# Recuento y Recuperación de Sufragios en OTP – Vote

Pablo García[1], Silvia Bast[1], Germán Montejano [2,3]

[1] Depto. de Matemática – Facultad de Ciencias Exactas y Naturales (UNLPam)
Av. Uruguay 151- (6300) Santa Rosa - La Pampa - Argentina
{pablogarcia, silviabast}@exactas.unlpam.edu.ar
http://www.exactas.unlpam.edu.ar

[2] Depto. de Informática – Facultad de Ciencias Físico Matemáticas y Naturales (UNSL)
Ejército de los Andes 950 – (5700) San Luis – San Luis – Argentina
gmonte@unsl.edu.ar
http://www.webfcfmyn.unsl.edu.ar

[3] Depto. de Informática – Facultad de Ingeniería (UNLPam)
Calle 9 esquina 10 – (6360) General Pico – La Pampa - Argentina
german.a.montejano@gmail.com
http://www.ing.unlpam.edu.ar

**Resumen.** OTP – Vote es un modelo para votación electrónica basado en One Time Pad que utiliza claves múltiples y distribuidas que funcionan finalmente como una llave única. El esquema garantiza anonimato incondicional en la medida en que se verifiquen las condiciones iniciales exigidas y, simultáneamente, permite llevar la seguridad computacional del escrutinio a los niveles que se deseen. El presente documento presenta una técnica que permite refinar el recuento definitivo de sufragios, proveyendo al sistema de una estrategia para la recuperación de votos inicialmente perdidos como producto de la ocurrencia de colisiones.

## 1 Introducción

La implementación de sistemas de voto electrónico se encuentra en la actualidad en un momento de grandes discusiones. No son pocos los argumentos que se oponen a la existencia de productos de software diseñados para tal fin, proclamando que, bajo esa modalidad es imposible garantizar la transparencia de los resultados. Por ejemplo, en los últimos días, un importante grupo de expertos informáticos de universidades nacionales argentinas se manifestaron en contra del voto electrónico[1] e invitaron formalmente a firmar una adhesión a tal rechazo[2].

[1] http://www.cronista.com/economiapolitica/Expertos-universitarios-lanzaron-una-campana-contra-el-voto-electronico-20161101-0113.html

[2] http://www.dc.uba.ar/solicitada-voto-electronico

En particular, en el seno de este equipo de investigación no se afirma que el voto electrónico deba ser inevitablemente utilizado. Por el contrario, se considera que el esquema manual, aún con determinadas limitaciones, puede seguir prestando buenos servicios.

El razonamiento propuesto es el inverso: se considera que no es imposible generar un sistema de voto electrónico confiable. En un momento de la historia donde se realizan operaciones que involucran altos montos de dinero desde un celular, afirmar que no se puede proteger un proceso que apenas llevará diez horas (a continuación los resultados se harán públicos) parece, como mínimo, apresurado.

En ese ámbito, OTP - Vote es una propuesta, aún en desarrollo, presentada en [1], que expone varios puntos relevantes:

- Criptografía basada en One Time Pad (OTP, [2]). Tal esquema se caracteriza por cumplir con las hipótesis y condiciones del "Secreto Perfecto" de Shannon, presentadas en [3]. Esto implica que si un criptoanalista intercepta solamente criptogramas y no cuenta con información adicional, es imposible obtener el mensaje claro, aún cuando se disponga de tiempo y recursos ilimitados. En este punto es importante destacar que el período con el que contará un atacante que trate de afectar el proceso es finito. Por ejemplo, en la Argentina, se comienza a votar a las ocho horas para cerrar los comicios a las dieciocho horas.
- Utilización de claves distribuidas que finalmente se combinan para funcionar como una sola. El concepto se deriva de los protocolos propuestos por Broadbent y Tapp en [4]. En este sentido, un intento fraudulento sólo podrá tener éxito si todos los veedores involucrados (entre los que podría haber imparciales y representantes de todas las fuerzas políticas) tienen un comportamiento malicioso.
- Almacenamiento basado en canales paralelos, tal como fue presentado en [5]. En dicha publicación se enuncian una serie de fórmulas matemáticas que describen con precisión su funcionamiento.

Concretamente, el esquema de OTP - Vote es el siguiente:

1. Se implementan dos matrices de bits, que se denominarán $V$ (matriz de votos) y $K$ (matriz de claves) con dimensiones similares:
   a. La cantidad de filas ($F$) será definida por la fórmula:

$$F = Q\,S \qquad (1)$$

   Donde:

   $Q$: #Cantidad de canales paralelos que se implementan.

   $S$: #Cantidad de filas de cada canal.

   b. La cantidad de columnas $C$ estará dada por la fórmula

$$C=Id\_Voto+Id\_cargo+Id\_Candidato+Pos_1+Pos_2+....+Pos_Q \qquad (2)$$

Donde:

*Id_Voto:* es un conjunto de bits que indicará unívocamente a un sufragio específico. El conjunto válido deberá ser definido para cada acto eleccionario. Es recomendable la aplicación de una redundancia razonable para que el conjunto de Id inválidos sea alto y de esa manera aportar a la detección de errores e intentos fraudulentos. Este principio se aplica también a los dos ítems siguientes.

*Id_cargo:* Este grupo de bits se utiliza para indicar a qué cargo corresponde un voto determinado. En la opción de cada elector habrá tantos Id's como cargos diferentes deban votarse.

*Id_Candidato:* Estos bits indican la elección del votante, que elegirá uno por cada *Id_cargo.*

$Pos_i$: En cada instancia de un voto específico se indican las posiciones, en formato binario, en las que el mismo sufragio fue almacenado en todos los canales.

2. Ambas matrices se inicializan en cero.
3. Sea $a$ el total de autoridades. Cada autoridad $i$ dispondrá de dos claves aleatorias $K_{i1}$ y $K_{i2}$, de las mismas dimensiones que $V$ y $K$.
4. Se realiza, sucesivamente, el XOR de cada clave $K_{i2}$ de las autoridades con el contenido de $K$, es decir: $K=K \oplus K_{12} \oplus K_{22} \oplus ... \oplus K_{a2}$ (3)
5. Se realiza, sucesivamente, el XOR de cada clave $K_{i1}$ de las autoridades con el contenido de $V$, es decir: $V=V \oplus K_{11} \oplus K_{21} \oplus ... \oplus K_{a1}$ (4)
6. Comienzo del proceso eleccionario.
7. Para cada votante $i$ se genera una clave aleatoria $V_i$ del mismo tamaño que $V$.
8. La clave $V_i$ se utiliza en dos operaciones:
   - XOR incondicional con la clave $K$. Es decir: $K=K \oplus V_i$ (5)
   - Codificación de cada voto individual. Esta codificación consiste en los siguientes pasos:
     - Construir un voto de acuerdo a las elecciones del usuario y las posiciones aleatorias generadas.
     - Construir una matriz temporal $T$ que contiene el voto en los slots elegidos aleatoriamente en cada canal y ceros en los demás.
   - Realizar, secuencialmente, las operaciones:
     - $Vi=Vi \oplus T$ (6)
     - $V=V \oplus V_i$ (7)

Luego de realizadas las operaciones, el sistema debe asegurar que no quede ninguna constancia de la clave $V_i$ ni del estado de la matriz $T$ aplicada para cada voto. Una manera posible es usar las mismas variables en todos los casos y vaciarlas explícitamente luego del último sufragio.

9. Completada la recepción de todos los sufragios, se cierra el proceso con tres acciones en secuencia:

   - Se solicita a las autoridades que faciliten sus claves $K_{i2}$ y se aplica XOR incondicional de cada una de ellas sobre la clave $K$. Es decir:

   $$K=K \oplus K_{12} \oplus K_{22} \oplus ... \oplus K_{a2} \quad (8)$$

   - Se solicita a las autoridades que faciliten sus claves $K_{i1}$ y se aplica XOR incondicional de cada una de ellas sobre la clave $V$. O sea:

   $$V=V \oplus K_{11} \oplus K_{21} \oplus ... \oplus K_{a1} \quad (9)$$

   - Se realiza la operación $V=K \oplus V \quad (10)$

Concluido el proceso, se obtiene una matriz $V$ en la cual cada fila representará alguna de las instancias siguientes:

- Todos ceros, cuando ese slot no haya recibido ningún voto.
- Un voto válido, si en ese slot sólo cayó un único voto.
- Un valor inválido producto de una colisión de dos o más sufragios.

Llegado este punto, el presente documento propone una técnica de recuperación de información para realizar el recuento de los votos de manera correcta y eficiente. La misma puede generalizarse para su aplicación a cualquier sistema que utilice la técnica de almacenamiento por canales paralelos.

## 2 Anonimato incondicional

La característica distintiva de un sistema de E – Voting, pasa por la necesidad de mantener eternamente la privacidad de los votantes.

En ese sentido, OTP – Vote, propone dos condiciones fundamentales:

- Separación de la identificación del votante y el sufragio en sí mismo. Si esa premisa puede cumplirse, puede pensarse en esquemas que no dejen ningún rastro digital en la información almacenada. Este es un punto que los sistemas manuales manejan apropiadamente: todos los votos en papel son quemados pasado un cierto tiempo de la elección. Es imposible plantear algo similar cuando la información es almacenada en medios informáticos y manipulada por muchas personas.

- Almacenamiento auténticamente aleatorio del sufragio en cada uno de los canales paralelos. Esta idea se inspira en [6] que menciona la importancia de la aleatoriedad pero propone utilizar un vector único para guardar los sufragios. La utilización de canales paralelos presenta la implementación de Q canales replicando cada voto una vez en cada uno de ellos, en posiciones aleatorias potencialmente diferentes.

Como consecuencia de lo expuesto, las colisiones (un voto se almacena en el mismo slot que otro en un canal determinado), son posibles. Existe mucha literatura sobre cómo la utilización de canales paralelos de slots disminuye significativamente la pérdida de votos (por ejemplo, [7], [8] y [9]); sin embargo, la misma sigue teniendo probabilidad positiva.

Se propone, entonces, una técnica concreta de recuperación que permita un recuento correcto de los sufragios almacenados y que, simultáneamente, incluya técnicas de recuperación de votos que pudieran haberse perdido por la ocurrencia de colisiones.

## 3 Almacenamiento en Canales Paralelos de Slots

La técnica de almacenamiento en canales paralelos de slots se deriva de [6], que propone guardar los sufragios en un arreglo unidimensional de slots y se establece que es posible generar anonimato incondicional en la medida en que la posición que cada voto ocupa sea auténticamente aleatoria.

Tal propuesta presenta dos características fundamentales:

1. Al exigir que la posición de almacenamiento de cada sufragio sea totalmente aleatoria, es posible que dos o más de ellos coincidan en un slot determinado, generando una colisión.
2. Es necesario implementar un número muy significativo de slots para proveer niveles aceptables de seguridad de que no se perderá ningún sufragio.

OTP – Vote propone aplicar la técnica de almacenamiento basada en canales paralelos, recopilada y actualizada en [10]. En ella, se exponen fórmulas matemáticas que permiten definir los valores óptimos para los siguientes parámetros:

$T$: # Total de slots a implementar ($T \in Z^+$).
$S$: # slots en cada uno de los canales paralelos ($S \in Z^+$) $\wedge$ ($S \leq T$).
$N$: # Votantes. ($N \in Z^+$).
$Q$: # Canales paralelos a implementar. ($Q \in Z^+$).

En consecuencia, el esquema de almacenamiento es el que se muestra en la Figura 1. La propuesta es replicar cada voto una vez en cada canal, en posiciones aleatorias potencialmente diferentes. Como consecuencia, un voto sólo se perderá si colisiona en todos los canales implementados.

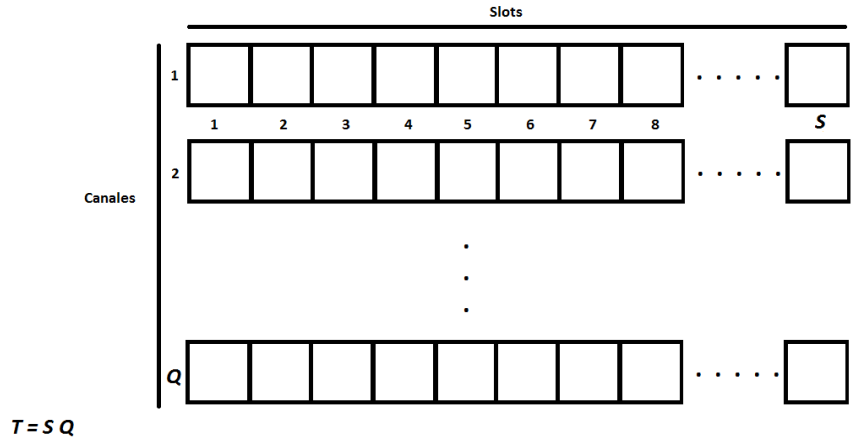


**Fig. 1.** Almacenamiento en Canales Paralelos

Más allá de lo expuesto, la probabilidad de no perder votos no puede llevarse a cero. En consecuencia, la técnica propuesta trabaja en la recuperación de datos inicialmente perdidos. Para ello, se propone incorporar dos bits adicionales a cada fila al terminar el proceso de votación. Dichos bits reciben el nombre de *Bits de Control (BC)* e inicialmente tendrán el valor 00. Tales dígitos binarios se modificarán durante el recuento de votos. Los valores que pueden tomar son:

00: La fila no ha sido revisada aún. Es el valor que se asigna a todos los *BC* de toda la matriz a modo de inicialización

01: Fila vacía. Es decir, todo el contenido de la fila son ceros. Esto sólo puede ocurrir cuando ningún voto cayó en ese slot.

10: Colisión. Se produce exclusivamente cuando dos o más votos fueron almacenados en la misma fila de la matriz. En este punto debe destacarse la importancia que cumple la redundancia en la codificación de los tres Id (voto, cargo y candidato). La existencia de múltiples codificaciones inválidas disminuye el riesgo de que una colisión, de cualquier grado, genere una codificación de un voto correcto

11: Voto válido. Esto ocurre cuando, a lo largo de todo el proceso de votación, la fila sólo recibió un sufragio.

## 4. Técnica para la Recuperación de Sufragios

Dado que no es posible evitar que la probabilidad de que se pierda uno o más votos por colisiones sea positiva, se desea generar una técnica de recuento que provea de alguna herramienta de recuperación. Simultáneamente, se busca que la misma opere de manera muy eficiente, dado que uno de los objetivos de los sistemas de voto

electrónico es proveer resultados del comicio de manera claramente más veloz que en el caso de una votación manual.

Una vez concluido el proceso de votación y realizadas las operaciones finales de XOR, se comienza con el recuento de sufragios. Se recorre la matriz de votos y se lleva a cabo una acción de acuerdo a la semántica del contenido de la fila. Como se mencionó previamente, cada fila puede almacenar todos ceros (si ningún voto cayó en ese slot), un voto válido o un conjunto de bits sin sentido producto de una colisión.

La solución es iterativa y las condiciones de corte posibles son:

1. No se encontró un voto nuevo en la presente iteración.
2. La cantidad de votos recuperados coincide con el número de votantes.

El algoritmo se describe a continuación:

```
Begin {Votación}
Votos_Recuperados=0
Nuevo_Voto=true
Repetir mientras (Votos_Recuperados<Votos_ Totales) and (Nuevo_Voto)
        Nuevo_Voto=false
        Para Cada Canal
                Para cada slot
                        Si es un voto válido:
                                Recuperar las posiciones donde se almacena ese
                                mismo sufragio en los demás canales
                                Realizar un XOR entre el voto válido obtenido y el
                                contenido de cada uno de los slots seleccionados.
                                BC=11
                                Nuevo_Voto=true
                                En caso contrario
                                        Si son todos ceros
                                                BC=01
                                        En caso contrario
                                                BC=10
                                                /* Colisión */
                Fin slot
        Fin Canal

Fin Repetir

End {Votación}
```

## 5 Un ejemplo Sencillo

El caso desarrollado a continuación elige valores deliberadamente bajos para todos los parámetros a los efectos de ilustrar el funcionamiento de la técnica de recuperación.

Se deja claro, sin embargo, la importancia de utilizar una redundancia significativa a los efectos de garantizar la seguridad del modelo.

De la misma manera, en el ejemplo se colocan algunos valores desfavorables en los parámetros, ignorando las fórmulas que deben guiar el diseño de los mismos. Esto se realiza a los efectos de que se produzcan más colisiones y así exponer el funcionamiento del modelo.

La elección – ejemplo se describe de la siguiente manera:

- Cantidad de Votantes: 4.
- Cantidad de Canales: 3.
- Cantidad de Bits para representar la Posición en cada Slot: 2.
- Cantidad de Slots por Canal: 4.
- Cantidad de Autoridades: 3.
- Cantidad de Candidatos: 3 (codificación adicional para el voto en blanco).
- Longitud del Id_Voto: 4 bits.
- Longitud del Id_Cargo: 1 bit.
- Longitud del Id_Candidato: 4 bits.
- Cantidad Total de Slots: 12.
- Longitud Total del Slot: 15 bits.
- Longitud de las Claves: 180 bits.

Para el ejemplo, los votantes eligieron las opciones expuestas en la tabla 1:

| Votante | Opción Elegida |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 3 |
| 4 | 3 |

Tabla 1: Opciones elegidas por los votantes

Los Id-Voto se muestran en la tabla 2:

| Votante | Id_Voto |
|---|---|
| 1 | 0010 |
| 2 | 1100 |
| 3 | 0011 |
| 4 | 0100 |

Tabla 2: Id-Voto correspondiente a cada elector

En el caso de los cargos, el ejemplo considera un cargo único y le otorga un solo bit. Concretamente, el único cargo se identificará con un 1. Es decir, ID_Cargo =1 para todos los votos válidos.

La tabla 3 muestra los Id-Candidato para cada postulante.

| Candidato | Id_Candidato |
|---|---|
| 1 | 0111 |
| 2 | 0110 |
| 3 | 1000 |
| Voto en Banco | 1101 |

Tabla 3: Tabla de Id_Candidato

En el presente ejemplo, las posiciones de almacenamiento de cada sufragio en cada canal se muestran en la tabla 4.

| Votante | Pos1 | Pos2 | Pos3 |
|---|---|---|---|
| 1 | 3 (11) | 1 (01) | 0 (00) |
| 2 | 3 (11) | 0 (00) | 1 (01) |
| 3 | 2 (10) | 1 (01) | 3 (11) |
| 4 | 3 (11) | 2 (10) | 3 (11) |

Tabla 4: Posición de almacenamiento de cada voto en cada canal

Como consecuencia de lo expuesto hasta ahora, la tabla 5 muestra el aspecto del sufragio de cada uno de los votantes.

| Votante | Voto Completo |
|---|---|
| 1 | 001010111110100 |
| 2 | 110010111110001 |
| 3 | 001111000100111 |
| 4 | 010011000111011 |

Tabla 5: Aspecto del sufragio de cada uno de lo votantes

En el ejemplo seleccionado, se designan tres autoridades. Obviamente, en un caso real es importante que ese número sea mayor. Incluso, cada partido político podría tener una y aumentar la transparencia del proceso.
La tabla 6 muestra el valor inicial $V$ y las claves de $K_{i1}$ de cada una de las autoridades, que operarán sobre $V$ mediante operaciones XOR. Tras las sucesivas operaciones, se obtiene el valor de $V$ con el cual se iniciará efectivamente el proceso de votación.

| $V$ | $K_{11}$ | $K_{21}$ | $K_{31}$ | $V=V \oplus K_{11} \oplus K_{21} \oplus K_{31}$ |
|---|---|---|---|---|
| 0000000000000 | 110110110000111 | 000000001001100 | 110110000010100 | 000000111011111 |
| 0000000000000 | 101100110101110 | 100110110010011 | 001110110101000 | 000100110010101 |
| 0000000000000 | 110100110001111 | 001101111111010 | 001010000011010 | 110011001101111 |
| 0000000000000 | 000011110101011 | 101110010011000 | 111111000010111 | 010010100100100 |
| 0000000000000 | 111010001011100 | 110000100100100 | 101110111101001 | 100100010010001 |
| 0000000000000 | 001011011000111 | 000111000000010 | 111110101011011 | 110010110011110 |
| 0000000000000 | 100101101100111 | 110101111001100 | 000000111011110 | 010000101110101 |
| 0000000000000 | 101000101001111 | 100110001100000 | 010011101001001 | 011101001100110 |
| 0000000000000 | 111110111011011 | 001001000111000 | 011101010110000 | 101010101010011 |
| 0000000000000 | 001101010000010 | 000101001000100 | 010011001011010 | 011011010011100 |
| 0000000000000 | 100100110111110 | 000010001011011 | 101000001011000 | 001110110111101 |
| 0000000000000 | 001111100010110 | 111101111101110 | 111110100110011 | 001100111001011 |

Tabla 6: Aplicación de las claves $K_{i1}$ de las autoridades a $V$

De la misma manera, la tabla 7 expone lo mismo pero para la matriz de claves $K$.

| $K$ | $K_{12}$ | $K_{22}$ | $K_{32}$ | $K=K \oplus K_{12} \oplus K_{22} \oplus K_{32}$ |
|---|---|---|---|---|
| 0000000000000 | 100111110101111 | 001100111001101 | 000000001001000 | 101011000101010 |
| 0000000000000 | 010000110001001 | 111110001101001 | 110111000101111 | 011001111001111 |
| 0000000000000 | 011001101010001 | 100111010110101 | 011001001110111 | 100111110010011 |
| 0000000000000 | 011111000000101 | 100100000111010 | 010001011011110 | 101010011100001 |
| 0000000000000 | 110010010110000 | 101100101100000 | 000000111101010 | 011110000111010 |
| 0000000000000 | 010000011100100 | 001100100000101 | 100011100010010 | 111111011110011 |
| 0000000000000 | 111001110010110 | 001111000000100 | 000100011111111 | 110010101101101 |
| 0000000000000 | 001100000110101 | 011101010001101 | 111001001000010 | 101000011111010 |
| 0000000000000 | 110011010100011 | 100000000110101 | 001100000000100 | 011111010010010 |
| 0000000000000 | 000100001100101 | 100000101011001 | 001011000101000 | 101111100010100 |
| 0000000000000 | 111110000000010 | 100110010100110 | 101011011001101 | 110011001101001 |
| 0000000000000 | 110010101101101 | 111110111111010 | 110110110111101 | 111010100101010 |

Tabla 7: Aplicación de las claves $K_{i2}$ de las autoridades a $K$

Completada la etapa anterior, comienza el proceso específico de votación. Como se explicó previamente, para cada sufragio se genera una clave aleatoria $V_i$ de tipo OTP, es decir de la misma medida que $V$ y $K$. Cada voto se lleva a cabo de la manera descripta previamente.

Al finalizar el proceso, $V$ y $K$ contienen los resultados encriptados. De manera simétrica se aplican los XOR de las claves de las autoridades (tablas 8 y 9).

| $V$ | $K_{11}$ | $K_{21}$ | $K_{31}$ | $V=V \oplus K_{11} \oplus K_{21} \oplus K_{31}$ |
|---|---|---|---|---|
| 000000111011111 | 110110110000111 | 000000001001100 | 110110000010100 | 011011111000011 |
| 000100110010101 | 101100110101110 | 100110110010011 | 001110110101000 | 110000111001001 |
| 110011001101111 | 110100110001111 | 001101111111010 | 001010000011010 | 000000011111101 |
| 010010100100100 | 000011110101011 | 101110010011000 | 111111000010111 | 001011000000011 |
| 100100010010001 | 111010001011100 | 110000100100100 | 101110111101001 | 110101010110110 |
| 110010110011110 | 001011011000111 | 000111000000010 | 111110101011011 | 101100101111000 |
| 010000101110101 | 100101101100111 | 110101111001100 | 000000111011110 | 110001101010101 |
| 011101001100110 | 101000101001111 | 100110001100000 | 010011101001001 | 101011110001011 |
| 101010101010011 | 111110111011011 | 001001000111000 | 011101010110000 | 111110001110101 |
| 011011010011100 | 001101010000010 | 000101001000100 | 010011001011010 | 111000011110000 |
| 001110110111101 | 100100110111110 | 000010001011011 | 101000001011000 | 100011111111101 |
| 001100111001011 | 001111100010110 | 111101111101110 | 111110100110011 | 101000011011000 |

Tabla 8: Aplicación final de las claves $K_{i1}$ de las autoridades a $V$

| $K$ | $K_{12}$ | $K_{22}$ | $K_{32}$ | $K=K \oplus K_{12} \oplus K_{22} \oplus K_{32}$ |
|---|---|---|---|---|
| 101011000101010 | 100111110101111 | 001100111001101 | 000000001001000 | 011011111000011 |
| 011001111001111 | 010000110001001 | 111110001101001 | 110111000101111 | 110000111001001 |
| 100111110010011 | 011001101010001 | 100111010110101 | 011001001110111 | 001111011011010 |
| 101010011100001 | 011111000000101 | 100100000111010 | 010001011011110 | 100000000111101 |
| 011110000111010 | 110010010110000 | 101100101100000 | 000000111101010 | 000111101000111 |
| 111111011110011 | 010000011100100 | 001100100000101 | 100011100010010 | 101001010101011 |
| 110010101101101 | 111001110010110 | 001111000000100 | 000100011111111 | 100010101101110 |
| 101000011111010 | 001100000110101 | 011101010001101 | 111001001000010 | 101011110001011 |
| 011111010010010 | 110011010100011 | 100000000110101 | 001100000000100 | 110100110000001 |
| 101111100010100 | 000100001100101 | 100000101011001 | 001011000101000 | 001010100000001 |
| 110011001101001 | 111110000000010 | 100110010100110 | 101011011001101 | 100011111111101 |
| 111010100101010 | 110010101101101 | 111110111111010 | 110110110111101 | 110100011000100 |

Tabla 9: Aplicación final de las claves $K_{i2}$ de las autoridades a $K$

Terminado el proceso de votación se realiza el XOR entre *V* y *K*, obteniendo la matriz de votos definitiva, que se muestra en la tabla 10.

| ***V*** | ***K*** | ***V=V⊕K*** |
|---|---|---|
| 011011111000011 | 011011111000011 | 000000000000000 |
| 110000111001001 | 110000111001001 | 000000000000000 |
| 000000011111101 | 001111011011010 | 001111000100111 |
| 001011000000011 | 100000000111101 | 101011000111110 |
| 110101010110110 | 000111101000111 | 110010111110001 |
| 101100101111000 | 101001010101011 | 000101111010011 |
| 110001101010101 | 100010101101110 | 010011000111011 |
| 101011110001011 | 101011110001011 | 000000000000000 |
| 111110001110101 | 110100110000001 | 001010111110100 |
| 111000011110000 | 001010100000001 | 110010111110001 |
| 100011111111101 | 100011111111101 | 000000000000000 |
| 101000011011000 | 110100011000100 | 011100000011100 |

Tabla 10: *V=V⊕K*

Esta última versión de *V* es la que contiene los resultados descriptados de la votación. La tabla 11 los elementos semánticos componentes de cada fila de *V*.

| ***Id_Voto*** | ***Id_Cargo*** | ***Id_Candidato*** | ***Pos1*** | ***Pos2*** | ***Pos3*** |
|---|---|---|---|---|---|
| 0000 | 0 | 0000 | 00 | 00 | 00 |
| 0000 | 0 | 0000 | 00 | 00 | 00 |
| 0011 | 1 | 1000 | 10 | 01 | 11 |
| 1010 | 1 | 1000 | 11 | 11 | 10 |
| 1100 | 1 | 0111 | 11 | 00 | 01 |
| 0001 | 0 | 1111 | 01 | 00 | 11 |
| 0100 | 1 | 1000 | 11 | 10 | 11 |
| 0000 | 0 | 0000 | 00 | 00 | 00 |
| 0010 | 1 | 0111 | 11 | 01 | 00 |
| 1100 | 1 | 0111 | 11 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 |
| 0111 | 0 | 0000 | 01 | 11 | 00 |

Tabla 11: Aspecto de la matriz *V* antes de comenzar el recuento

Según la técnica propuesta se incorporan los bits de revisión. Inicialmente, el valor de ambos bits será 00, que indica que aún los slots no fueron inspeccionados. Al concluir la primera pasada de control, la matriz tendrá el aspecto presentado en la Tabla 12:

| ***Id_Voto*** | ***Id_Cargo*** | ***Id_Candidato*** | ***Pos1*** | ***Pos2*** | ***Pos3*** | ***BC*** |
|---|---|---|---|---|---|---|
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0011 | 1 | 1000 | 10 | 01 | 11 | 11 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 10 |
| 1100 | 1 | 0111 | 11 | 00 | 01 | 11 |
| 0010 | 1 | 0111 | 11 | 01 | 00 | 11 |
| 0100 | 1 | 1000 | 11 | 10 | 11 | 11 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |

Tabla 12: Aspecto de la matriz *V* al finalizar la primera iteración del recuento

Finalmente, al terminar el control, el aspecto de la matriz es el que se observa en la tabla 13.

| ***Id_Voto*** | ***Id_Cargo*** | ***Id_Candidato*** | ***Pos1*** | ***Pos2*** | ***Pos3*** | ***BC*** |
|---|---|---|---|---|---|---|
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0011 | 1 | 1000 | 10 | 01 | 11 | 11 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 1100 | 1 | 0111 | 11 | 00 | 01 | 11 |
| 0010 | 1 | 0111 | 11 | 01 | 00 | 11 |
| 0100 | 1 | 1000 | 11 | 10 | 11 | 11 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |
| 0000 | 0 | 0000 | 00 | 00 | 00 | 01 |

Tabla 13: Aspecto de la matriz *V* al finalizar el recuento

De donde pueden obtenerse los votos en formato original, que se identifican por tener *BC*=11. Se observa que el resultado de la elección es correcto (dos votos para el candidato 1 y dos para el candidato 3) y que no se perdió ningún sufragio. Para el caso seleccionado, la totalidad de los votos habían sido recuperados en la primera iteración; la segunda sólo actualizó valores en los bits de control.

El ejemplo expuesto es sólo a los efectos de mostrar el funcionamiento del modelo. Se implementó un simulador que permite realizar votaciones de mayor porte y, en todos los casos los resultados son satisfactorios si se cumplen dos condiciones básicas:

1. La selección de los parámetros *S* y *Q* se realiza aplicando las fórmulas enunciadas en [10].
2. Se aplica redundancia significativa para la codificación de *Id- Voto, Id-Candidato y Id-Cargo*. En [1] se exponen especificaciones muy precisas en ese sentido.

## 6 Conclusiones y Problemas Abiertos

Se expusieron argumentos por los que se considera que construir un sistema de voto electrónico confiable no es imposible. La mayoría de los puntos a considerar en ese sentido son similares a los de muchas transacciones que se llevan a cabo electrónicamente sin inconvenientes. Se considera que el aspecto diferencial de los sistemas de E - Voting (es decir, el anonimato incondicional) no es una meta imposible.

Como consecuencia de lo anterior, se trabaja en el desarrollo de un producto que permita una votación segura, que se denomina OTP - Vote. Dentro de ese proyecto, el presente documento se encarga de presentar una técnica de recuento confiable con recuperación de votos perdidos por colisión. La misma presenta varias características destacables:

- Por aplicación de almacenamiento basado en canales paralelos, la probabilidad de que no exista al menos una réplica válida de un voto dado se puede llevar hasta cualquier nivel exigido con un nivel de eficiencia mayor que si se aplica un único canal unidimensional.
- En el caso de que exista algún voto que colisionó en todos los canales, la aplicación de las técnicas descriptas permite asegurar que la probabilidad de que ese voto no pueda ser recuperado puede llevarse a niveles despreciables.
- La probabilidad de que una colisión genere una representación válida de sufragio, también puede llevarse a los niveles deseados mediante la aplicación de la redundancia apropiada.
- La técnica resulta particularmente ágil por tratarse de operaciones XOR, que se pueden implementar con una equivalencia directa con operaciones de bajo nivel.

De la misma manera, OTP - Vote presenta algunas cuestiones que deben ser resolverse a futuro:

- La forma exacta en que se implementarán técnicas de verificabilidad “End to End”. En efecto, es necesario proveer al esquema propuesto de estrategias que permitan que cada votante pueda controlar que su voto fue tenido en cuenta correctamente y que cualquier persona pueda verificar la integridad del acto eleccionario. Una de las principales dificultades para que el voto electrónico sea aceptado pasa, precisamente, por poder demostrarle a la sociedad que los resultados obtenidos son correctos y, simultáneamente, que la privacidad de los votantes será protegida eternamente.
- La selección de un método criptográfico (por ejemplo, El Gamal, RSA, etc.) para las transmisiones entre las estaciones de votación y los servidores, en el caso de que se aplique tal metodología. Este punto reforzará la seguridad de la propuesta, proveyendo al mismo tiempo de constancias de integridad.
- El refinamiento de protocolos antifraude para la garantía de que el comicio no pueda ser arruinado por maniobras deliberadas o por accidente. Se trabaja sobre antecedentes desarrollados previamente por este equipo de trabajo (por ejemplo, [10]), que se basan en commitments de Pedersen y logaritmos discretos y que presentan una mejora sustancial con respecto a propuestas anteriores basadas en Bit Commitments con XOR (BCX).
- La revisión cuidadosa de la técnica propuesta en busca de variantes que aumenten la eficiencia en la utilización del almacenamiento. Se desea establecer si es posible mejorar el nivel de recuperación con el mismo nivel de almacenamiento, o incluso más bajo.

## Referencias


1. **Bast, S.:** "Optimización de la Integridad de Datos en Sistemas de E-Voting". Tesis para obtener el grado de Magister en Ingeniería de Software de la Facultad de Ciencias Físico Matemáticas y Naturales de la Universidad Nacional de San Luis. Directores: Dr. Germán Montejano y Mg. Pablo García. Defendida el día 14/12/2016.

2. **Nagaraj N., Vaidya V., Vaidya P:** "Revisiting the one-time pad," International Journal of Network Security, vol. 6, no. 1, pp. 94-102, 2008.

3. **Shannon, C.:** "Communication Theory of Secrecy Systems". Bell System Technical Journal 28 (1949) 656–715.

4. **Broadbent A., Tapp A.:** "Information-Theoretically Secure Voting Without an Honest Majority". In Proceedings of the IAVoSS Workshop On Trustworthy Elections (WOTE 2008).

5. **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). Disponible en: http://sedici.unlp.edu.ar/handle/10915/42066.

6. **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.

7. **García, P., van de Graaf J., Hevia A., Viola A.:** "Beating the Birthday Paradox in Dining Cryptographers Networks". En "Progress in Cryptology – Latincrypt 2014". Springer International Publishing. ISSN: 0302-9743. ISSN (electrónico): 1611-3349. ISBN: 978-3-319-16294-2. ISBN (eBook): 978-3-319-16295-9. Ps. 179 – 198. Octubre, 2014.

8. **García P., van de Graaf J., Montejano G., Riesco D., Debnath N., Bast S.:** "Storage Optimization for Non-Interactive Dining Cryptographers (NIDC)". The International Conference on Information Technology: New Generations. 2015. Las Vegas, Nevada, USA. Disponible en: http://ieeexplore.ieee.org/document/7113449/

9. **García P., Bast S., Fritz E., Montejano G., Riesco D., Debnath N.,** "A Systematic Method for Choosing Optimal Parameters for Storage in Parallel Channels of Slots". IEEE International Conference on Industrial Technology (ICIT 2016). 14 - 17 March 2016 / Taiwan, Taipei. En: http://ieeexplore.ieee.org/document/7475019/.

10. **García, P.:** "Una Optimización para el Protocolo Non - Interactive Dining Cryptographers". ISBN-13: 978-3-639-85270-7. ISBN-10: 3639852702. EAN: 9783639852707. Idioma: Español. Editorial Académica Española (https://www.eae-publishing.com/). 2017.