Un Método de Detección de Integridad de una Urna Digital en Grandes Elecciones

Roger Jardí Cedó, Xavier Taixés Ventosa y Jordi Castellà Roca
Universitat Rovira i Virgili, Departament d'Enginyeria Informàtica i Matemàtiques,
UNESCO Chair in Data Privacy, Av. Països Catalans 26. E-43007, Tarragona, Spain
Email: {roger.jardi, xavier.taixes, jordi.castella}@urv.cat

***Resumen*—Este trabajo presenta un método para detectar la eliminación de votos de la urna digital de una votación electrónica. Más concretamente, se pretende detectar posibles adiciones, modificaciones o supresiones de votos ya emitidos mediante la inserción de *dummy votes*. Este método toma mayor importancia en sistemas de votación electrónica diseñados para grandes elecciones, donde existe un gran número de partidos y votantes, y es más difícil el control de todos los votos.**

I. Introdución

Las elecciones son unos de los pilares de una democracia. Mediante una votación, la sociedad puede elegir a sus representantes, tomar una decisión o expresar una opinión. Por este motivo, alguien podría estar interesado en manipular los resultados electorales sin ser descubierto. Para detectar y evitar cualquier tipo de alteración en el proceso de votación son necesarios mecanismos de control y de seguridad.

En 1964 se introdujo en EE. UU. el primer sistema de voto electrónico. Respecto a los sistemas tradicionales, la evolución de estos sistemas ha supuesto una mejoría en varios aspectos como es el caso de la eficiencia en el escrutinio, la solución de problemas sujetos a errores humanos o incluso, la accesibilidad para personas discapacitadas o analfabetas. No obstante, estos cambios han hecho replantear un conjunto de problemáticas, ya conocidas en los sistemas antiguos, relacionadas con la seguridad. Las más importantes ([Ros10], [AJCCR10]) son el secreto de voto, el anonimato del votante, la verificación del proceso o la integridad de la urna.

Este trabajo se centra en el estudio de un método para detectar la eliminación de votos de la urna digital (integridad) en una votación electrónica.

Estructura del documento. El siguiente apartado I-A introduce el estado del arte. En la Sección II se propone un método para verificar la integridad de la urna digital mediante *dummy votes*. En la Sección III se describe como adaptar este método a un sistema de voto basado en una papeleta de voto con estructura vectorial. Finalmente, en la Sección IV se presentan las conclusiones del trabajo.

I-A. Estado del arte

En la urna digital se registran los votos provenientes de los votantes. Ésta debe permanecer intacta desde que los votos son emitidos por los votantes hasta el final de las elecciones. Es decir, para que una urna digital sea íntegra no puede haber:

- *modificación* de votos existentes en el registro;
- *adición* de votos falsos;
- *supresión* de votos existentes en el registro.

Mediante la *firma digital* de cada voto por parte de cada votante se puede detectar la modificación de los votos y su adición (incluso permite detectar la adición de más de un voto por un mismo votante).

La detección en supresión de votos puede resultar un trabajo más complejo. Los métodos más utilizados en los sistemas de votación electrónica actuales [AJCCR10] son

- *Sistema de auditoría* con logs entrelazados (*immutable logs*) [SDW08], [Nor09a], [Nor09b].
- A través de la *verificación individual* de los votantes [MN06], [SDW08], [Nor09a].
- Mediante el *registro paralelo* que realizan algunos sistemas de votación [Mer00], [BMQR07].

En el primer método, cuando un voto llega a la urna se registra secuencialmente junto al resto de votos que han llegado anteriormente. La cadena creada permite detectar cualquier eliminación. No obstante, el sistema de auditoría no es invulnerable. Un voto podría ser eliminado durante el periodo de tiempo comprendido entre la llegada del voto al sistema y el registro del mismo. En este caso, el sistema de auditoría no llegaría a detectar esta eliminación.

El segundo mecanismo se basa en la propiedad de verificación individual que tienen muchos sistemas de votación y que permite comprobar a cada votante que su voto ha sido tenido en cuenta en el escrutinio. De este modo, en caso de haber algún voto eliminado el votante lo podría detectar. Sin embargo, es difícil que todos los votantes verifiquen su voto y por lo tanto, el grado de detección de la integridad de la urna depende del número de votantes que decidan verificar su voto ([Har09], [SCC$^+$10]). Por este motivo, la posibilidad de detectar un voto eliminado puede verse reducida.

El tercer mecanismo almacena los votos en dos registros diferentes, normalmente en formatos distintos (en papel y digitalmente). Para verificar la integridad de la urna hace falta comparar ámbos registros. A pesar de ello, podría darse el caso de eliminación de un mismo voto en ámbos registros. Además, el esfuerzo para realizar la comprobación es costoso y a veces susceptible a errores [AJCCR10].

Además de estos problemas, la complejidad para garantizar la integridad de la urna puede verse incrementada a medida que aumenta la cantidad de votos y sobretodo, en caso que el entorno sea desconfiable.

Dado que los sistemas anteriores tienen todos alguna carencia en seguridad es interesante combinar varios métodos [SDW08], [Nor09a], [Nor09b] y estudiar nuevas técnicas que, aplicadas conjuntamente, aporten mejoras al respecto.

II. Integridad de la urna digital: Dummy Votes

En este artículo se propone un método para detectar alteraciones en la integridad de la urna digital mediante la inserción de *dummy votes* sin que estos afecten a los resultados electorales.

La existencia de *dummy votes* en la urna digital permite comprobar, una vez finalizada la votación, que éstos no han sido eliminados. Si se da el caso, se puede afirmar con una cierta probabilidad que por extensión, ningún voto correcto ha sido eliminado. No obstante, el atacante no debe poder diferenciar los *dummy votes* ya que podría eliminar los votos correctos y dejar los *dummy*. Por este motivo, los *dummy votes* deberán ser *indistinguibles* de los votos válidos durante el proceso electoral e *identificables*, solamente, una vez finalizado el proceso de votación.

La *indistinguibilidad* de los *dummy votes* durante el proceso electoral evitará que el atacante sepa cúales son *dummy* y cuáles son reales. De este modo, si decide eliminar alguno, en una cierta probabilidad, estará eliminando un *dummy vote* y por consiguiente, será detectado. La *identificabilidad* de los *dummy votes* permitirá verificar que todos los *dummy votes* están en la urna una vez finalizada la votación. Además, permitirá separar los *dummy votes* de la urna y así, hacer el recuento de forma correcta.

La *probabilidad* de *detectar* algún cambio en la integridad de la urna digital depende directamente del número de *dummy votes* y del tamaño del electorado. Cuanto mayor sea el porcentaje de *dummy votes* en función del número de votos reales emitidos, mayor será la probabilidad de detección. El grado de detección se estudia en la sección II-B.

No obstante, la *seguridad* de este mecanismo se basa también en la correcta construcción de los *dummy votes* y su inserción en la urna, ya que de lo contrario, la detección de posibles modificaciones en la urna no tendría sentido. Además, la comprobación de la integridad de la urna debe realizarse adecuadamente y de manera pública.

II-A. Descripción del método

Las fases más importantes de este método son cuatro, (i) *generación*, (ii) *publicación*, (iii) *inserción* y, (iv) *apertura y verificación* de los *dummy votes*. En los siguientes apartados se detalla en que consisten estos pasos.

II-A1. Generación de los Dummy Votes: Para que un *dummy vote* sea *indistinguible* deberá tener la misma apariencia que un voto válido. Por lo tanto, deberá estar construido de la misma forma que este. No obstante, el *dummy vote* deberá tener alguna diferencia que permita identificarlo una vez finalizado el periodo de votación. Los votos válidos son emitidos por votantes reales, en cambio, los *dummy votes* han sido creados por la misma autoridad electoral. Esta autoridad, encargada de cumplir las funciones de organización y vigilancia de las elecciones, dispone de un conjunto de claves de votación generadas por un esquema umbral (p.ej. [Sha79]).

Ambos tipos de votos son firmados digitalmente para demostrar su autenticidad mediante firmas certificadas por una misma autoridad de certificación. Mientras que los firmantes de los votos válidos son votantes reales, los firmantes de los *dummy votes* son votantes llamados de auditoría.

Esta misma entidad genera *d - dummy votes* para poner a prueba la urna digital. El contenido de cada voto, es decir, la opción de voto seleccionada en cada *dummy vote* será determinada por esta entidad de forma secreta. La cantidad total de *dummy votes* también debe ser secreta para mantener su *indistinguibilidad* con los votos válidos. Como se verá en la Sección II-B, la cantidad de *dummy votes* afecta directamente a la probabilidad de detección de integridad de la urna digital.

II-A2. Publicación de los Dummy Votes: Una vez realizado el proceso de generación y previo al inicio de las elecciones, los *dummy votes* deben ser registrados de manera *inidentificable* por la entidad de confianza hasta finalizar el proceso electoral para dejar constancia de cuáles son y poder ser separados a posteriori. Es decir, se debe hacer un compromiso de cuales son los *dummy votes* sin desvelar (temporalmente) su apariencia, contenido, o cantidad. De lo contrario, el atacante podría eliminar votos correctos sin ser descubierto.

Para ello son cifrados en bloque en un sobre digital y publicados en una *bulletin board* (BB) por la misma autoridad electoral. De este modo, hasta el proceso de apertura (§II-A4), no se conocerán. Además, la firma digital garantiza la autenticidad de los *dummy votes*, es decir, los *dummy votes* publicados son los mismos que los generados por la entidad de confianza.

II-A3. Inserción de los Dummy Votes: Con el proceso electoral abierto, los *dummy votes* son *insertados* en la urna electoral de manera que ni el momento ni el modo de inserción permitan *distinguir* los *dummy votes* del resto. Es decir, la inserción de los *dummy votes* deberá mantener su *indistiguibilidad*. Por este motivo, debe realizarse progresivamente, paralela a los votos válidos y con una proporción adecuada.

II-A4. Apertura y verificación de los Dummy Votes: Una vez finalizadas las elecciones, cualquier persona puede verificar que todos los *dummy votes* se han mantenido *íntegros* en la urna digital gracias a su *identificabilidad*, es decir, que los votos *dummy* de la urna son los mismos que los generados y posteriormente publicados en la *bulletin board*.

Para ello, se hace pública la urna digital con todos los votos cifrados recibidos, incluyendo los *dummy* y se abre el sobre digital situado en la *bulletin board* que contiene los *dummy votes*. A partir de este momento, cualquiera puede *identificar* cada uno de los *dummy votes* dentro de la urna digital mediante la comparación de cada uno de los *dummy votes* situados en la *bulletin board* con los de la urna.

Además de la apertura del sobre digital también se abren los votos de la *bulletin board* permitiendo así la verificación de su correcta generación. La apertura de los *dummy votes* significa el conocimiento universal del contenido de los *dummy*.

Para obtener los resultados, se realiza el recuento con los *dummy votes* dentro de la urna, y posteriormente, se restan.

La existencia de todos los *dummy votes* en la urna digital evidencía con una cierta probabilidad la *integridad de toda la urna* ya que, si alguien hubiera intentado alterarla, con una determinada probabilidad modificaría alguno de ellos, y por lo tanto, se detectaría en la verificación de los *dummy votes*.

II-B. Análisis de la probabilidad de detección de integridad

El sistema presentado proporciona un nivel de detección de votos eliminados que depende del número de *dummy votes* insertados en la urna. En caso de que algún *dummy vote* sea eliminado, se detectará el fraude. Así, cuantos más *dummy votes* existan en la urna o cuantos más votos se eliminen, más alta será la probabilidad de detección. En este apartado se estudia el comportamiento de la probabilidad de detección en función de la variabilidad de la cantidad de *dummy votes* y/o votos eliminados.

En unas elecciones, sea n el número de votos emitidos. Supongamos que añadimos m *dummy votes* aparentemente indistinguibles de los votos válidos. La probabilidad de eliminar un voto que no sea un *dummy vote*, es

$$\frac{n}{n+m}$$

Dado que se trata de un muestreo sin reposición, podemos deducir que la probabilidad de eliminar d votos de tal forma que ninguno de ellos sea un *dummy vote*, entonces es

$$\frac{n!(n+m-d)!}{(n-d)!(n+m)!}$$

De este modo, la seguridad que nos da este sistema es el caso complementario: la probabilidad p de detectar que un atacante elimine uno o más *dummy votes*:

$$p = 1 - \frac{n!(n+m-d)!}{(n-d)!(n+m)!}$$

Si expresamos los *dummy votes* y los votos eliminados en función de n, de forma que $m = n \cdot r$ y $d = n \cdot s$, se puede ver que para valores suficientemente grandes de n, esta expresión se aproxima con

$$p = 1 - \frac{n!(n+m-d)!}{(n-d)!(n+m)!} \approx 1 - e^{-nrs} \qquad (1)$$

El cuadro I muestra unas aproximaciones de la probabilidad de detección en función de la cantidad de *dummy votes* y el porcentaje de votos eliminados respecto al censo.

dummy votes	s	% eliminados	p
100.000	10^{-4}	0.01 %	0,99995
100.000	10^{-5}	0.001 %	0,63
500.000	10^{-5}	0.001 %	0,993
1.000.000	10^{-5}	0.001 %	0,99995

Cuadro I
APROXIMACIÓN DE LA PROBABILIDAD DE DETECCIÓN EN FUNCIÓN DE LA CANTIDAD DE *dummy votes* Y EL PORCENTAJE DE VOTOS ELIMINADOS RESPECTO AL CENSO

Como se puede comprobar en el primer y último ejemplo de la tabla, si multiplicamos por 10 el número de *dummy votes*, reducimos a una décima parte el número de votos que se podrían eliminar sin detección, manteniendo p invariante. Esto se debe a que la expresión 1 mantiene la relación entre r y s.

Dado un valor $\epsilon \in (0, 1)$, si se desea que la probabilidad de no detección de la eliminación de la razón s entre el número de eliminados y el censo sea inferior a ϵ, podemos usar la aproximación de la expresión 1 para deducir que el número de *dummy votes* m necesarios tiene que ser, como mínimo,

$$m > \frac{-\ln(\epsilon)}{s}$$

Por ejemplo, si se quiere detectar con una probabilidad del $0{,}99999$ una eliminación de más de un $0{,}001\,\%$ de votos ($s = 0{,}00001$), el mínimo de *dummy votes* que deberán ser insertados será:

$$m > \frac{-\ln(0{,}00001)}{0{,}000001} = 1151292{,}5 \qquad (2)$$

Por consiguiente, se puede determinar la cantidad de *dummy votes* necesaria según el nivel de detección que se desee para un sistema de votación electrónica. Además, cabe señalar que es probable que el nivel de detección en un caso real sea mayor. En caso que la participación no llegara al 100 %, la probabilidad de detección sería superior a la calculada, ya que en este caso, en vez de depender del censo, ésta depende de la relación entre el número de *dummy votes* y los votantes. A pesar de esto, los cálculos se han realizado sobre el censo, ya que el número de *dummy votes* tiene que ser fijado antes de las elecciones.

III. ADAPTACIÓN DE LOS DUMMY VOTES A VECTOR BALLOT

Martin Hirt en 2001 ([Hir01],§5.5.2) introdujo en su Tesis Doctoral un voto con estructura vectorial. Aggelos Kiayias y Moti Yung en 2004 y 2010 [KY10], [KY04] desarrollan esta idea para diseñar un sistema eficiente de votación llamado *VectorBallot*. Este representa cada voto utilizando un vector de c posiciones, tantas como opciones de voto sean requeridas. Cada posición del vector es el criptograma del cifrado de un 0 o un 1 con un criptosistema homomórfico aditivo (p.ej. Paillier [Pai99]) según si se quiere votar por esta opción de voto o no. La agregación de los resultados se hace por columnas, es decir, componente a componente, y finalmente se descifran tantos valores agregados como opciones de voto, resolviendo así el problema de la limitación del tamaño de las elecciones.

La seguridad del sistema se basa en la correcta construcción de los votos. Para ello utiliza un conjunto de pruebas de conocimiento nulo (Zero Knowledge Proof o ZKP) [GMW87] no interactivas con el fin de verificar que el valor cifrado de cada posición del vector sea un "0" o un "1", y que no exista más de un 1 cifrado en todo el vector, es decir, que el votante no haya votado a más de una opción de voto. La sección III-A describe en más detalle la construcción del voto, las ZKPs utilizadas y el tamaño de un voto.

Además, el sistema permite votos *write-in* (votar por candidatos que no aparecen en la lista electoral) mediante la combinación de técnicas de *mixing* y técnicas *homomórficas* bajo

una misma interfaz de usuario. En este caso, la construcción del voto es modificada añadiendo 2 posiciones más al vector, una para indicar con un *flag* si el voto es *write-in* o no y la otra para el voto *write-in* cifrado. El escrutinio de estos votos no se realiza de forma homomórfica, sino como si se tratara de un sistema basado en *mixing* únicamente. Los votos *write-in* son separados de los otros y anonimizados mediante técnicas de *mixing*. Luego son descifrados y contados uno a uno.

Las ventajas de esta nueva forma de realizar la construcción del voto son notables. Aparte de permitir el voto *write-in*, la eficiencia en el recuento es mayor que los sistemas homomórficos tradicionales. Otra ventaja es la *gran cantidad de votos y partidos* que pueden representarse y agregarse. El tamaño de la clave pone un límite al número de votos muy elevado y el número de opciones de voto no está limitado ya que siempre se puede añadir una componente más.

No obstante, en términos de eficiencia y desde el punto de vista de la emisión del voto, requiere un mayor coste computacional y de comunicación. Es necesario probar la corrección del voto. Al contrario de otros sistemas homomórficos que sólo requieren una ZKP, en este caso, el número de ZKPs es $c+1$, tantas como elementos del vector más una. Las ZKPs son adjuntadas con el voto para una posterior verificación. Esto aumenta notablemente el tamaño de memoria que ocupa un voto en la urna.

III-A. Formato y tamaño de un VectorBallot

Siguiendo la notación de [KY10], sea $\mathcal{E}(x)$ una función de cifrado del mensaje x. Sea c el número de opciones de voto existentes, y sea $C = (C_1, C_2, \ldots, C_c)$ el vector cuya posición C_i representa la i-ésima opción de voto. Así, $C_i = \mathcal{E}(1)$ si se vota por la opción en posición i, o $C_i = \mathcal{E}(0)$ en caso contrario. Sea $Q^{m,V}$ el predicado definido como $Q^{m,V} = 1$ si y sólo si $V = \mathcal{E}(m)$, y $Q^{m,V} = 0$ en caso contrario.

Cada voto estará formado por el vector C y un conjunto de ZKPs. Cada posición del vector C_i sólo puede contener un "0" o un "1", con lo que, para demostrar que la componente i del voto está bien construida sin desvelar el contenido del voto, se debe adjuntar una ZKP $Q^{0\vee 1, C_i}$ correspondiente. Esta ZKP sirve para demostrar que un valor cifrado corresponde con uno entre dos valores, en este caso el "0" y el "1". En [CGS97], §3.2 se describe cómo construir este tipo de ZKPs.

Además, para demostrar que el votante ha votado por una única opción o en blanco, es decir, que no existe más de un "1" cifrado en todo el vector, se calcula la agregación de todas las posiciones del vector $C_{\text{agg}} = C_1 + C_2 + \ldots + C_c$ utilizando las propiedades homomórficas aditivas que proporciona el criptosistema. Entonces se comprueba que este valor C_{agg} es también un "0" o un "1" con $Q^{0\vee 1, C_{\text{agg}}}$.

Como el conjunto de ZKPs es una combinación de ANDs,

$$(\wedge_i Q^{0\vee 1, C_i}) \wedge Q^{0\vee 1, C_{\text{agg}}} \ ,$$

simplemente se deben comprobar todas y cada una de las ZKP individualmente [Hir01],§5.2.3.2.

Suponiendo que utilizamos el criptosistema de ElGamal con una clave de 2048 bits, el criptograma representado en cada posición del vector ocupará 512 bytes ($2 \cdot 2048$ bits). El tamaño del vector viene determinado por la siguiente expresión:

$$C_{size} = |C| = 512 \cdot c \text{ bytes} \tag{3}$$

El tamaño de cada prueba de conocimiento nulo que permite comprobar la consistencia de cada posición i del vector es la siguiente [Bra05]:

$$|zkp_i| = 8 \cdot 512 \text{ bytes} = 4 \text{ KB}$$

El tamaño que ocupan todas la ZKPs para verificar la correcta construcción del vector, suponiendo que no se permita votos *write-in*, viene determinado por c ZKPs, que verifican la consistencia de cada posición del vector, más otra ZKP que permite verificar que el votante ha votado en blanco o por una sola opción de voto:

$$|zkp| = 4(c+1) \text{ KB} \tag{4}$$

Además, cada voto V lleva la firma F del votante. Por lo tanto habrá que sumarle el tamaño que ocupa esta firma. Suponiendo una firma RSA estándar, esto suponen 2048 bits.

Teniendo en cuenta las expresiones 3 y 4, y el tamaño de la firma digital, se obtiene que el tamaño total de un voto es el siguiente:

$$|V| = |C| + |zkp| + |F| = 4608c + 4352 \text{ bytes} \tag{5}$$

Si comparamos el tamaño requerido para almacenar cada voto con el espacio que pueda ocupar un voto homomórfico, éste es mucho superior. No obstante, las consecuencias del tamaño del voto hay que analizarlas en el caso de unas elecciones en las que la cantidad de votos sea elevada.

III-B. El VectorBallot en grandes votaciones

A continuación se analiza el espacio que ocupará la urna en un sistema que utiliza *VectorBallot* en un contexto con un gran número de votos y partidos. Para calcular el volumen total, se multiplica el tamaño del voto por el número de votantes. Se tomaran como referencia –por volumen– las elecciones generales de la India de 2009 y las presidenciales de los EE. UU. de 2008, y –por proximidad– las generales españolas de 2011. Para simplificar los cálculos y encontrar una cota superior del volumen de datos, en el cuadro II se calcula el tamaño del voto tomando una circunscripción única. A efectos prácticos no todos los partidos se presentan en todas las circunscripciones y se podría reducir el tamaño de los votos.

País	Partidos	T. Voto	Censo	Est.	Part.	Real
India	≈350	1.5 MB	714M	1PB	428M	630TB
EE. UU.	24	112KB	213M	22TB	133M	14TB
España	62	283KB	34M	9TB	25M	7TB

Cuadro II
TAMAÑO DE VOTO (**T. VOTO**) Y VOLUMEN TOTAL DE DATOS, POR PAÍSES. **EST.**: ESTIMACIÓN DEL VOLUMEN DE DATOS CON UN 100 % DE PARTICIPACIÓN. **PART.**: PARTICIPACIÓN REAL.

Como se puede observar, existe un gran cantidad de datos que se transmiten desde los dispositivos de emisión de voto y son almacenados en la urna electoral. Las consecuencias de

esto, en un sistema de votación, son varias. (i) El sistema de urna digital tiene que ser capaz de procesar y albergar tal cantidad de datos. (ii) El sistema debe soportar los altos picos de demanda que puedan existir durante el periodo, limitado en muchos casos a pocas horas en que está permitido votar, y evitar una posible denegación de servicio (DoS). (iii) El sistema tendrá una demanda de cálculo notable, pues tendrá que agregar todas las posiciones de los vectores con sus acumulados respectivos y comprobar todas la ZKPs. Además, (iv) cuantos más datos haya en la urna más difícil y complejo es garantizar la detección de modificaciones.

III-C. Aplicación de los Dummy Votes

En esta sección se describe como proveer la detección de integridad en una urna digital de un sistema de votación basado en el esquema *VectorBallot*. Siguiendo las propiedades básicas como *indistinguibilidad* e *identificabilidad* de los *dummy votes* y las directrices presentadas en la Sección II, se detallan las modificaciones del esquema del *VectorBallot* [KY04], [KY10] necesarias para que éste posea un mecanismo de detección de la integridad de la urna. Las adaptaciones han sido realizadas teniendo en cuenta un entorno de votación supervisado, es decir controlado por autoridades, y basado en el uso de DREs.

III-C1. Generación y publicación de los Dummy Votes: Para que un *dummy vote* sea *indistinguible* debe tener el mismo aspecto que un voto válido. En este caso, un voto según la propuesta del *VectorBallot* está formado por un vector de *c* posiciones, *c+1* ZKPs y una firma digital. Entonces, las acciones que la entidad de confianza debe realizar para *generar* un *dummy vote* correctamente son las siguientes:

1. Generar un número aleatorio $r \in [0, c-1]$.
2. A todas las posiciones del vector excepto la posición r cifrar el valor 0 utilizando la clave pública de la votación.
3. En la posición r cifrar el valor 1.
4. Generar las pruebas de cada posición y del agregado de acuerdo a su contenido.
5. Generar una pareja de claves para un usuario de auditoría.
6. Calcular el resumen de todo el vector y firmarlo con la clave privada anterior.

El resultado final es un *dummy vote indistinguible* e *identificable*. La entidad de confianza deberá repetir estas operaciones tantas veces como *dummy votes* se requieran, es decir, *d* veces.

En cambio, el *VectorBallot* no afecta al modo de *publicación* de los *dummy votes*. La publicación se realiza de igual manera que en método descrito en la Sección II-A2.

III-C2. Inserción de los Dummy Votes: En un entorno supervisado basado en el uso de DREs como máquinas de emisión de votos, los *dummy votes* generados por la entidad de confianza son enviados a través de un canal de comunicación seguro a cada DRE. Entendiendo que un DRE es una entidad de confianza, la inserción se hace automáticamente por el DRE en intervalos de tiempo aleatorios en función de la cantidad de *dummy votes* a enviar. Así, el modo y el momento de inserción de cada *dummy vote* mantiene su *indistinguibilidad*.

III-C3. Apertura y verificación de los Dummy Votes: En un sistema de votación basado en *VectorBallot*, el escrutinio se realiza mediante la agregación por componentes gracias a las propiedades homomórficas. De este modo, se obtiene un valor agregado por cada opción de voto. Entonces, se descifran con la clave privada de votación y se les restan el número de *dummy votes* por opción de voto. Los valores resultantes corresponden a los resultados electorales.

III-D. Incremento del Volumen de datos

La introducción de *dummy votes* incrementará el volumen de datos de la urna. Si se trabaja con un sistema en el que la cantidad de datos es proporcional al número de votantes, como es *VectorBallot*, y se toma un número de *dummy votes* del orden de $\frac{-\ln(\epsilon)}{s}$ (por la desigualdad 2), entonces, la razón entre *dummy votes* y censo es del orden de

$$\frac{m}{n} \approx \frac{-\ln(\epsilon)}{d}$$

donde d se corresponde con el número de votos eliminados.

En el cuadro III se ve como, si se elimina sólo un $0{,}01\,\%$ de los votos ($s = 0{,}0001$) con una probabilidad de detección del $99{,}995\,\%$, se necesita introducir 100.000 *dummy votes*. También se observa que, comparando con los datos del cuadro II, estos valores se corresponden con la eliminación de 71.400 votos en la India, 21.300 en EE. UU. o 3.400 en España. El incremento de volumen de datos es de 151GB en la India, 11GB en los EE. UU. o 27GB en España, sobre unos volúmenes de 1PT, 22TB y 9TB, respectivamente, lo que es asumible con la tecnología actual.

Tal y como se ha visto anteriormente, tendremos la misma probabilidad de detección si se introducen 1.000.000 *dummy votes* y se elimina un $0{,}001\,\%$ de votos (7.140 votos en la India, 2.130 en EE. UU. o sólo 340 en España), aunque entonces el incremento de datos será diez veces mayor.

III-E. Umbral del nivel de detección

En algunas elecciones se ha dado el caso que los resultados han dependido de unos pocos votos. Un ejemplo son las elecciones presidenciales del año 2000 de los EE. UU. en Florida. El resultado final se decidió por un ajustado margen de 537 votos sobre un censo de unos 11 M. de personas. Esto representa un 0,005 % del censo, comparable con el nivel de detección presentado en los ejemplos anteriores. Además, destaca que sólo en el condado de Duval se contaron 27000 votos nulos (0.25 % sobre el censo de Florida) probablemente atribuibles a errores del sistema de votación [Mer02]. Teniendo en cuenta lo anterior, en función de cada caso se deberá elegir una tolerancia suficientemente satisfactoria.

IV. Conclusiones y trabajo futuro

En este artículo se ha descrito un método para detectar posibles ataques contra la integridad de la urna digital en elecciones electrónicas. Este mecanismo permite la verificación de la integridad de la urna de manera probabilística. La seguridad de este método se basa en la *indistinguibilidad*

dummy votes (m)	s	India			EE. UU.			España			p	$\epsilon = 1 - p$
		m en %	Tamaño	d	m en %	Tamaño	d	m en %	Tamaño	d		
100.000	10^{-4}	0,014	151GB	71400	0,047	11GB	21300	0,29	27GB	3400	0,99995	0,00005
100.000	10^{-5}	0,014	151GB	7140	0,047	11GB	2130	0,29	27GB	340	0,63	0,37
500.000	10^{-5}	0,070	753GB	7140	0,23	54GB	2130	1,47	135GB	340	0,993	0,007
1.000.000	10^{-5}	0,14	1,5TB	7140	0,47	107GB	2130	2,9	270GB	340	0,99995	0,00005

Cuadro III
m en % : TANTO POR CIENTO DE *dummy votes* SOBRE EL CENSO, POR CADA PAÍS. Tamaño: INCREMENTO DE TAMAÑO DEBIDO A LOS *dummy votes*. d: VOTOS ELIMINADOS. **P**: PROBABILIDAD DE DETECCIÓN. ϵ: TOLERANCIA

y la *identificabilidad* de los *dummy votes* junto con su correcta generación, distribución, inserción en la urna y verificación.

Se ha demostrado que la probabilidad de detección puede ser configurable según las necesidades de la votación. Ésta depende estrictamente de la cantidad de *dummy votes* existentes en la urna y del número máximo de votos que pueden ser eliminados. Aumentar el grado de detección supone aumentar la cantidad de *dummy votes*, y por consiguiente, el tamaño de la urna. No obstante, con una cantidad moderada de *dummy votes* se puede conseguir un grado suficiente de detección.

En el sistema presentado, la generación de los *dummy votes* es realizada por una Trusted Third Party (TTP). Como *trabajo futuro* sería deseable diseñar un mecanismo de generación de *dummy votes* distribuido sin TTP. Los *dummy votes* se generarían sin que nadie conociera la cantidad de *dummy votes* por opción de voto. Otro aspecto a mejorar puede ser la reducción del coste que supone verificar la existencia de todos los *dummy votes* dentro de la urna, ya que se debe realizar una búsqueda exhaustiva. Además, también puede resultar interesante adaptar este mecanismo a un sistema de votación diseñado de forma jerárquica donde los escrutinios se realizan parcialmente en cada zona electoral.

AGRADECIMIENTOS

Los autores son los únicos responsables de los puntos de vista expresados en esta artículo, que ni necesariamente reflejan la posición de esta organización, ni la comprometen. Este trabajo está parcialmente financiado por el Ministerio de Ciencia e Innovación (a través de los proyectos eAEGIS TSI2007-65406-C03-01, CO-PRIVACY TIN2011-27076-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004 y Audit Transparency Voting Process IPT-430000-2010-31), por el Ministerio de Industria, Comercio y Turismo (a través de los proyectos eVerification2 TSI-020100-2011-39 y SeCloud TSI-020302-2010-153) y por la Generalitat de Catalunya (con la beca 2009 SGR 1135).

REFERENCIAS

[AJCCR10] Jordi Pujol Ahulló, Roger Jardí-Cedó, and Jordi Castellà-Roca, *Verification systems for electronic voting: A survey*, Electronic Voting 2010, EVOTE 2010, 4th International Conference, Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting.CC, July 21st - 24th, 2010, in Castle Hofen, Bregenz, Austria (Robert Krimmer and Rüdiger Grimm, eds.), LNI, vol. 167, GI, 2010, pp. 163–177.

[BMQR07] Jens-Matthias Bohli, Jörn Müller-Quade, and Stefan Röhrich, *Bingo voting: Secure and coercion-free voting using a trusted random number generator*, Proceedings of the First International Conference on E-Voting and Identity (VOTE-ID '07), Lecture Notes in Computer Science, vol. 4896, Springer, October 4-5 2007, pp. 111–124.

[Bra05] Felix Brandt, *Efficient cryptographic protocol design based on distributed el gamal encryption*, In Proceedings of 8th International Conference on Information Security and Cryptology (ICISC, SpringerVerlag, 2005, pp. 32–47.

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers, *A secure and optimally efficient multi-authority election scheme*, Springer-Verlag, 1997, pp. 103–118.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson, *How to prove all np-statements in zero-knowledge, and a methodology of cryptographic protocol design*, Proceedings on Advances in cryptology—CRYPTO '86 (London, UK), Springer-Verlag, 1987, pp. 171–185.

[Har09] Larry Hardesty, *Cryptographic voting debuts. a new system for ensuring accurate election tallies, which mit researchers helped to develop, passed its first real-world test last tuesday*, Massachusetts Institute of Technology News (2009), Online Apr. 2012.

[Hir01] Martin Hirt, *Multi-party computation: Efficient protocols, general adversaries, and voting*, Ph.D. thesis, ETH Zurich, September 2001, Reprint as vol. 3 of *ETH Series in Information Security and Cryptography*, ISBN 3-89649-747-2, Hartung-Gorre Verlag, Konstanz, 2001.

[KY04] Aggelos Kiayias and Moti Yung, *The vector-ballot e-voting approach*, In FC 2004, volume 3110 of LNCS, Springer-Verlag, 2004, pp. 72–89.

[KY10] ———, *The vector-ballot approach for online voting procedures.*, Towards Trustworthy Elections'10, 2010, pp. 155–174.

[Mer00] Rebecca Mercuri, *Electronic vote tabulation checks and balances*, Ph.D. thesis, University of Pennsylvania,School of Engineering and Applied Science, Department of Computer and Information Systems, Philadelphia, PA, USA, October 2000, Supervisor-Norman I. Badler.

[Mer02] ———, *Government: a better ballot box?*, IEEE Spectr. **39** (2002), no. 10, 46–50.

[MN06] Tal Moran and Moni Naor, *Receipt-free universally-verifiable voting with everlasting privacy*, Proceedings of 26th International Cryptology Conference (CRYPTO '06) (Cynthia Dwork, ed.), Lecture Notes in Computer Science, vol. 4117, Springer, September 2006, pp. 373–392.

[Nor09a] Norwegian Ministry of Local Government and Regional Development, *E-vote 2011: Contractor solution specification*, December 2009, Online Feb. 2010.

[Nor09b] ———, *E-vote 2011: Project directive for e-valg 2011*, February 2009, Online Feb. 2010.

[Pai99] Pascal Paillier, *Public-key cryptosystems based on composite degree residuosity classes*, Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT '99), Springer Verlag, 1999, pp. 223–238.

[Ros10] Burton Rosenberg (ed.), *Handbook of financial cryptography and security*, Chapman & Hall/CRC, 2010.

[SCC+10] Alan T. Sherman, Richard Carback, David Chaum, Jeremy Clark, Aleksander Essex, Paul S. Herrnson, Travis Mayberry, Stefan Popoveniuc, Ronald L. Rivest, Emily Shen, Bimal Sinha, and Poorvi L. Vora, *Scantegrity mock election at takoma park.*, Electronic Voting (Robert Krimmer and Rüdiger Grimm, eds.), LNI, vol. 167, GI, 2010, pp. 45–61.

[SDW08] Daniel Sandler, Kyle Derr, and Dan S. Wallach, *Votebox: a tamper-evident, verifiable electronic voting system*, Proceedings of the 17th conference on Security symposium (SS '08) (Berkeley, CA, USA), USENIX Association, 2008, pp. 349–364.

[Sha79] Adi Shamir, *How to share a secret*, Commun. ACM **22** (1979), no. 11, 612–613.