acuerdo a lo expuesto, la precisión de la cota propuesta es dependiente de la proporción de colisiones múltiples que se producen, en promedio, en un acto electoral. A los efectos de verificar tal proporción, se implementa un simulador de actos eleccionarios que totaliza la cantidad de colisiones múltiples. Cada simulación se realizó con 1000 actos eleccionarios. Se incorporan tres variables, que representan promedios de dichas corridas:

- ***SV*:** representa la cantidad de slots vacíos al terminar la simulación
- ***SCS*:** Indica en cuántos slots se produjeron colisiones simples, es decir que sólo involucraron a dos votos.
- ***SCM*:** Informa la cantidad de slots en los que se produjeron colisiones múltiples, es decir de más de dos votos.

Algunos resultados obtenidos se detallan en la tabla 1, a continuación:

N	S	Q	T=Q S	SV	SCS	SCM
10	20	3	60	35,941	4,562	0,656
10	40	6	240	186,317	5,572	0,368
40	80	3	240	145,153	18,414	3,21
40	160	6	960	747,336	24,314	1,455
120	240	3	720	435,378	54,19	9,977
120	480	6	2880	2243,245	69,773	6,527
120	480	10	4800	3742,858	119,216	11,367

Tabla 1: Análisis de Colisiones

Los resultados expuestos permiten afirmar que la proporción de colisiones múltiples es poco significativa para valores que puedan resultar de interés en la presente investigación. Además, los resultados mejoran cuando se utilizan los valores óptimos para los parámetros involucrados, aplicando las fórmulas siguientes, presentadas en [13]:

- Para un número determinado de votantes (*N*), la cantidad de slots (*S*) que se recomienda implementar para cada canal paralelo se expresa por la formula:

$$S = \left| \frac{N}{ln\ 2} \right| + 1 \qquad (14)$$

- Para valores dados de *T* (cantidad total de slots) y *N*, existe un número óptimo de canales paralelos. Tal valor se expresa en la ecuación:

$$Q_{ot} = \ln 2 \frac{T}{N} \quad (15)$$

- Donde Q_{ot} es el número de canales óptimo teórico. Sin embargo, este valor no es entero, razón por la cuál debe ser llevado al número entero siguiente:

$$Q_{oa} = |Q_{ot}| + 1 \qquad (16)$$

Q_{oa} es el número de canales óptimo que puede aplicarse en la práctica. En consecuencia, puede definirse un valor apropiado para la cantidad total de slots:

$$T = Q_{oa}\, S \qquad (17)$$

- El valor esperado para la variable *PVP* (Porcentaje de Votos Perdidos) se obtiene aplicando la ecuación siguiente:

$$|PVP| = \left(1 - e^{-\frac{N}{S}}\right)^Q \qquad (18)$$

Debe mencionarse que al aplicar valores óptimos no solamente se reduce el número de colisiones múltiples; paralelamente, la posibilidad de que se pierdan votos disminuye severamente. Para ejemplificar la situación, la Tabla 2 muestra el comportamiento del sistema para

valores diferentes de *N*, pero aplicando valores óptimos en los parámetros involucrados.

N	S	Q	T=QS	SV	SCS	SCM
30	44	5	220	110,337	25,904	6,725
60	87	4	348	174,279	42,28	11,112
120	174	3	522	261,275	62,355	16,817
240	347	3	1041	519,896	125,585	33,686
480	693	3	2079	1036,23	252,1	66,59

Tabla 2: Análisis de Colisiones con Valores Óptimos

Una codificación a nivel de bits para administrar Colisiones

Otro punto importante es analizar de qué manera pueden codificarse los votos de manera que se obtengan herramientas adicionales a los efectos de administrar correctamente las colisiones.

En NIDC, el almacenamiento de los votos se lleva a cabo mediante una operación XOR. En consecuencia, pueden darse determinados casos que lleven a confusión sobre cómo deben ser interpretados los sufragios. Se propone, en consecuencia, un modelo de codificación que se conforma de tres componentes:

- **Identificador del Voto**: es un conjunto de bits que identifica de manera única el sufragio específico. El conjunto de identificadores válidos se define de manera previa a la elección y se le otorga a un elector sin que el sistema reciba el número de documento asociado.

- **Identificador del Cargo:** es un conjunto de bits que identifica de manera única a un cargo para el que se elegirá un candidato en el presente acto lectivo.

- **Identificación de la Opción:** Conjunto de bits que cuya semántica se relaciona con la elección de un votante específico para un cargo determinado.

Es necesario hacer algunas consideraciones sobre los tres elementos mencionados:

- El Identificador de Voto (*ID_V*) es único para un voto específico. Para asegurar el anonimato, el sistema asigna un *ID_V*, elegido al azar de un conjunto definido previamente al acto electoral. Esta acción debe realizarse de manera separada de la identificación del votante, de manera que en el sistema electrónico no registre relación alguna entre el identificador del voto y el número de documento del elector.

- El Identificador del Cargo (*ID_C*) es conceptualmente innecesario, dado que el cargo podría identificarse por la posición de los bits del *ID_O*. En este ejemplo se mantiene porque, como se verá más adelante, permite aumentar la seguridad en la detección de errores.

- El identificador de la Opción (*ID_O*) indica la opción exacta de un votante para un candidato en un cargo específico. El número de combinaciones válidas estará dada por la cantidad de postulantes para el cargo en cuestión, más una representación para el voto en blanco. La cantidad de bits que se elija para este código definirá la seguridad del mismo.

Se especifican, entonces, los siguientes parámetros:

- *#V*: Cantidad de bits destinados a representar el *ID_V*.

- *#C*: Cantidad de bits que se destinan a almacenar el *ID_C*.

- *#O*: Cantidad de bits que se implementan para representar el *ID_O*

Aparecen, entonces, algunos casos problemáticos que deben analizarse cuidadosamente:

1. Si colisionan dos o más votos cuyos identificadores de Voto, sometidos a operaciones XOR dan como resultado otro *ID_V* válido, podrían aparecer inconvenientes.

Por ejemplo, sean los siguientes Identificadores de Voto:

ID_V1=0101

ID_V2=1100

ID_V3=1001

Es evidente que si se produce una colisión entre los votos 1 y 2:

0101 XOR 1100 = 1001

El resultado obtenido coincide con el *ID_V3*.

Es factible generar un conjunto de *ID_V* que evite esta situación, pero es costoso. Por el contrario, se propone una técnica que permite reducir la probabilidad de errores hasta el nivel que se desee:

"Para evitar que el voto sea interpretado incorrectamente es necesario que el problema sea detectado a partir de la observación de los atributos restantes".

Entonces:

- El Identificador del Cargo (*ID_C*) es único para un cargo determinado. En consecuencia:
    - Las colisiones de orden par, darán por resultado una secuencia de *#C* ceros. Esto puede afirmarse por la propiedad de la operación XOR ([14]).
    - Las colisiones de orden impar dan por resultado un *ID_C* válido. Concretamente, el correspondiente al cargo en cuestión.
- El Identificador de Opción (*ID_O*) es único para cada candidato que se postula para un cargo específico. En consecuencia, la probabilidad de que una colisión genere un ID_O válido está dada por el cociente:

$$\frac{CCV}{CCT} \tag{19}$$

Donde *CCV* es la cantidad de combinaciones que representan votos válidos y *CCT* es la cantidad de combinaciones. *CCV* será dependiente de la cantidad de postulantes para un cargo y *CCT* será proporcional a los bits que se asignen a ese almacenamiento.

2. Si tres votantes colisionan en el mismo slot se da una situación irregular si, mientras el primero votó por un candidato, los otros dos votaron por otro. Concretamente, el *ID_O* es incorrecto para el *ID_V* que figura en la base de datos.
   Por ejemplo, sean:

   - *#V*: 8 bits.
   - *#C*: 4 bits.
   - *#O*: 8 bits.

   Con esos valores, definimos los identificadores asociados:

   Identificadores de Voto:
   - *ID_V1*= 10000001
   - *ID_V2*= 00100100
   - *ID_V3*= 00011000

   Identificador de Cargo:
   - *ID_C1* = 1000

   (El *ID_C* es único, por tratarse de un cargo específico.)

   Identificadores de Opción:
   - *ID_O1*= 00001001
   - *ID_O2*= 00010000

   (Son dos *ID_O* porque se considera un ejemplo en el que sólo hay dos candidatos para el cargo *ID_C1*)

La Tabla 3 muestra el detalle de los valores elegidos para los identificadores para un ejemplo, basado en tres votos que presenta inconvenientes:

Votante	ID_V	ID_C	ID_O
Voto 1	10000001	1000	00001001
Voto 2	00100100	1000	00010000
Voto 3	00011000	1000	00010000

Tabla 3: Identificadores

La Tabla 4 expone la representación completa de cada uno de los tres votos.

Votante	Voto Completo
Voto 1	10000001100000001001
Voto 2	00100100100000010000
Voto 3	00011000100000010000

Tabla 4: Votos Completos

Entonces, se avanza en el proceso de votación.

El primer votante deposita su sufragio, el cual aplica XOR sobre el contenido del slot. Por ser el primero, lo encuentra vacío. En consecuencia, el proceso es el que se muestra en la Tabla 5:

0000000000000000000
XOR
10000001100000001001
=
10000001100000001001

Tabla 5: Voto Número 1

A continuación, sufraga el segundo votante. El contenido inicial del slot es el voto anterior. Se produce entonces la primera colisión, por lo que se aplica la operación XOR entre el contenido del slot y el segundo voto, como se observa en la Tabla 6.

10000001100000001001
XOR
00100100100000010000
=
10100101000000011001

Tabla 6: Voto Número 2

A continuación se recepciona el tercer voto, que se almacena en el mismo slot, produciendo una nueva colisión. El resultado se muestra en la Tabla 7.

10100101000000011001
XOR
00011000100000010000
=
10111101100000001001

Tabla 7: Voto Número 3

En la Tabla 8, se muestra el detalle de los bits que quedaron en el slot luego de producirse las colisiones múltiples, en función de los identificadores.

ID_V	**ID_C**	**ID_O**
10111101	1000	00001001

Tabla 8: Detalle del voto resultante de la colisión múltiple

Puede observarse que, luego de la colisión triple el contenido del slot puede interpretarse de manera incorrecta, dado que el campo *ID_O* almacena una opción válida (00011001, ver Tabla 3), como así también el atributo *ID_C* (1000) .

Solución a los Problemas Planteados

La forma de minimizar hasta un valor deseado la probabilidad de que alguno de los problemas mencionados se produzca tiene que ver con la redundancia que se aplique en cada uno de los campos.

Suponiendo que se utilizan 16 bits para representar los *ID_V* (es decir, #V=16), y se deben representar 200 votos, que es, aproximadamente, el número de votantes en una mesa de la República, la cantidad de representaciones posibles (RP) es:

$$RP = 2^{16} = 65536 \quad (20)$$

En consecuencia, la probabilidad de:

VV="*una colisión genera un valor válido de ID_V*"

es:

$$P(VV) = 200/65536 = 0{,}0030517578125 \quad (21)$$

De la misma manera, *RP* toma los siguientes valores para otros valores de *#V*:

#V	**RP**	**P (VV)**
32	4294967296	0,000000046566128730
64	18446744073709551616	1,08420217248550e-17
128	3,402828463463e+38	5,87747175411143e-37
256	1,157920893e+77	1,72723371101888e-75

Tabla 9: Análisis de P(VV) para ID_V

Es evidente que la incorporación de redundancia reduce hasta el nivel que se desee el valor de *P(VV)*. Sin embargo, todavía puede mejorarse esa condición a través del análisis del resultado obtenido de los campos restantes:

- El *ID_C*, si bien resulta conceptualmente innecesario, aumenta la probabilidad de corregir errores que pudieran producirse. En efecto, para valores razonables de los parámetros incluidos, una proporción significativa de las colisiones involucra sólo a dos votos. En ese caso, este campo mostrará un valor de #c ceros consecutivos. Se define:

  OC = "*Una colisión da un valor de ID_C válido*"

  Es posible aceptar que:

  $$P(OC) = \frac{1}{2} \quad (22)$$

  Dado que las colisiones que involucren números pares de votos permitirán detectar el error, mientras que las que se refieran a cantidades impares ocultarán el mismo.

- El campo *ID_O* también permite aumentar el nivel de seguridad. En este caso, la probabilidad de:

  OV= "*Una colisión da un valor de ID_O válido*"

  está dada por:

  $$P(OV) = \frac{\#\ CC}{2^{\#O}} \quad (23)$$

Suponiendo 10 candidatos para el cargo, la probabilidad mencionada toma los siguientes valores (Tabla 10):

#O	P(OV)
32	0,00000000232830643653869628906 25
64	5,42101086242752217003726 4004e-19
128	2,93873587705571876992184 1345e-38
256	8,63616855509444462538635 1862e-77

Tabla 10: Análisis de P(VV) para ID_O

Se define el evento:

ERROR = "Una colisión genera un voto, en el que todos los identificadores toman valores válidos.

Luego, teniendo en cuenta que se trata de sucesos independientes, la probabilidad de interpretar mal un slot en el que se produjo una colisión es:

$$P(ERROR) = P(VV)\,P(OC)\,P(OV) \quad (24)$$

Por ejemplo, suponiendo que nueve de cada diez colisiones serán simples (es decir, entre dos votos), y que se codifica con 256 bits tanto el ID_V como el ID_O, P(error) toma el siguiente valor:

$$P(ERROR) = (\frac{200}{2^{256}})\ 0{,}9\ (\frac{10}{2^{256}}) = 1{,}3425013e-151 \quad (25)$$

Además de que el valor obtenido es extremadamente bajo, lo importante es que a través del aumento de la redundancia en el almacenamiento de los identificadores, el mismo puede llevarse a cualquier valor deseado.

Por otro lado, es importante destacar que un voto se almacena en Q canales diferentes. Definimos entonces el evento:

PÉRDIDA = "Un voto produce ERROR en todos los canales"

Entonces, la probabilidad de que el error se produzca en todos los canales es:

$$P(PÉRDIDA) = P(ERROR)^{Q} \quad (26)$$

La afirmación anterior surge por aplicación de una propiedad de los sucesos independientes. En efecto la pérdida de un mismo voto en canales diferentes no presenta relación entre sí.

También debe destacarse que *P(ERROR)* toma valores muy bajos si se utiliza la redundancia apropiada. Luego, el valor de *P(PÉRDIDA)*, será mucho menor.

Conclusiones

Se considera demostrado que la técnica de almacenamiento basada en canales paralelos de slots aumenta de manera significativa la eficiencia del almacenamiento de datos anónimos, resultando una variante muy ventajosa con respecto a la utilización de un vector único. Al mismo tiempo, permite llevar la seguridad al nivel que se desee. Los resultados empíricos indican que el comportamiento es superior en todos los aspectos elegidos.

El presente trabajo profundiza el análisis en dos aspectos:

- Proporción de colisiones múltiples.

- Elementos a considerar para llevar la probabilidad de interpretación incorrecta de votos almacenados al nivel que se desee.

Con respecto a la proporción de colisiones múltiples, los valores empíricos mostrados previamente son poco

significativos, lo cual avala la cota propuesta en la fórmula (1). La aplicación de dicha fórmula, en consecuencia, genera una cota confiable (en el sentido de que se cumple correctamente), pero que además es razonablemente cercana al valor en el que la inecuación se convierte en una igualdad.

Con respecto a la probabilidad de detectar errores que lleven a una interpretación incorrecta de un voto almacenado, el esquema propuesto permite llevar la probabilidad de error a cualquier nivel que se desee, mediante el aumento de bits de redundancia, tal como se observa en la fórmula (24), que funciona para un canal único y puede generalizarse para *Q* canales (26).

Por último, debe destacarse que la propuesta presentada, si se combina con la aplicación de una metodología "One Time Pad" garantiza el anonimato eterno del votante y da niveles computacionales de seguridad tan altos como se desee para la base de datos de sufragios.

Reconocimientos

Los autores del presente trabajo agradecen permanentemente la enorme generosidad demostrada por PhD. Jeroen van de Graaf (DCC - UFMG, Brasil), durante la estadía de diez meses del Magister Pablo García en 2012.

Referencias

[1] Uzal R., van de Graaf J., Montejano G., Riesco D., García P.: Inicio de la Línea de Investigación: Ingeniera de Software y Defensa Cibernética. Memorias del XV Workshop de Investigadores en Ciencias de la Computación 2013 (WICC 2013). Ps. 769 - 773. ISBN: 9789872817961.

[2] García P., van de Graaf J., Montejano G., Bast S., Testa O.: "Implementación de Canales Paralelos en un Protocolo Non Interactive Dining Cryptographers". 43° Jornadas Argentinas de Informática e Investigación Operativa (JAIIO 2014), Workshop de Seguridad Informática (WSegI 2014). Trabajo en aceptado para su presentación.

[3] Jakobsson M., Juels, A., Rivest, R.: "Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking"- AUSENIX Security 02. Vol. 7. Ps. 339-353. 2002.

[4] Chaum D.: "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability". Journal of Cryptology. 1988.

[5] 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.

[6] Van de Graaf J., Montejano G., García P.: "Manejo de Colisiones en un Protocolo Non Interactive Dining Cryptographers". Anales de las 42 Jornadas Argentinas de Informática e Investigación Operativa (JAIIO, ISSN: 1850-2776). Workshop de Seguridad Informática (WSegI 2013, ISSN: 2313-9110). Ps. 29 a43. Septiembre 2013.

[7] Van de Graaf J., Montejano G., García P.: 37. "Optimización de un Protocolo Non-Interactive Dining Cryptographers". Congreso Nacional de Ingeniería Informática / Sistemas de Información. CoNaIISI 2013. 21 y 22 de noviembre de 2013. Córdoba, Argentina.

[8] 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.

[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] 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.

[11] 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. 2013.

[12] García P., van de Graaf J., Hevia A., Viola A.: Beatingt he Birthday Paradox in Dining Cryptographer Networks. The Third International Conference on Cryptology and Information Security in Latin America, Latincrypt 2014. September 17- 19, 2014. Florianopolis, Brasil. Lecture Notes in Computer Science, Springer (2014).

[13] P. García, G. Montejano, J. van de Graaf and. D.Riesco, N. Debnath,and S. Bast.: Storage Optimization for Non Interactive Dining Cryptographers(nidc). In IEEE Conference Publications Editor, Conference on Information Technology - New Generations (ITNG), 2015 12th International Conference on, pages 55 – 60. IEEE, 2015.

[14] Grajski D.: Principios de Diseño Digital. Capítulo 3. Prentice Hall 1997. ISBN10: 8483220040, ISBN13: 139788483220047.