Scalable Multi-Source Video Streaming Application over Peer-to-Peer Networks

Carlos Hernández Gañán

Department of Telematics Engineering

Polytechnical University of Catalonia

A thesis submitted for the degree of

Master of Science

9th September 2009

I thank the Father, Lord of heaven and earth, for hiding the things I learned during this thesis from the learned and wise, and revealing them to me.

Acknowledgements

This thesis arose in part out of years of research that has been done before I came to Telematics Engineering group. By that time. I have worked with a great number of people whose contribution in assorted ways to the research and the making of the thesis deserved special mention. It is a pleasure to convey my gratitude to them all in my humble acknowledgment.

First of all, I would like to express my gratitude to my supervisor, José Luis Muñoz, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate his vast knowledge and skills in many areas, and his assistance in writing reports. Above all and the most needed, he provided me unflinching encouragement and support in various ways. His truly scientist intuition has made him as a constant oasis of ideas and passions in science, which exceptionally inspire and enhance my growth as a student, a researcher and a scientist want to be.

I am deeply indebted to Jorge Mata. Without his guidance, support and good nature, I would never have been able to develop this thesis successfully. I benefited greatly from his ideas and insights. His involvement with his originality has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

Some debts are hard to put into words. My research colleagues Javier Parra-Arnau, Óscar Esparza, Juan José Alins all know why their names are here.

My last, but not least gratitude is for my parents, it is difficult to find words to express my gratitude and thanks to both of you.

I realize that not all people who contributed either directly or indirectly to my study are mentioned in this page. From the deepest of my heart, I would like to thank all of you...

TABLA DE CONTENIDO

LISTA DE FIGURAS

OBJETO DEL TRABAJO

Este trabajo surge en el marco del Máster de Ingeniería telemática de la Universidad Politécnica de Catalunya. Concretamente, este trabajo trata de ofrecer una visión general de la votación electrónica y en concreto comentar un nuevo esquema de votación surgido en 2006. Éste es introducido por Ronald Rivest y compañía en el *Workshop On Privacy In The Electronic Society* en el artículo "Self-contained paper-based cryptographic voting" [11].

Así, el principal objetivo de este trabajo es comentar este nuevo método de votación entrando con especial detalle en los temas criptográficos. Partiendo de la base del artículo citado, se introducirán los nuevos conceptos introducidos en este artículo y cómo éstos mejoran los diferentes aspectos de seguridad que envuelven cualquier esquema de votación electrónica.

Con tal de cumplir tal finalidad, hemos estructurado el trabajo de la siguiente manera. Primeramente, comenzaremos el trabajo haciendo una introducción a la motivación del uso de la votación electrónica y cómo en diferentes países del mundo se han realizado diferentes pruebas para ver cómo la criptografía puede mejorar tanto los aspectos de seguridad como el escrutinio.

Después de esta introducción para contextualizar el marco en el que actualmente se encuentra la votación electrónica, se comentaran los diferentes aspectos que rodean a unas elecciones durante el proceso de votación y recuento. Así, se comentan las diferentes propiedades requeridas y deseadas durante la votación como medio democrático. A continuación, se explicitaran los esquemas tradicionales de votación, comentando sus puntos fuertes y sus debilidades. Para finalizar este bloque, se introducen los diferentes objetos que la criptografía podría mejorar durante unas votaciones.

En el segundo bloque de este trabajo, se introducen los conceptos criptográficos necesarios para entender diferentes aspectos que giran en torno a una votación electrónica. Así, en primer lugar introducimos la notación que se usa durante la explicación de dichos conceptos. Seguidamente, se da una visión general de conceptos básicos de criptografía como la criptografía de clave pública y los diferentes tipos de seguridad. De la misma forma se introducen los esquemas de clave pública homomórfica y umbral ya que son de especial interés en el artículo.

En el tercer bloque, que constituye el núcleo del trabajo, se explica el artículo citado. Así, se comenta el esquema propuesto por los autores y cómo éste hace uso de la criptografía para asegurar los diferentes puntos deseables de la votación criptográfica, introducidos en la primera sección del trabajo. Siguiendo el esquema del artículo, además de comentar la propuesta también se tratan las diferentes amenazas que puede sufrir el modelo propuesto y como se defiende contra éstas.

Finalmente, se tratan las conclusiones y el impacto que tiene este nuevo modelo de votación mediante el uso de técnica criptográficas.

INTRODUCCIÓN

Puede parecer que los intentos de utilizar las Tecnologías de la Información y de las Comunicaciones (TICs) en los diversos aspectos del voto electrónico (VE) son recientes, pero no es así. De hecho una de las primeras aplicaciones de las tecnologías electromecánicas de finales del siglo XIX fue su uso para el ejercicio del VE y del recuento de votos posterior. Así Thomas Alva Edison en 1.869 firmó una aplicación de patente para un sistema de grabación de voto eléctrico el cual luego sería utilizado para su primera patente, ya que nadie quiso utilizarla. En 1.892 Jacob H. Myers diseña la AVM {Automatic Voting Machine) que se utilizó en varias ocasiones en el estado de New York. Era una máquina basada en mecanismos de levas que se siguieron utilizando posteriormente en otras máquinas similares (Davis y Boma machines). Con la aparición de los primeros ordenadores a mediados de los años cuarenta se retomó la posibilidad de utilizarlos para el VE y varios prototipos vieron la luz a mediados de los 60. Más tarde se han venido utilizando de modo generalizado en todo el mundo para el recuento de votos y el cálculo de resultados finales. La idea de modernizar los procesos electorales utilizando tecnologías basadas en la electrónica proviene de pensadores como Fromm (1.955), Fuller (1.963). Arterton (1.987) y Rheingold (1.993). En la actualidad raro es el país que no haya intentado desarrollar pruebas de voto electrónico con diversos tipos de soluciones y tecnologías.

Hoy día es común que en las noticias que elaboran los medios de comunicación sobre el desarrollo de procesos electorales aparezca el término voto electrónico, bien como referencia al sistema empleado, bien como complemento a la noticia para indicar hacia dónde se encaminan los sistemas de votación. El uso de este término se remonta a 1964 cuando, por primera vez, se emplearon ordenadores en EEUU para realizar ciertas funciones ligadas al proceso electoral. Desde entonces, este término viene empleándose para identificar sistemas de votación

de naturaleza muy diversa. A continuación se mencionan algunas de las experiencias de voto más recientes o relevantes que han empleado ordenadores en alguna fase del proceso electoral.

En EEUU en las elecciones presidenciales de noviembre de 2000 el 69% de los votantes utilizó la vía electrónica para emitir su voto, utilizando diversos y anticuados mecanismos como la tarjeta perforada, el voto óptico y la máquina electrónica de registro automático. Estos sistemas presentan como principales inconvenientes la confianza ciega que se deposita en los expertos que supervisan los procesos y la falta de mecanismos de transparencia, lo que lleva a cuestionar en numerosas ocasiones la validez de estos sistemas. Como caso destacable de fallo de estos sistemas cabe mencionar el que tuvo lugar en el estado de Florida, donde la falta de normativa y control propició que muchos votantes que emplearon el método de tarjeta perforada no pudieran saber con certeza qué opción era la que habían marcado.

También hay que destacar las experiencias llevadas a cabo en Brasil. Este país aprobó en octubre de 1995 la Ley Electoral que marca las directrices del voto electrónico con la intención de eliminar el fraude electoral y reducir el tiempo de escrutinio. El proceso de votación se lleva a cabo a través de una especie de cajero automático, dotado de un monitor, en el que van apareciendo los candidatos y donde los votantes pueden realizar su selección oprimiendo un botón. Al finalizar la jornada electoral, se bloquea la urna mediante una clave y automáticamente se imprime una copia de los resultados, a la vez que se obtiene un disquete que se lleva de inmediato a un Centro de Recuento para su cómputo. La urna electrónica fue el único método de votación en las elecciones a presidente de la República en octubre de 2002 donde 115 millones de votantes lo emplearon.

Venezuela también ha incluido en el Reglamento General Electoral las instrucciones para que el proceso de votación, escrutinio y publicación de

resultados del proceso de votación se realicen de manera automática. A diferencia del caso de Brasil este Reglamento no especifica el funcionamiento de ninguna máquina de voto en particular. En las pasadas Elecciones Municipales de 2000 se confió a una empresa española la automatización del proceso de votación. Con este sistema, el elector emite el voto en la urna electrónica y automáticamente se acumula para su recuento y difusión sin intervención humana. Este proceso tiene como característica singular que es auditable por empresas y organizaciones externas al proceso electoral. Sin embargo, las primeras implantaciones de voto electrónico en los procesos electorales venezolanos no han sido muy afortunadas y han estado plagadas de problemas, básicamente motivados por la desconfianza hacia los resultados obtenidos.

En Europa se han realizado también varías experiencias. En Bélgica, se iniciaron en 1991 con una prueba piloto en el cantón de Verlaine. El método empleado es el de tarjeta con banda magnética que es entregada a cada elector en el momento de su identificación. Posteriormente, éste graba su opción de voto, utilizando para ello una cabina electoral que dispone de mía pantalla, en la que se presentan las distintas opciones, y un lápiz óptico con el que se realiza su selección. Después, acude a la Mesa Electoral donde se introduce su voto en la urna. Como resultado de las pruebas realizadas se ha ido sustituyendo el sistema tradicional de voto mediante papeleta por el de tarjeta magnética. En las últimas elecciones municipales celebradas el pasado 8 de octubre de 2000. este procedimiento fue usado por el 44% de los electores, no estando todavía extendida su aplicación a todos los electores debido al coste que supone la implantación del sistema.

También se han hecho varios experimentos de voto a través de Internet, ligados a elecciones gubernamentales. En 1996 en EEUU, el Partido Reformista en las elecciones a Gobernador puso el voto a través de Internet a disposición de los miembros del partido que no podían acudir a la convención, aunque con mías garantías de seguridad limitadas, ya que no garantizaba el anonimato del voto.

Desde entonces se han venido realizando en este país diversas experiencias análogas, con resultados no tan prometedores como se esperaban en cuanto a participación ciudadana y aceptación del sistema. En el año 2000 el Partido Demócrata de Arizona también ofreció la opción de voto por Internet en sus primarias presidenciales. Esta ha sido la única elección gubernamental a gran escala de carácter vinculante. En este año el Partido Demócrata canadiense tiene planeado celebrar sus elecciones primarias haciendo uso de voto electrónico a través de Internet.

En el estado de California, la Secretaría de Estado convocó a la *Internet Voting Task Force* para estudiar la posibilidad de emplear Internet para llevar a cabo las elecciones en California. Se reunieron expertos en el campo de seguridad, legislación y participación ciudadana y elaboraron un informe, publicado en enero de este año. Este informe recoge los requisitos de seguridad exigibles al nuevo sistema de votación y pone de relieve la necesidad de avanzar con cautela en el proceso de introducción del nuevo sistema de votación, ya que la posibilidad de amenazas o pirateo del sistema pondría en peligro el esfuerzo realizado. Sin embargo, afirma que. a pesar de los retos que supone el desarrollo del nuevo sistema, es técnicamente posible utilizar Internet para desarrollar un método de votación, al menos tan seguro como los sistemas actuales.

A este respecto, esta Secretaría encargó a la empresa Safevote la preparación y realización de una prueba de un sistema de votación electrónica a través de Internet llevada a cabo en el condado de Contra Costa. California, a primeros de noviembre de 2000.

También a nivel europeo se han realizando inversiones importantes para la puesta en marcha de sistemas de votación electrónica avanzados. Dentro del V Programa Marco de Tecnologías para la Sociedad de la Información, el proyecto CyberVote (septiembre 2000-marzo 2003) está desarrollando un prototipo de un

sistema de votación diseñado para ser usado en elecciones locales, regionales, nacionales o europeas, basado en el empleo de las tecnologías móviles y fijas de Internet para que los votantes accedan al sistema. En este proyecto participa un consorcio liderado por EADS Matra Systémes & Information de Francia y que agrupa conjuntamente: British Telecommunications de Gran Bretaña. XOKIA Research Centre de Finlandia. K.U.Leuven Research & Development de Bélgica. Technische Universiteit Eindoven de Holanda. Freie Hansestadt Bremen de Alemania. Mairie d'Issy-les-Moulineaxus de Francia y Kista Stadsdelsámnd de Suecia. A primeros de diciembre de 2002 se llevó a cabo una experiencia piloto con este sistema en la Universidad de Bremen (Alemania), permitiendo a los profesores, estudiantes y personal de la universidad eligieran sus representantes en distintos consejos. La votación se llevó a cabo desde kioscos donde los votantes se autenticaron haciendo uso de mía tarjeta inteligente. El prototipo también será probado en el año 2003 en elecciones que tendrán lugar en Alemania. Francia y Suiza.

El proyecto *E-PoII* (Electronic pollina system for remote voting operations) [4] ha sido desarrollado por el consorcio formado por Siemens (Alemania), el Ministerio del Interior italiano. Ancitel (Italia). France Telecom y AECs (Francia) y Municipium (Polonia), durante el periodo comprendido entre septiembre de 2000 y agosto de 2002. Su objetivo ha sido el desarrollo de mía Red Europea de Voto Virtual (European Virtual Ballot Network. EVBN) capaz de soportar el nuevo proceso de votación. El proyecto ha investigado las comunicaciones móviles de banda ancha basadas en el estándar UMTS para proporcionar a la red el ancho de banda y la seguridad requeridos. El sistema de reconocimiento de votantes se basaba en una tarjeta inteligente innovadora, con un lector de huellas dactilares que permitía realizar el reconocimiento de los votantes con total seguridad. Este proyecto ha incluido la realización de diversas pruebas piloto en Avellino (Italia. 2001). y en Marignac. Campobasso y Vandoeuvre les Xancy (Francia. 2002).

También a nivel europeo la empresa *VoteHere* ha patentado un sistema de voto electrónico que incluye el registro seguro de los votantes y su autenticación, empleando credenciales basadas en criptografía de clave pública y el envío de los votos a una unía remota. Este sistema fue empleado el pasado mes de mayo dos circunscripciones locales del Reino Unido las primeras elecciones institucionales con voto a través de Internet.

Asimismo, en el ámbito nacional existe un creciente interés en favorecer la votación electrónica. El 6 de marzo de 2001 se aprobó en el Pleno del Senado la creación de una Ponencia formada por la Comisión de la Sociedad de la Información y del Conocimiento y la Comisión Constitucional para estudiar la implantación de los sistemas electrónicos de ejercicio del derecho a voto y recuento, así como la reforma de referente al Régimen Electoral General (5 1985). la Ley Orgánica sobre regulación de las distintas modalidades de referéndum (2 1980). la ley Orgánica sobre la iniciativa legislativa popular (3.1954) y de cuantas otras sean necesarias al respecto.

En conclusión, la introducción del voto electrónico es inevitable y positiva, pero en cualquier caso debe ser gradual y contemplar el debate y las pruebas necesarias.

1. LA SEGURIDAD EN EL VOTO

1.1. PROPIEDADES DE LA VOTACIÓN

Para ilustrar las complejidades de la votación, es útil considerar un escenario hipotético con los siguientes caracteres:

- **Alicia** y **Adriana**, dos votantes,
- **Carlos**, un coaccionador que deseas influir a Alicia,
- **Rojo** y **Azul**, opciones entres las que un votante puede escoger en la elección.

Alicia quiete votar **Azul,** mientras que Adriana quiere votar **Rojo**. Carlos quiere coaccionar Alicia para que vote **Rojo**, cambiando así su elección. Carlos, el coaccionador, puede un interventor u oficial de las elecciones.

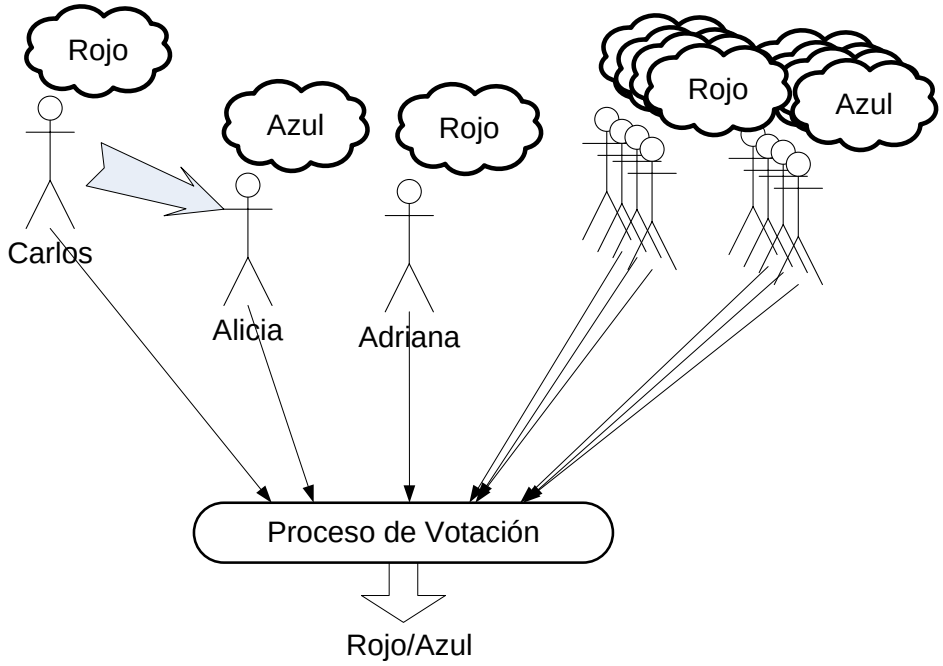

Figura 1. Participantes en un proceso electoral

La Figura 1 muestra el escenario planteado donde Carlos quiere coaccionar a Alicia para que cambie su voto.

1.1.1. Verificabilidad vs. Secretismo

En unas elecciones, existe un conflicto funcional entre verificabilidad y secretismo. Por una parte, Alicia quiere verificar que el proceso de votación ha funcionado de manera correcta, y en particular quiere que su vota haya sido contado de manera apropiada como Azul. Sin embargo, si Alicia obtiene suficiente información del proceso de votación de manera que pueda convencer a Carlos de qué votó, y entonces la venta de votos se convierte en una amenaza: Carlos podría ofrecer a Alicia dinero para que cambie su voto de Azul a Rojo.

De alguna manera, se busca que Alicia obtenga suficiente información para que verifique personalmente que su voto fue contado como Azul, pero no demasiada información para que pueda convencer a Carlos. Concretamente, si Alicia vota Azul y Adriana vota Rojo, ambas deberían recibir la seguridad de que su voto ha sido contado de acuerdo a sus preferencias. Además, ambas pueden decirle a Carlos que supuestamente votaron Rojo. Alicia estaría mintiendo, y Adriana estaría contando la verdad pero Carlos no podría discernir la diferencia. Así, Carlos no tiene ningún incentivo para comprar votos, ya que el no puede saber si su dinero sería usado de forma útil.

1.1.2. Modelado de la Amenaza

Una crítica común y reciente de los fallos del sistema de votación compara la votación en sí misma con sistemas complejos existente, como el procesado de transacciones en banco o las operaciones de los aviones [8]. A primera vista, la crítica parece garantizada: si un puede construir grandes cilindros de aluminio, cargarlos con cientos de personas, ponerlos a volar miles de ellos cada día a la mitad de la velocidad del sonido y a una altura de 9 kilómetros, aterrizar en el destino correcto, todo con menos de un accidente fatal por año, incluso a pesar de la existencia de adversarios malévolos, entonces seguro que se puede construir un sistema de votación fiable. De forma similar, si un banco procesa millones de transacciones diariamente, registrando cada euro que entra y sale de

la cuenta de cada cliente, entonces seguro que pueden registra de forma fiable 100 millones de votos una vez cada 4 años y proporcionar un conteo adecuado.

Estas analogías cometen tres errores. Primero, el incentivo para hacer unas elecciones está bastamente subestimado. En segundo luego, el modelo del adversario para aviones requiere menos restricciones que para unas elecciones. Y por último, el fallo de detección y el proceso de recuperación en el caso de fallo un avión y un banco está generalmente bien entendido: gracias a auditorias, se pueden tomar las acciones oportunas. En el caso de unas elecciones, no queda claro que los fallos puedan ser detectados y, si lo son, la detección es muchas veces cara o incluso imposible.

Incentivo. Influenciar el resultado de unas elecciones nacionales vale mucho dinero. La campaña presidencial cuesta millones de euros para los partidos más fuertes, y, por supuesto, no hay un ganador a priori. Así, un mecanismo para influir el resultado de las elecciones podría valer incluso más que estas campañas electorales. Aunque las pegas con importantes –el fraude en unas elecciones es un delito– uno no puede ignorar los beneficios de cometer este fraude. Incluso con las penas establecidas y aunque las ganancias fueran pequeñas, existe evidencia empírica de que el fraude al votar es una hecho habitual.

Adversarios. El modelo de amenazas para la seguridad de un avión está completamente definido: se asume que los pasajeros pueden ser adversarios, lo que conduce a números chequeos de seguridad y demás prohibiciones. Por el contrario, generalmente se asume que hay tiempo suficiente antes de que despegue un avión para asegurarse que los pilotos no son adversarios. En el momento en que los pilotos embarcan, se asume que son honestos. La presencia de un copiloto índice previsión frente a fallos aleatorios, aunque no supone casi ninguna defensa frente a ataques maliciosos: daños significativos pueden ser infligidos por un piloto granuja, como ha sucedido durante la historia. Afortunadamente, muy poco pilotos adquieren nunca la motivación para cometer actos malévolos.

En el caso de la banca personal, el modelo de amenaza del cliente también se define: los adversarios son normalmente extraños –ladrones, bandidos de cajeros automáticos, etc.–. Cabe remarcar, que todos los datos bancarios están disponibles para los participantes honestos: tanto para los oficinistas del banco como para los clientes que pueden ver las transacciones en sus cuentas bancarias.

Contrariamente, en unas elecciones, cualquier participante, incluyendo las personas de dentro pueden querer manipular los resultados. Los votantes pueden ser corruptos. Los interventores también pueden estar corruptos. Es decir, no se puede hacer ninguna suposición de honestidad de ningún participante, y todos ellos están motivados potencialmente a cometer fraude, ya que el resultado es muy importante para todos ellos.

Detección de fallos y recuperación. En la aviación, los fallos son difíciles de perdérselos: los aviones con fallos se estrellan trágicamente, como es obvio. Por ello, se guarda listas de los nombres de los pasajeros y de la tripulación, del equipo a bordo, y del propio vuelo, incluyendo el uso de “cajas negras” para recuperas estos datos en caso de estrellamiento. Si el avión se estrella, entonces se inicia una larga y cara investigación, para determinar cuáles fueron los problemas técnicos y así cambia las recomendaciones de seguridad.

En la banca, la situación es bastante similar. Existe mucha inversión y énfasis en detectar y resolver fallos: los clientes reciben facturas con cada transacción que pueden usar para reclamar posibles fallos. Además, se pueden pedir duplicados de estas facturas con facilidad. Por otra parte, se efectúan grabaciones de vídeo de las transacciones realizadas en cajeros automáticos. De la misma forma, se guarda los registros a las actividades electrónicas. Si un cliente de un banco o el mismo banco encuentra una discrepancia, una revisión de los rastros de las transacciones conduce al descubrimiento del malfuncionamiento y posterior rectificación del problema con poca sobrecarga.

Sin embargo, la detección de fallos en unas elecciones usando los protocolos de votación actuales es bastante difícil. Es bastante probable que un fraude cometido pase inadvertido, dado que una parte de la información e destruida de forma voluntaria para

asegurar la seguridad de la votación. Por otra parte, si un error se detecta, no queda claro como se puede resolver: la única solución puede ser relanzar las elecciones, lo que podría cambiar el resultado de las mismas.

Fijando las Analogías. Si el sistema votación se compara con la banca, se debería imaginar un sistema de banca donde el banco no pudiese saber el balance de situación del cliente, e incluso el cliente no pudiere probar su balance a su esposa, aunque el cliente reciba la seguridad de que su dinero está a salvo. Por otro lado, si el sistema de votación se compara a la aviación, entonces se debería imaginar que los pilotos están intentado estrellarse de forma habitual, y que se debiere asegurar que ellos casi nunca tienen éxito, aunque en ese mundo imaginario los accidentes de avión fuesen difíciles de detecta. Estas analogías permiten apreciar los requerimientos que afronta un sistema de votación.

1.1.3. Revisando un proceso parcialmente secreto

Una votación es particularmente difícil porque requiere de una revisión pública del proceso que a su vez debe ser secreto. Este secretismo no puede ser garantizado confiando en una tercera parte: incluso los auditores no deben conocer lo que los ciudadanos votan. Además, esta auditoría debe convencer a observadores que mutuamente desconfían entre ellos.

Estos requisitos aparentemente contradictorios indican la necesidad de criptografía. Sin embargo, antes de que exploremos las soluciones ofrecidas mediantes técnicas criptográficas, se van a introducir las propiedades de los sistemas de votación clásicos.

1.2. SISTEMAS DE VOTACIÓN CLÁSICOS

En el conflicto que existe entre la necesidad de que las votaciones sean secretas y que puedan ser auditadas, los sistemas electorales a menudo favorecen a uno de estos compromisos. Todos los sistemas usados en España desde la introducción del voto secreto han favorecido la posibilidad de auditorías. Típicamente, el secretismo se asegura mediante la disociación física de la identidad del votante y el voto, por ejemplo mediante la deposición de un voto anónimo en una caja. Por otra parte, la revisión de unas

elecciones generalmente depende de una cadena de custodia adecuada de los votos y los elementos para votar.

1.2.1. Seguridad de la cadena de custodia

Algunas soluciones de votación usan un tipo de máquina de votación para ayudar a los votantes a preparas y emitir un voto. Estas máquinas las construyen compañías privadas, de acuerdo con varios estándares. A la misma vez, agencias independientes testean, evalúan y certifican estos equipos de votación. Además, interventores y demás oficiales ejecutan testes adicionales, normalmente durante las semanas previas a unas elecciones, y, algunas veces, en paralelo a las mismas elecciones, para asegurar que dichas máquinas funcionan como se espera. El día de las elecciones, los votantes emiten sus votos por medio de estas máquinas, y los votos resultantes son introducidos en una caja, ya sea física o digital. Seguidamente, los oficiales transportan estas cajas a un lugar donde contar los votos. Finalmente, el resultado total se publica. Esta aproximación presenta tres limitaciones principales:

1. **Los votantes deben ser verificador por un apoderado:** sólo los oficiales de las elecciones pueden asegurar que varios procedimientos de test son adecuados para empezar. Los votantes reciben algunas indicaciones indirectas de los resultados del test. Sin embargo, los votantes no pueden verificar de forma directa por ellos mismos que las cajas de votación (físicas o digitales) son entregadas al final del día electoral de la forma correcta.

2. **La verificación depende de la cadena de custodia:** los oficiales de las elecciones deben mantener una cadena de custodia para defender contra problemas. Para que unas elecciones funcionen correctamente, cada transición debe llevarse a cabo de forma apropiada, y la cadena de custodia de las máquinas y los votos debe ser respetada estrictamente todas las veces. Un solo fallo puede acarrear la corrupción de los resultados de las elecciones.

3. **La recuperación es muy difícil:** los mecanismos detectores de errores son pocos y muy poco precisos, tan sólo son capaces de discernir errores honesto, pero rara vez ataques maliciosos. Si se detecta un error, algunos sistemas de votación permiten un recuento de votos. Dicho recuento sólo puede encontrar fallos del recuento de votos, sin embargo: la recuperación de un fallo de integridad de las cajas de votación conlleva volver a realizar las elecciones, ya que es normalmente imposible decir qué votos son legítimos y cuáles son fraudulentos,

En otras palabras, para implementar el voto secreto, los sistemas de votación actuales se basan en un apoderado de confianza, y una cadena de custodia mediante un mecanismo de verificación, que tiene un "punto de no retorno" que ocurre cuando el voto es entregado. Esta aproximación es altamente restrictiva y conduce a errores. Ya sea papel, escáner óptico, o pantalla táctil, los métodos clásicos de elecciones comprometen la verificabilidad para alcanzar el voto secreto.

La verificación del proceso de custodia se puede ver en la Figura 2.

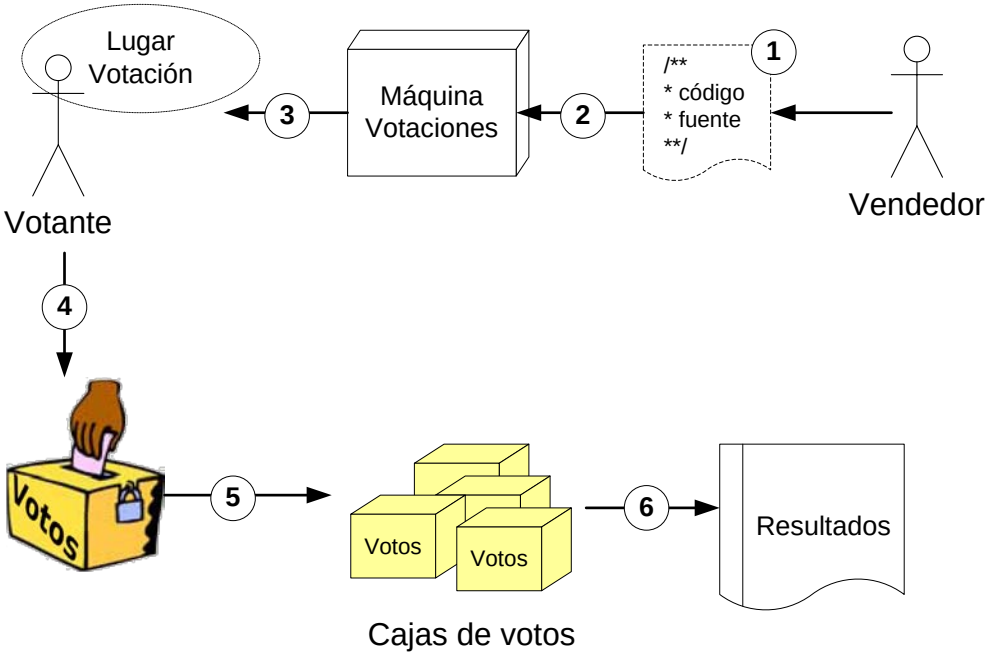

Figura 2. Cadena de custodia proceso de votación

La figura muestra como cada paso debe ser verificado. (1) El código fuente de las máquinas de votación se lee y se comprueba. (2) Se verifica la instalación de las máquinas de votación para asegurar que el software verificado es el que está instalado en éstas. (3) Las máquinas de votación se sellan antes de las elecciones, siendo seguras contra cualquier tipo de ataque físico. (4) Los oficiales de las elecciones se aseguran que sólo los votantes legítimos emiten un voto. (5) Las cajas de votación se sellan y se recogen con cuidado. (6) El recuento se lleva a cabo en un lugar seguro, asegurando que no se inyectan ni quitan votos maliciosamente.

1.3. ESQUEMAS DE VOTACIÓN CRIPTOGRÁFICOS

Ahora consideraremos sistemas de votación criptográficos a alto nivel. Pondremos especial atención a la naturaleza de la verificación final que proporcionan dichos sistemas, y cómo, consecuentemente, cualquier observador puede verificar el funcionamiento correcto de unas elecciones mediante criptografía.

La Figura 3 muestra un esquema de una posible votación extremo a extremo. Sólo son necesarios dos puntos de comprobación. (1) El recibo obtenido de la interacción de votante con la máquina de voto se compara con el tablón de anuncios y el votante comprueba que sea correcto. (2) Cualquier observador comprueba que sólo los votantes elegibles emiten votos y que todas las acciones de conteo que se muestran en el tablón de anuncios son válidas.

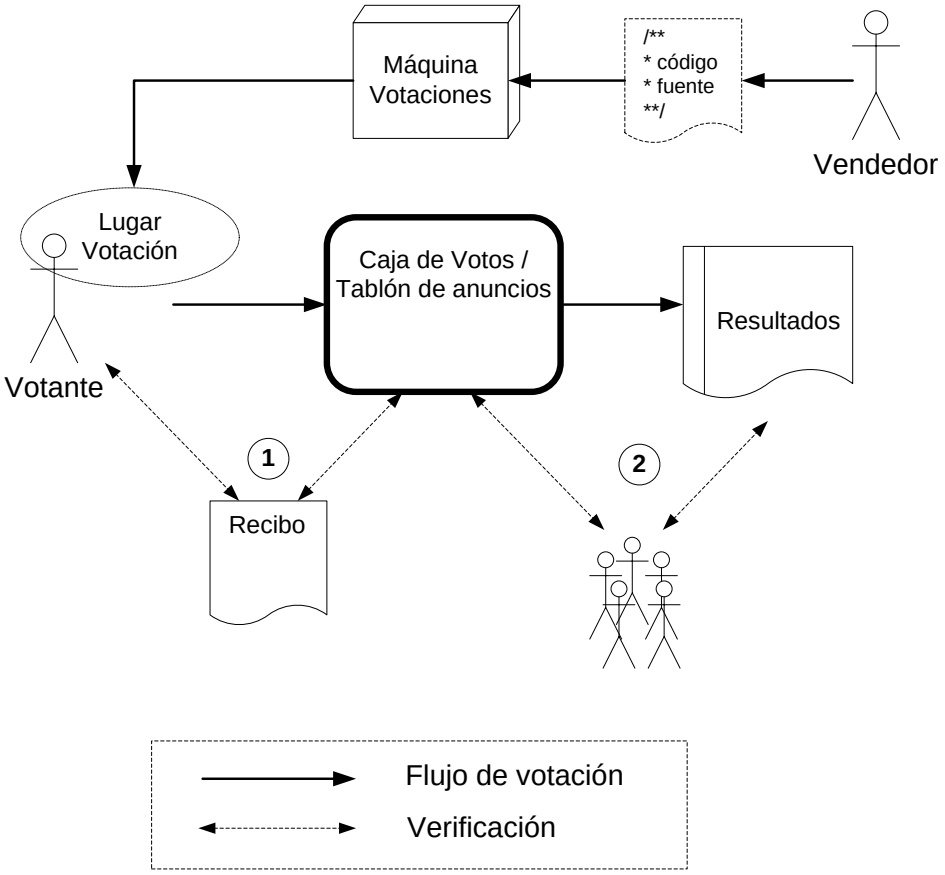

Figura 3. Votación extremo a extremo

1.3.1. Verificabilidad extremo a extremo

Cuando se tratan con sistemas complejos, la ingeniería de software se ha respaldado en el principio "extremo-a-extremo", donde, para mantener el sistema simple, las "inteligencias" del sistema se mantienen en niveles altos de abstracción, en vez de enterrarlas profundamente en la pila de protocolos [7]. Por ejemplo, cuando se encaminan paquetes por Internet, se hacen muy pocas suposiciones acerca el mecanismo de transporte subyacente. Por el contrario, se llevan a cabo comprobaciones tanto por el destinatario como por el emisor para asegurar la integridad extremo a extremo y además se aplican firmas digitales para impedir modificaciones maliciosas de los datos. No se necesitan detalles del encaminamiento de tráfico en ningún caso, a cambio, se preserva una propiedad de inicio a fin, a pesar que lo que suceda por el medio.

Aunque no todos los sistemas son susceptibles de tal diseño, los sistemas de votación sí que lo son. En vez de inspeccionar el código fuente de la máquina de votación y asegurar que la máquina de votación está verdaderamente corriendo ese código, la verificación extremo a extremos comprueba tan sólo la salida de esta máquina. En vez de mantener, un registro de la cadena de custodia de todas y cada una de las cajas de votación, la votación extremo a extremo verifica los resultados de la votación.

Como consecuencia inmediata, no se necesita tener privilegio alguno para verificar las elecciones. En un entorno de cadena de custodia, se tiene que estar atento durante el proceso para estar seguro de su correcta ejecución, y por tanto, sólo los oficiales de las elecciones pueden hacer esto directamente. En un entorno de verificación extremo a extremo, cualquiera puede comprobar las entradas y las salidas mediante pruebas matemáticas. Los detalles del proceso interno se vuelven irrelevantes, y la verificabilidad se vuelve universal, tal y como se muestra en la Figura 3.

La criptografía hace que la verificación extremo a extremo sea posible. A un alto nivel, los sistemas de votación criptográficos renuevan los sistemas de votación de antaño, donde todos los ciudadanos legítimos votaban públicamente, efectuándose el conteo en público para que todo el mundo lo pudiera ver y revisar. Los sistemas criptográficos aumentan esta aproximación con:

1. Encriptación que hace que el voto sea secreto, y
2. Pruebas de conocimiento-cero para proporcionar al público auditor del proceso de conteo.

1.3.2. Tablón de anuncios de los votos

Los protocolos de votación criptográficos giran en torno a un tablón de anuncios digital. Como su nombre indica, el tablón de anuncios es público y visible para todos, ya sea mediante teléfono o interfaces web. Todos los mensajes

posteados en el tablón de anuncios son autenticados, y se asume que cualquier dato escrito en el tablón no puede ser borrado o alterado. En la práctica, implementar tal tablón de anuncios es uno de los aspectos ingenieriles de mayor reto de la votación criptográfica, ya que se debe preocupar de asuntos que van más allá de la corrupción de datos, como ataque de negación de servicio tanto para publicación de datos como para acceso a estos. Sin embargo, existen soluciones para resolver este problema.

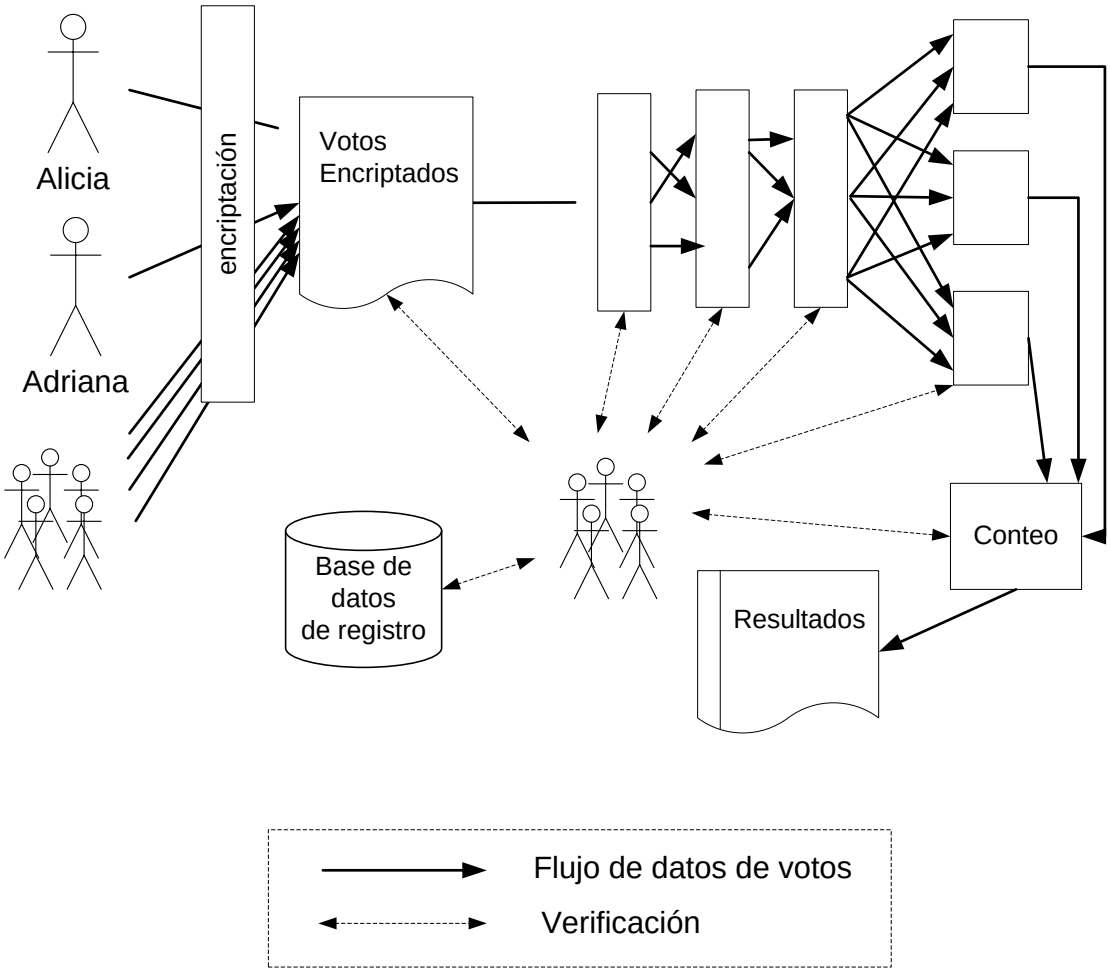

Figura 4. Votación criptográfica a alto nivel

En este tablón de anuncios, se publican los nombres o número de identificación de los votantes, para que cualquiera pueda decir quien ha votado y corroborar

que el votante es uno de los legítimos que aparece en el registro público. Junto con el nombre de cada votante, se publica su voto de forma encriptada, para que ningún observador pueda decir qué ha escogido los votantes. Así, dos procesos giran en torno al tablón de anuncios. El proceso de emitir voto permite a Alicia prepara su voto encriptado y publicarlo en el tablón de anuncios. Y, el proceso de conteo que involucra a los oficiales de las elecciones a que lleven a cabo diversas operaciones de suma de los votos encriptados y producir un conteo desencriptado, con pruebas de que el proceso es correcto que también se postean en el tablón de anuncios para que lo vean todos los observadores.

Efectivamente, el tablón de anuncios es el punto de transferencia verificable de votos identificables a no identificables. Primero los votos aparecen en el tablón de anuncios encriptados y junto a la identidad del votante. Después de múltiples transformaciones que llevan a cabo los oficiales de las elecciones, los votos acaban en el tablón de anuncios, desencriptados pero ahora desvinculados de la identidad del votante. Mientras los esquemas de votación clásica llevan a cabo una entrega completa y ciega –por ejemplo el voto se introduce en una caja–, la votación criptográfica realiza una entrega controlada, donde los votantes de forma individual pueden rastrear la entrada de su voto en el sistema, y cualquier observador puede verificar el procesado de esos votos encriptados en un conteo desencriptado. Este proceso se ilustra en Figura 4.

1.3.3. Un Recibo secreto de votante

Antes de que el voto encriptado de Alicia aparezca en el tablón de anuncios, debe prepararlo usando algún proceso que le de seguridad personal de que su voto ha sido encriptado correctamente. Con respecto a esto, el sistema no puede permitir que Alicia le transfiera a Carlos la misma seguridad, ya que esto le permitiría a Carlos influenciar en su decisión. Con este propósito, todos los sistemas de votación criptográficos requieren que los votantes vayan a un lugar de votación controlado y privado: es el único método conocido para que

establece una interacción verdaderamente privada que previene de coacciones al votante.

En mucho esquemas criptográficos, Alicia interacciona de forma privada con una máquina de votación. Ella escoge de la misma forma que los haría con una máquina de votación clásica, respondiendo a cada pregunta, verificando sus respuestas y finalmente confirmando su voto. Una vez hecho esto, la máquina procede a la encriptación del voto de Alicia, y empieza una interacción de verificabilidad con Alicia.

Este interacción es un tipo de prueba de conocimiento cero, donde la máquina prueba a Alicia que su voto encriptado de hecho corresponde con las elecciones hechas, sin revelar exactamente los detalles de esta correspondencia. En un esquema particularmente interesante, llamado Neff's MarkPledge, Alicia obtiene un recibo impreso que incluye el voto encriptado y algunos códigos de confirmación. Mediante la comparación de los códigos en el recibo con los que aparecen en la pantalla de la máquina, Alicia puede tener la certeza de que la máquina encriptó su voto de forma correcta. Además, como Alicia puede fácilmente proclamar, en un momento posterior, que un código de confirmación diferente apareció en la pantalla, ella no puede ser coaccionada. Este esquema se representa en la Figura 5.

Usando este recibo encriptado, Alicia puede verificar que su voto encriptado aparece correctamente en el tablón de anuncios. Dado que el voto está encriptado, ella puede dar incluso una copia de su recibo a las organizaciones políticas para que puedan verificar, en su nombre, la presencia de su voto encriptado en el tablón de anuncios.

Figura 5. Recibo secreto del esquema de verificación propuesto por Neff

Votación criptográfica basada en papel. En 2004, Chaum fue el primero en proponen un esquema criptográfico que usa votos de papel. El voto en cuestión se forma de manera especial: después de rellenar el voto, Alicia físicamente lo parte en dos mitades predeterminadas, destruye una, y emite la otra a la vez que hace una copia de esta misma mitad y se la lleva a casa como recibo. Este separación encripta de forma eficiente la elección de Alicia: sólo los oficiales de las elecciones con las llaves secretas adecuadas pueden recuperar la elección de Alicia durante el proceso de recuento.

En la mayoría de esquemas basados en papel, el voto en papel debe ser verificado antes de votar para asegurar que las dos mitades son consistentes la una con la otra. Sin esta verificación, un voto creado de forma fraudulenta podría corromper el registro apropiado de la intención del votante. En la última versión de Chaum, Punchscan, una segunda, verificación post-elecciones puede también verificar que el voto de Alicia es correcto. El artículo analizado en este trabajo [11] es una extensión del esquema de Chaum, con métodos que simplifican la fase de verificación pre-votación.

1.3.4. Recuento de votos

Una vez que todos los votos encriptados se postean en el tablón de anuncios, se procede al recuento. Cabe destacar que ningún oficial de las elecciones por sí solo puede desencriptar estos votos de forma individual: la clave secreta requerida para la desencriptación se comparte entre un número de oficiales de las elecciones, quien deben colaborar para cualquier operación de desencriptación. Estos es extremadamente importante, ya que cualquier desencriptación en este punto violaría el secretismo del voto de los votantes en cuestión. El proceso de desencriptación debe estar muy bien controlado para proteger la privacidad.

Existen principalmente dos técnicas para cumplir este propósito. La primera usa una forma especial de encriptación –llamada encriptación homomórfica– que habilita la suma de votos aunque estén encriptados, de manera que sólo el resultado final necesite desencriptación. La segunda usa una versión digital de "agitar la caja de votos", donde los votos de forma individual son barajados y barullados varias veces por diferentes partes, no relacionadas con la identidad del votante, y sólo después desencriptados.

Encriptación umbral de clave pública aleatoria. Antes de presentar un método de recuento, es importante denotar que todos los sistemas de votación criptográficos usan un tipo de encriptación especial denominada encriptación umbral de clave pública aleatoria. La propiedad de clave pública asegura que cualquiera puede encriptar usando una clave pública. La propiedad de umbral asegura que sólo un quórum de fiduciarios (mayor que el umbral), cada unos con su propia parte de la clave secreta, puede desencriptar.

Además, usando encriptación aleatoria, un único texto en claro, por ejemplo "Azul", puede ser encriptado de muchas maneras posible, dependiendo de la elección del valor aleatorio seleccionado cuando se encripta. Sin esta propiedad, dado que la mayoría de elecciones sólo ofrecen un número limitado de

opciones, por ejemplo “Rojo” o “Azul”, un atacante podría simplemente intentar encriptar de manera determinista todas las posibles opciones para descubrir, mediante la simple comparación del texto cifrado, qué votó cada persona. Con el valor aleatorio, un atacante tendría que intentar todos los posibles factores aleatorios. Seleccionando un criptosistema con un conjunto de valores aleatorios suficientemente gran, se asegura que esto nunca es posible.

Recuento bajo encriptación. Usando una forma especial de encriptación aleatoria de clave pública llamada encriptación de clave pública homomórfica, es posible combinar dos encriptaciones en una tercera encriptación cuyo calor esté relacionado con las dos originales, por ejemplo su suma. Usando clave pública, es posible coger una encriptación de x y una encriptación de y y obtener una encriptación de $x + y$, todo sin saber x ó y ó $x + y$.

En un esquema de votación homomórfico los votos se encriptan como 0 para “Azul” y 1 para “Rojo”. Entonces, todos los votos encriptados resultantes son sumados de forma homomórifca, uno por uno, para producir un única encriptación del número de votos para “Rojo”. Después, los fiduciarios pueden desencriptar el único valor encriptado para descubrir el recuento de votos “Rojo”. La diferencia entre el número de votos y el recuento de votos “Rojo” da el número de votos “Azul”. Esta propuesta se puede extender a más de dos opciones mediante la codificación de múltiples “contadores” dentro de un único texto cifrado. Además, una prueba de conocimiento cero se requiere para cada voto presentado, con el fin de asegurar que cada voto es verdaderamente la encriptación de 0 ó 1, y no, por ejemplo, 1000. Sino, un votante malicioso podría fácilmente desajustar la cuenta por mucho con un simple voto.

El voto homomórfico es particularmente interesante porque toda la operación homomórfica es verificable públicamente por cualquier observador, quien puede simplemente recalcularla usando sólo la clave pública. Los fiduciarios necesitan sólo desencriptar un recuento encriptado para cada carrera electoral.

Desafortunadamente, el voto homomórfico no soporta votos donde el candidato deseado no esté registrado: los contadores homomórficos encriptados deben ser asignados a los candidatos antes de que las elecciones empiecen.

Agitar la caja de votos virtual. Una forma diferente de recuento se puede alcanzar usando una *mixnet*, como describió Chaum. Mixnets típicamente usan un esquema de encriptación realeatorizable, el cual permite a cualquier coger una encriptación de un mensaje y producir una nueva encriptación del mismo mensaje con una aleatoriedad alterada. Estos es particularmente interesante poque, si alguien cogiera simplemente un conjunto de encriptaciones y las reordena, sería trivial comparar las encriptaciones barajadas con las encriptaciones sin barajar. Como las encriptaciones aleatorias son únicas, la propiedad de realeatorización es necesaria para realizar barajados indisinguibles.

En una *mixnet*, una secuencia de servidores mezcladores, cada uno normalmente operado por un partido político diferente, coge todos los votos encriptados del tablón de anuncios, los baraja y realeatoriza de acuerdo a un orden y a un conjunto de valores aleatorios mantenidos en secreto, y postea el conjunto de textos cifrados resultantes en el tablón de anuncios. El próximo servidor mezclador entonces realiza una operación similar, y así hasta el último servidor. Después, todos los fiduciarios cooperan para desencriptar las encriptaciones resultante, las cuales ya han sido desvinculadas de la identidad del votante correspondiente.

Es razonable confiar que al menos uno de los servidores agitará la caja de votos suficientemente bien para que la privacidad esté asegurada. Sin embargo, no es razonable confiar que ningún servidor remplazará un voto encriptado con un voto propio. En otras palabras, confiamos a los servidores mezcladores la privacidad del voto, pero no les confiamos que los votos sean correctos. Así, cada servidor mezclador debe proveer una prueba de conocimiento cero que

éste realizó una mezcla correcta, nunca removiendo, introduciendo, o cambiando los votos subyacentes. Estos tipos de pruebas son complicadas, pero un número eficiente de esquemas son conocidos e implementados.

Las votaciones basadas en *mixnet* son más difíciles de operar que las basadas en votaciones homomórficas, porque al re-encriptación y los procesos de barajado deben ser ejecutados bajo una base de confianza, manteniendo secretos los detalles del barajado del resto. Sin embargo, este tipo de votaciones presentan dos ventajas importantes: el conjunto completo de votos se preserva para la revisión de las elecciones, y soporta votos de forma libre donde no aparezca los candidatos oficiales. Como resultado, este tipo de esquemas de votación ofrecen las implementaciones más prometedoras para el mundo real incluso aunque son más complejos.

1.3.5. La promesa de la votación criptográfica

Con las votaciones criptográficas prometiendo la posibilidad de mejorar significativamente las auditorías de las elecciones, uno se puede cuestiones por qué las elecciones reales han rechazo estas técnica hasta la fecha. De hecho, ha sido recientemente cuando los esquemas de votación criptográfica se han convertido razonablemente usables por el votante medio. Hace falta más investigación para asegurar que los procesos adicionales de verificación del votante pueden realizarse en un entorno realista. Más importante todavía, un esfuerzo significativo de educación se requiere porque el poder de unas elecciones verificadas criptográficamente está muy alejado de lo intuitivo.

2. Conceptos preliminares criptográficos

Protocolos para unas elecciones democráticas dependen de un número de bloques criptográficos que sirven de base. En este capítulo se revisan los conceptos y la notación de estos bloques base. Se empieza con un pequeña revisión de la criptografía de clave pública, sus definiciones de seguridad, y los principales algoritmo que se usan en protocolos prácticos. Se revisan los criptosistemas homomórficos, las propiedades interesantes que tiene y las consecuencias de seguridad de dichas propiedades. Después, se consideran los criptosistemas umbrales, donde el proceso de generación de clave y desencriptación puede ser distribuido entre fiduciarios, una tarea de gran importancia en sistemas de votación. También se revisan pruebas de conocimiento cero, otro componente crítico de una votación verificable de manera universal, y brevemente se describe un programa de ofuscación, que es de particular importancia en el artículo en cuestión [11]. También se cubre la composición universal, un marco para comprobar que los protocolos son seguros.

2.1. NOTACIÓN

Se denota con κ el principal parámetro de seguridad y se dice que una función $\epsilon(\cdot)$ es negligible si para cada constante c existe una constante κ_0 tal que $\epsilon(\kappa) < \kappa^{-c}$ para $\kappa > \kappa_0$. Se denota por PT PPT, y PT*, el conjunto de tiempos polinómicos uniformes, tiempos polinómicos uniformes probabilísticos, y tiempos polinómicos no uniformes de máquinas Turing respectivamente. Se usa la notación $\overset{R}{\leftarrow}$ para denotar o un muestreo uniforme y aleatorio de un conjunto o distribución, o la asignación de un proceso aleatorio con elección uniforme de valores aleatorios.

2.2. ENCRIPTACIÓN DE CLAVE PÚBLICA

La encriptación de clave pública fue sugerida en primer lugar por Diffie y Helman en 1976 [9], y implementada en primer lugar por Rivest, Shamir y Adleman en 1977. En su núcleo, es una idea simple aunque en cierto modo contra la intuición: cualquiera puede encriptar un mensaje destinado a Alicia, pero sólo Alicia puede desencriptarlo. Siendo más precisos, Alicia puede generar un par de claves compuestas por una clave pública *pk* y una clave secreta *sk*. Después ella distribuye *pk*, pero se guarda *sk* para ella misma. Usando *pk*, Blas puede encriptar un texto en claro *m* en un texto cifrado *c*. El texto cifrado *c* se envía a Alicia, que es la única que puede descífralo ya que es la única que posee la clave privada *sk*.

De manera más formal, se puede definir un criptosistema de clave pública como sigue:

Un criptosistema de clave pública PKCS es un conjunto de tres algoritmos PPT $\mathcal{D}, \mathcal{G}, \mathcal{E}$, tales quem dados un parámetro de seguridad κ, se definen las siguientes operaciones:

- **Generación de un par de claves:** claves pública y privada puede ser generadas por cualquiera usando el algoritmo público $\mathcal{G}$.

$$(pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^k)$$

- **Encriptación:** un mensaje llano *m* en el espacio de mensaje M_{pk} puede encriptarse usando la clave pública *pk* y el algoritmo de encriptación $\mathcal{E}$. Este proceso es normalmente aleatorizado, usando el valor aleatorio $r \in R_{pk}$.

$$c = \mathcal{E}_{pk}(m; r)$$

  Se denota C_{pk} como el espacio de texto cifrado *c*.

- **Desencriptación:** un texto cifrado *c* en el espacio de texto cifrados C_{pk} puede ser desencriptado usando la clave secreta *sk* y el algoritmo de desencriptación $\mathcal{D}$. Este proceso es siempre determinístico: dado un texto cifrado siempre se desencripta dando lugar al mismo texto en claro bajo una clave secreta dada.

$$m = \mathcal{D}_{sk}(c)$$

  Dado tal criptosistema, se pueden considerar diferentes definiciones de seguridad.

2.2.1. Seguridad IND-CPA

Intuitivamente, se dice que un criptosistema es semánticamente seguro si, dado un texto cifrado *c*, un adversario no puede determinar ninguna propiedad del texto en claro subyacente *m*. En otras palabras, un adversario no puede extraer ninguna información semántica del texto en claro *m* desde un encriptación de *m*. La seguridad semántica fue definida por primera vez en 1982 por Goldwasser y Micali, quien también mostró que la seguridad semántica es equivalente a textos cifrados indistinguibles dados unos textos en claro escogidos. Esta última definición es conocida como seguridad GM o IND-CPA.

En esta definición, dado una clave pública *pk*¸ el adversario escoge entre dos texto en claro m_0 y m_1 y después es usando en *c*, un texto cifrado de uno de estos texto en claro, escogidos aleatoriamente. Si el adversario no puede adivinar cuál de los dos textos en claro fue escogido para la encriptación con una probabilidad mayor al 50%, entonces se dice que el esquema es seguro contra el ataque de elección de texto en claro.

Así se puede definir formalmente la Seguridad IND-CPA, como dado un criptosistema de clave pública PKCS = $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ se dice que es IND-CPA-seguro si existe una función negligible *v(·)* tal que, para todo $Adv \in PT^*$:

$$\Pr\left[(pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^k); (m_0, m_1, estado) \leftarrow Adv(escoge, pk);\right.$$

$$b \overset{R}{\leftarrow} \{0,1\}; c \overset{R}{\leftarrow} \mathcal{E}_{pk}(m_b); b' \leftarrow Adv(adivina, c, estado) \quad :$$

$$\left. b = b' \right] < \frac{1}{2} + v(\kappa)$$

Se conocen una serie de esquemas eficientes que son IND-CPA-seguros.

RSA con relleno OAEP. En el también conocidos como "RSA puro", se seleccionan dos números primos seguros *p* y *q*, *pk* = (*n*,*e*) donde *n* = *pq* y $e \wedge \phi(n)$, y *sk* = *d* donde *ed* = 1 mod ϕ(*n*). La encriptación se realiza como *c* = m^e mod *n*, y la desencriptación como *m* = c^d mod *n*. Claramente, dado que la operación de encriptado es determinística dado *m* y *pk* , el "RSA puro" no es IND-CPA-seguro: un adversario puede encriptar m_0 y m_1 y compararlos con el texto cifrado.

RSA puede ser hacerse IND-CPA-seguro usando relleno de mensajes como OAEP. En vez de encriptar el mensaje *m* tal cual, RSA-OAEP encripta *m*||*OAEP(m),* donde *OAEP*(*m*) incluye aleatoridad.

El Gamal. El Gamal [10] es el primer ejemplo de un criptosistema IND-CPA-seguro. Considerando *g* el generador de un subgrupo de oren *q* de Z_p^* donde *p* es primo y *q* es un factor de *p-1* y primo grande. La generación de claves envuelve seleccionar un *x* ∈ Z_p^*, en cuyo punto *sk* = *x* y *pk* = *y* = g^x mod *p*. Entonces la encriptación viene dada por:

$$c = (\alpha, \beta) = (g^r, m \cdot y^r), r \overset{R}{\leftarrow} Z_p^*$$

Y la desencriptación se realiza como:

$$m = \frac{\beta}{a^x}$$

Paillier. Paillier es otro buen ejemplo de un criptosistema IND-CPA-seguro. Considerando *n* = *pq* en el entorno RSA. Considerando $\lambda = mcm(p-1, q-1)$ y la función $L(x) = (x-1)/n$. Considerar un generado *g* de $Z^*_{n^2}$, formado de tal manera que *g* = 1 mod *n*. La clave pública entonces es simplemente *n*, mientras que la clave privada es *λ*. El encriptado de $m \in Z_n$ se realiza como $c = g^m r^n \bmod n^2$ para un $r \in Z_n$ aleatorio. De la misma forma, el desencriptado se realiza como:

$$m = \frac{L(c^{\lambda} mod\ n^2)}{L(cg^{\lambda} mod\ n^2)}\ mod\ n$$

Como este criptosistema es de especial importancia durante el artículo [11], se proporciona una breve explicación de éste. Recordando que:

- $\phi(n) = (p-1)(q-1)$ es la función de Euler.
- $\lambda = mcm(p-1, q-1)$ es el resultado de la función de Carmichael sobre *n*.
- El orden de $Z^*_{n^2}$ es $n\phi(n)$.
- Para cualquier a ∈ $Z^*_{n^2}$:
    - $a^{\lambda} = 1\ mod\ n, \quad a^{\lambda n} = 1\ mod\ n^2$

Así, considerando la función para desencriptar definida anteriormente, en particular el denominador. Dado que *g* = 1 mod *n*, que puede ser visto como *g* = $n\alpha$ + 1 para algún entero α.

$$L\left(g^{\lambda} mod\ n^2\right) = \frac{\left((1+n\alpha)^{\lambda} mod\ n^2\right) - 1}{n}$$

$$= \frac{(n\alpha\lambda) mod\ n^2}{n}$$

$$= \alpha\lambda \; mod \; n^2$$

Cabe notar que la exponenciación anterior se reduce a la multiplicación porque todos los otros monomios en la expansión son múltiplos de n^2. Esto se ve fácilmente porque r^n cancelará por la exponenciación a λ.

$$\mathrm{L}\left(\mathrm{c}^{\lambda} \mathrm{mod}\ \mathrm{n}^2\right) = \alpha\lambda \bmod \mathrm{n}^2$$

Y de esta forma el descencriptado funcionar como se ha especificado.

2.2.2. Seguridad IND-CCA

Que no se pueda distinguir respecto a los textos en claro escogidos por los adversarios no es suficiente para todas las aplicaciones. De forma intuitiva, se debería considerar la posibilidad de que el adversario pudiera obtener el desencriptado de unos pocos textos cifrados escogidos antes de recibir el texto cifrado en cuestión. Esta noción de seguridad se denomina IND-CCA, conocida informalmente como “seguridad contra ataques durante la comida”. El modelo es tal que el adversario podría tener acceso a una caja de desencriptado mientras el propietario está “fuera comiendo” (posiblemente metafóricamente). Más tarde, el adversario intentará usar la información obtenida durante la comida para desencriptar otros textos cifrados.

Así, un criptosistema de clave pública PKCS = $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ se dice que es IND-CCA-seguro si existe una función negligible $v(\cdot)$ tal que, para todo $Adv \in PT^*$ dado un oráculo de desencriptado ODec(·):

$$\Pr\left[(pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^k); (m_0, m_1, estado) \leftarrow Adv^{ODec_{sk}(\cdot)}(escoge, pk);\right.$$

$$b \overset{R}{\leftarrow} \{0,1\}; c \overset{R}{\leftarrow} \mathcal{E}_{pk}(m_b); b' \leftarrow Adv(adivina, c, estado):$$

$$\left. b = b' \right] < \frac{1}{2} + v(\kappa)$$

Sin embargo, esta versión de seguridad IND-CCA no es tan interesante como su variante, la seguridad IND-CCA2.

2.2.3. Seguridad IND-CCA2

La definición de IND-CCA2 da al atacante que ataca durante la hora de la comida incluso más poder: después de que el texto cifrado en cuestión se ha transmitido, el atacante tiene acceso al oráculo de desencriptación ODec(·), al cual el atacante puede preguntar sobre cualquier cosa, excepto sobre el texto cifrado en cuestión, incluso puede pedir la desencriptación de cualquier otro texto cifrado, incluso aquéllos derivadores del texto cifrado en cuestión.

Así, un criptosistema de clave pública PKCS = $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ se dice que es IND-CCA-seguro si existe una función negligible *v(·)* tal que, para todo $Adv \in PT^*$ dado un oráculo de desencriptado ODec(·):

$$\Pr\left[(pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^k); (m_0, m_1, estado) \leftarrow Adv^{ODec_{sk}(\cdot)}(escoge, pk);\right.$$

$$b \overset{R}{\leftarrow} \{0,1\}; c \overset{R}{\leftarrow} \mathcal{E}_{pk}(m_b); b' \leftarrow Adv^{ODec_{sk}(\cdot)}(adivina, c, estado):$$

$$\left. b = b' \right] < \frac{1}{2} + v(\kappa)$$

Donde $ODec_{sk}(\cdot)$ es un oráculo de desencriptación que responde a todas las peticiones, excepto aquéllas relativas al texto cifrado *c* a las cuáles responderá con NULL.

La seguridad IND-CCA2 se considera el estándar de oro de los criptosistemas de clave pública (aunque, en algunos casos, el estándar alternativo de conocimiento de texto en claro también se considera). Efectivamente, la única manera conocida de obtener un texto cifrado nuevo para un mensaje dado es encriptar el texto en claro uno mismo.

2.2.4. Seguridad IND-RCCA

Para alguna aplicaciones sucede que la seguridad IND-CCA2 puede ser exagera. En particular, dado un texto cifrado *c*, puede ser aceptable dejar a cualquier crear un nuevo texto cifrado *c'* tal que $\mathcal{D}_{sk}(c) = \mathcal{D}_{sk}(c')$. Así, cualquier que no tenga la clave *sk* seguiría siendo incapaz de generar un texto cifrado cuyo texto en claro estuviere relacionado de alguna manera con *c* diferente de la igualdad.

De esta forma, existe una posición intermedia entre la seguridad IND-CPA y IND-CCA: la seguridad IND-RCCA, la cual permite específicamente al adversario generar un texto cifrado *c'* "nuevo" de un texto cifrado *c*, tal que $\mathcal{D}_{sk}(c) = \mathcal{D}_{sk}(c')$.

De manera formal, un criptosistema de clave pública PKCS = $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ se dice que es IND-RCCA-seguro si existe una función negligible *v(·)* tal que, para todo $Adv \in PT^*$ dado un oráculo de desencriptado ODec(·):

$$\Pr\left[(pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^k); (m_0, m_1) \leftarrow Adv^{ODec_{sk}(\cdot)}(escoge, pk);\right.$$

$$b \overset{R}{\leftarrow} \{0,1\}; c \overset{R}{\leftarrow} \mathcal{E}_{pk}(m_b); b' \leftarrow Adv^{ODec_{sk,m0,m1}(\cdot)}(adivina, c,) :$$

$$\left. b = b' \right] < \frac{1}{2} + v(\kappa)$$

Donde $ODec_{sk,m0,m1}(\cdot)$ es un oráculo de desencriptación que responde a todas las peticiones, excepto para texto cifrados que se desenciptan a m_0 o m_1.

2.3. ENCRIPTACIÓN DE CLAVE PÚBLICA HOMOMÓRFICA

Los criptosistemas de clave pública homomórficos exhiben una propiedad algebraica de especial interés: cuando dos textos cifrados se combinan de una manera específica y computable públicamente, el texto cifrado resultante

codifica la combinación de textos en claro subyacentes bajo una operación específica de grupo, normalmente suma o multiplicación,

Así, un criptosistema de clave pública PKCS = $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ se dice que es homomórfico para relaciones binarias $(\oplus, \otimes)$ si:

- $\forall (pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^k)$,
  $dado\ un\ mensaje\ en\ dominio\ M_{pk}, \left(M_{pk}, \oplus\right)\ forma\ un\ grupo.$

- $\forall (pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^k)$,
  $dado\ un\ rango\ de\ texto\ cifrado\ C_{pk}, \left(C_{pk}, \otimes\right)\ forma\ un\ grupo.$

- $\forall (pk, sk) \overset{R}{\leftarrow} \mathcal{G}(1^k),\ \forall (c_1, c_2) \in C^2_{PKCS,pk}$,

$$\mathcal{D}_{sk}(c_1 \otimes c_2) = \mathcal{D}_{sk}(c_1) \oplus \mathcal{D}_{sk}(c_2)$$

2.3.1. Re-encriptación

Una consecuencia inmediata de la propiedad de un sistema homomórfico es su capacidad para realizar reencriptación: dado un texto cifrado *c*, cualquiera puede crear un texto cifrado diferente *c'* que codifica el mismo texto plano como *c*. Cabe recordar que PKCS es homomórfico para $(\oplus, \otimes)$ si $\left(M_{pk}, \oplus\right)$ forma un grupo, lo que significa que existe un texto en claro identidad m0 tal que, $\forall m \in M_{pk}, m \oplus m_0 = m$. Por lo tanto, dado un criptosistema homomórfico PKCS, se puede definir un algoritmo de reencriptación como:

$$\mathcal{RE}_{pk}(c; r) = c \otimes \mathcal{E}_{pk}(m_0; r)$$

Si $\mathcal{D}_{sk}(c) = m$, entonces $\mathcal{D}_{sk}\left(\mathcal{RE}_{pk}(c)\right) = m$.

2.3.2. Criptosistemas de seguridad homomórfica

La maleabilidad de textos cifrados en criptosistemas homomórficos limita la seguridad de tales esquemas. En particular, la capacidad de reencriptar de forma inmediata indica que el sistema no es IND-CCA2-seguro, y puede ser como mucho IND-RCCA-seguro. Incluso más significativo, la capacidad de crear un texto cifrado de diferentes pero relacionados textos en claro rompe incluso la seguridad IND-RCCA. Específicamente, un adversario puede coger el texto cifrado en cuestión *c*, crear $c' = c \otimes \mathcal{E}_{pk}(\tilde{m})$ para algún $\tilde{m}$ conocido por el adversario, preguntar a ODec con *c'* y obtener *m'*, y calcular $m = m' \oplus \tilde{m}^{-1}$.

No se ha llegado a saber todavía si los esquemas homomórfico pueden llegar a ser IND-CCA-seguros: ¿puede ODec ayudar al éxito de un adversario si sólo puede usarse antes de que el texto cifrado bajo análisis sea creado? Así, conocemos que un criptosistema homomórfico puede ser IND-CPA-seguro, pero no conocemos si puede ser IND-CCA-seguro.

2.3.3. Esquemas homomórfica en la práctica

Un número de esquemas prácticos son homomórficos.

RSA. En RSA puro, la encriptación se realiza como $c = m^e \bmod n$. Así, claramente $c_0 \times c_1 = (m_0 \times m_1)^e mod\ n$. Por lo tanto, RSA puto es homomórfico en la operaciones (×,×). No obstante, cabe recordar que RSA puro no es ni IND-CPA-seguro, lo que significa que es inútil para muchas aplicaciones. Por otra parte, RSA-OAEP, es bastante útil, pero pierde la propiedad homomórfica debido a la no maleabilidad del relleno OAEP.

El Gamal. En El Gamal, se encripta como $c = (g^r, m \cdot y^r)$. Por lo tanto, si se define ⊗ producto elemento a elemento de parejas de textos cifrados, entonces El Gamal es homomórfico para (×,⊗):

$$(g^{r1}, m_1 \cdot y^{r1}) \otimes (g^{r2}, m_2 \cdot y^{r2}) = (g^{r1+r2}, m_1 m_2 \cdot y^{r1+r2})$$

EL Gamal es particularmente interesante porque exhibe homomorfismo y es IND-CPA-seguro.

El Gamal exponencial. Si El Gamal es homomórfico para la multiplicación de un texto en claro en la base, entonces uno se ve tentado inmediatamente a adaptar El Gamal para usar un texto en claro en el exponente para exhibir una suma homomórfica. De hecho, esto puede hacer, pero a un alto coste: el desencriptado requiere realizar un logaritmo discreto, el cual de forma inherente limita el dominio M del texto en claro a tamaño polinómico. El Gamal exponencial se define como:

- **Generación de clave:** la misma que en El Gamal, seleccionar un primo *p* tal que otra primo grande *q* divide (*p* – 1). Seleccionar *g,* un generador del subgrupo Z_p^* de orden q. $M_{pk} = Z_q$. *sk* = *x*, donde *x* se escoge aleatoriamente de Z_q^*. *pk* = *y* = g^x mod *p*.

- **Encriptación:** parecida a la de El Gamal, excepto que ahora el texto en claro está en el exponente.

$$\mathcal{E}_{pk}(m;r) = (\alpha,\beta) = (g^r, g^m \cdot y^r) mod\ p$$

- **Desencriptado:** similar al de El Gamal, excepto que ahora se necesita un logaritmo discreto.

$$\mathcal{D}_{sk}(\alpha,\beta) = log_g \left[\frac{\beta}{\alpha^x}\right] mod\ p$$

- **Suma homomórfica:** exactamente la misma que la multiplicación homomórfica de El Gamal, usando una operación de cifrado que realiza la multiplicación elemento a elemento sobre las parejas de texto cifrados.

$$\mathcal{E}_{pk}(m_1; r_1) \otimes \mathcal{E}_{pk}(m_2; r_2) = (g^{r1}, g^{m1} \cdot y^{r1}) \otimes (g^{r2}, g^{m2} \cdot y^{r2})$$

$$= (g^{r1+r2}, g^{m1+m2} \cdot y^{r1+r2})$$

$$= \mathcal{E}_{pk}(m_1 + m_2; r_1 + r_2)$$

En práctica, la desencriptación limita el dominio del mensaje, por ejemplo unos pocos trillones de posibles mensajes como mucho. La desencriptación se realiza normalmente por medio de varios algoritmos basado en logaritmos discretos, por ejemplo el algoritmo "paso de bebe paso de gigante", que requiere un tiempo $O(\sqrt{m})$.

Paillier. En Paillier, la encriptación se realiza como $c = g^m r^n \bmod n^2$. Claramente, este esquema es homomórfico para $(+,\times)$ sobre un espacio de textos en claro Z_n:

$$\mathcal{E}_{pk}(m_1; r_1) \times \mathcal{E}_{pk}(m_2; r_2) = (g^{m1}, r_1^n) \times (g^{m2}, r_2^n)$$

$$= g^{m1+m2}(r_1 r_2)^n$$

$$= \mathcal{E}_{pk}(m_1 + m_2; r_1 r_2).$$

Cabe destacar que la descripción de Paillier es eficiente, lo que significa que el dominio de textos en claro puede ser superpolinómica mientras retiene el homomorfismo aditivo.

Paillier Generalizado: Damgård generalizó el esquema de Paillier, para que una clave pública con módulo *n* pueda encriptar textos en claro en Z_{n^s} en textos cifrados en $Z_{n^{s+1}}$. Así, cálculos en módulo n^2 se reemplazan por cálculos en módulos n^{s+1}. De tal forma, para esta versión generalizada escribimos:

$$\mathcal{E}_{n,s}^{pai}(m) = g^m r^{n^s} mod n^{s+1}$$

y se usa M_{n^s} y C_{n^s} para denotar el espacio de mensajes correspondientes en Z_{n^s} y el espacio de textos cifrados $Z^*_{n^{s+1}}$. Damgård probó que la seguridad del esquema generalizado sigue la del esquema original. Las propiedades del criptosistema de Paillier extendido pueden ser vistas como una serie de criptosistemas de Paillier típicos en paralelo:

- El orden del grupo $Z_{n^{s+1}}$ es $\phi(n)n^s$.
- El orden de (n + 1) es n^s en $Z_{n^{s+1}}$, así se usa g = n + 1 como la base para la encriptación: su exponente está perfectamente dimensionados para textos llanos en Z_{n^s}.
- Si se denota $s = r^{n^s}$, entonces: $s^{\phi(n)} = 1\, mod\, n^{s+1}$ y $s^{\lambda} = 1\, mod\, n^{s+1}$. Así, cuando se calcula c^{λ} durante la desencriptación, el valor aleatorio se cancela y se obtiene:

$$c^{\lambda} = g^{m\lambda} mod\, n^{s+1}$$

El Paillier generalizado proporciona dos proporciona dos propiedades interesantes de alto nivel:

1. *Largos textos en claro con mejor eficiencia:* usando la misma clave pública, *n*, textos en claro más largos que |*n*| pueden ser encriptados, mientras que la sobrecarga del texto cifrado permanece |*n*|.
2. *Encriptación por capas:* los textos en claro pueden ser encriptados varias veces bajo la misma clave pública, cada vez añadiendo una sobrecarga de |*n*| bits. Estos es interesante en el hecho de que los órdenes del grupo mantienen todas las propiedades homomórficas intactas en cada capa.

2.4. CRIPTOSISTEMAS DE CLAVE PÚBLICA UMBRAL

En muchas aplicaciones, incluyendo las votaciones, es deseable permitir el desencriptado sólo cuando un quórum de "fiduciarios" está de acuerdo. Con otras palabras, la clave secreta *sk* no está disponible a una única parte. A cambio, *l* fiduciarios compartes *sk*: fiduciario *i* posee la parte $sk^{(i)}$. Si un mínimo *k* de los *l* fiduciarios participan, entonces se puede desencriptar. Si menos de *k* fiduciarios participan, entonces las propiedad de seguridad del criptosistema se preservan completamente.

Existen dos propuestas para generar las partes compartidas $\{sk^{(i)}\}$. La propuesta más simple es para el "repartidor" es generar (*pk, sk*) de forma normal, partir *sk* en partes, y después distribuir estas partes entre los fiduciarios apropiados. Una propuesta más seguro es hacer que los fiduciarios todos juntos generen el par de claves, de manera que ninguna parte conocerá nunca la clave secreta *sk* de forma completa durante el proceso.

2.4.1. Compartición de secretos

El concepto de compartición de secretos es crítico para la implementación de criptografía umbral, introducida por Shamir en 1979. En este esquema, un secreto *s* en un cuerpo finito se comparte como $s^{(1)}$, $s^{(2)}$, …, $s^{(l)}$, donde cualquier subconjunto de tamaño *k* de estas *n* partes revela *s* $(k \leq l)$, pero ningún subconjunto de tamaño más pequeño que *k* no revela nada de *s*. La implementación de Shamir produce un polinomio *P(x)* de grado *k* – 1 sobre el cuerpo finito en cuestión, tal que *P*(0) = *s*, y cada parte $s^{(i)}$ es un punto (*x, y*) tal que *y* = *P*(*x*). Usando los coeficientes de Lagrange para interpolación polinómica, *k* puntos son suficientes para recuperar el polinomio *P*, y así *s* = *P*(0). No obstante, menos de *k* puntos, esconderán *s* como explica la teoría del información.

Cada recordar que este proceso de interpolación define polinomios de interpolación de Lagrange para cada punto. Dado $\{(x_i), (y_i)\}_{i\in[1,k]}$ los *k* puntos que queremos interpolar, se denotan como $\lambda_i(x)$ la interpolación polinómica que corresponde al punto *i*:

$$\lambda_i(x) = \prod_{\mathrm{j}=1,\mathrm{j}\neq\mathrm{i}}^{\mathrm{k}} \frac{(\mathrm{x} - \mathrm{x_j})}{(\mathrm{x_i} - \mathrm{x_j})}$$

Entonces el polinomio interpolado es:

$$P(x) = \sum_{i=1}^{k} \lambda_i(x) y_i$$

Dado que sólo buscamos *P*(0), el secreto, podemos saltarnos la computación de los coeficientes del polinomio real, e ir directamente a:

$$s = P(0) = \sum_{i=1}^{k} y_i \left(\prod_{\mathrm{j}=1,\mathrm{j}\neq\mathrm{i}}^{\mathrm{k}} \frac{-\mathrm{x_j}}{(\mathrm{x_i} - \mathrm{x_j})} \right)$$

Cabe destacar que cada parte es la pareja (*x, y*). Así, se puede permitir que todas las x_i sean públicas, permaneciendo las correspondientes y_i secretas. Esto permite que cualquiera pueda computar los coeficientes de Lagrange, listos para ser combinados con los valores y_i reales en el momento adecuado.

2.4.2. Cálculo seguro multipartito

En 1986, Yao mostró que cualquier cálculo en el que intervienen múltiples entidades (multipartito) puede realizarse usando representación enrevesada de un circuito que implementa el cálculo. La técnica de Yao implica una descomposición puerta-a-puerta y bit-a-bit del cálculo. Así, aunque es un método increíblemente poderoso como método genérico, también es bastante ineficiente a la práctica.

Claramente, los criptosistemas umbrales pueden implementar usando un esquema simple de compartición de secretos de Shamir y dos circuitos enrevesados: uno que genera, parte, y distribuye la pareja de claves (*pk, sk*) a todos los fiduciarios, y otro que combina las partes para realizar la desencriptación real. En la práctica, sin embargo, es mejor encontrar un criptosistema que suporte explícitamente algún mecanismo eficiente para operaciones umbrales.

2.4.3. Esquema eficiente umbrales

El Gamal. La estructura algebraica de El Gamal lo hace particularmente útil para operaciones umbrales, como se describió en primer lugar en este entorno por Desmedt y Frankel. Considerar la compartición de las partes la clave secreta *x* ($x_1, x_2, \ldots, x_l$) usando el método de Shamir. Cada parte x_i se asocia con una parte de la clave pública, $y_i = g^{x_i}$. Destacar que *x* y *y* no deben confundirse con las coordenadas de los puntos usando el esquema de Shamir: aquí estos valores son la clave privada y pública, respectivamente.

Entonces, la desencriptación en este esquema se hace:

$$m = \frac{\beta}{\prod \alpha^{x_i \lambda_i(0)}}$$

donde α^{x_i} puede ser calculado por cada fiduciario de manera independiente.

Más importante todavía es que entonces es posible alcanzar la generación de claves de forma distribuida, cuando ninguna parte, ni siquiera el "repartidor", aprende la llave secreta completamente. El protocolo funciona como sigue, a alto nivel:

1. Cada fiduciario genera un parte secreta $x_i \in Z_q^*$, y publica $y_i = g^{x_i}$.

2. Cada fiduciario comparte sus partes x_i con todos los demás participantes usando el esquema de verificación de compartición de secreto *k-sobre-l*, para que ninguna entidad tramposa pueda captar las partes.

3. Entonces, cada subconjunto *k* de fiduciarios puede realizar las *lk* operaciones para desencriptar un texto cifrado de El Gamal con dos capas de acciones umbrales, una capa para reconstituir las acciones de cada $x_{i,}$ y otra capa para reconstituir las acciones de las x_i en las acciones del secreto total *x*.

RSA. Obtener las propiedades umbrales de RSA es más exigente, en particular con respecto a la generación de clave, lo que requiere que el producto de dos primos sea obtenido sin que ninguna parte conozca estos dos primos. Esquemas eficientes son conocidos para la desencriptación umbral. Además, también se conocen esquemas más eficientes que el cálculo genérico multipartito, aunque son un poco más lentos para aquellos sistemas basados en logaritmos discretos como El Gamal. A alto nivel, estos protocolos varios definen las siguientes operaciones:

- **Desencriptado:** parecido a la configuración de El Gamal, la ideal general del desencriptado umbral de RSA es compartir un exponente secreto, en este caso el exponente privado *d*, para luego usarlo en la interpolación de Lagrange para recombinar *k* de esos *l*. El trabajo precio asumía que la interpolación era demasiado difícil para realizarla sobre Z_n^*, ya que el orden del grupo es desconocido, y escogía realizar interpolación polinómica sobre un subgrupo de Z_n^*. Shoup mostró que, si los factores primos del módulo de RSA son “primos seguros” –por ejemplo cada unos es el doble de otro primo más 1– entonces la interpolación polinómica es, de hecho, posible sobre Z_n^*.

- **Generación de claves:** Boneh y Franklin implementaron por primera vez un esquema relativamente eficiente para de la distribución de claves de

RSA, tal que ninguna entidad aprende la factorización: cada fiduciario obtiene una parte del exponente secreto *d*, siendo en este punto posible la desencriptación umbral como se explico antes.

Paillier. Paillier usa la teoría de números de forma similar a RSA, pero el proceso de desencriptación no es exactamente el mismo. Fouque mostró como aplicar el método de Shoup de la desencriptación umbral de RSA a Paillier. Damgård mostró un método relacionado que también aplica a la generalización del Paillier.

3. Marcar & Votar

Este apartado cobre el trabajo aparecido en el *Workshop on Privacy in the Electronic Society*, en octubre de 2006, llevado a cabo por Ronald L. Rivest.

3.1. INTRODUCCIÓN

La criptografía permite juntar el secretismo del voto y la revisión pública de las votaciones. Los votos se encriptan y postean en un tablón de anuncios público, junto con el nombre del votante en texto en claro. Cualquiera puede ver que Alicia ha votado, aunque, por supuesto, no puede ver qué ha votado. Los votos encriptados son anónimos y se cuentan usando técnicas que son verificables públicamente.

La mayoría de los esquemas de votación requieren de un equipamiento complejo y de auditorías. Cierto grado de complejidad es inevitable, ya que el objetivo del voto criptográfico es hacer unas elecciones de forma correcta mientras se confía en terceras partes los mínimos posible. Desafortunadamente, esta complejidad a menudo radica en el modo de la adopción. Si se requiere de una experiencia significativa para entender cómo funciona el sistema de votación, y si la operatividad del sistema es particularmente compleja, los oficiales de las elecciones y el público pueden resultar reticentes a su adopción. Entonces, la pregunta es ¿cuánto se pueden simplificar el proceso de voto mientras se mantienen la verificabilidad criptográfica?

Sistemas de voto y superficies de marcado. La propuesta de los autores del artículo bajo estudio proponen el método “Marcar y Votar” [11], que usa superficies de marcado, aunque no debe ser confundido con un juego de probabilidad. Los autores esperan, que dado que el público ya está familiarizado

con superficies para marcar, el uso de su algoritmo servirá como chispa para extender el tema de la votación criptográfica.

3.1.1. Marcar & Votar

Rivest propone el método “Marcar & Votar” (M&V), que es un método de votación criptográfica que proporciona la posibilidad de revisión de las elecciones de manera pública usando tecnología simple.

1. **Votos de papel:** la emisión del voto se basa completamente en el uso de papel y bolígrafo.

2. **Auditoría auto-contenida en el voto:** los votos contienen toda la información necesaria para ser revisados, no hay necesidad de interacciones con los oficiales de las elecciones.

3. **Escrutinio simple:** los votos se recuenta usando contadores homomórficos encriptados en vez de *mixnets*. Cualquiera puede fácilmente verificar el recuento final, y los oficiales de las elecciones sólo necesitan cooperar para desencriptar un solo texto cifrado por vuelta.

Lo que tiene que hacer el votante es simple:

- **Registrarse:** Alicia se registra y obtiene un voto con un orden de candidatos aleatorio. Los oficiales de las elecciones no deberían ver este orden de candidatos. El votó se perfora a lo largo de la mitad vertical, con los nombres de los candidatos en la mitad izquierda y los correspondientes círculos de escaneo óptico a la derecha. Un código de barras 2D se posiciona justo debajo de casillas de selección de la derecha. Una superficie de marcado etiquetado como “vacío si marcado” se pone justo debajo del código de barras, y una perforación adicional separa la superficie de marcado del resto de la mitad derecha. (véase Figura 6).

- **Revisión [opcional]:** Alicia puede seleccionar un segundo voto para revisión. Ella marca la superficie de marcado, entrega el voto vacío a un ayudante (por ejemplo a un partido político o una organización activista en la que confíe) y recibe confirmación de que el voto está bien hecho1. Esto le da a Alicia la confianza de que su primer voto también está bien hecho: si suficiente votantes realizan la revisión, incluso un puñado de votos malos serás detectados (véase Figura 7).

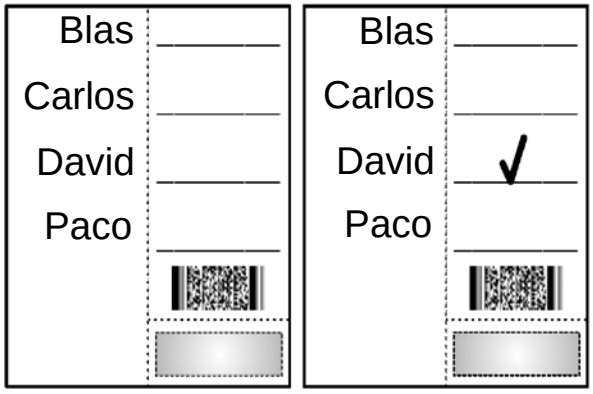

Figura 6. Voto tipo "Marca&Vota", antes y después de elegir

- **Marcar la selección:** Alicia entra en la cabina de aislamiento para hacer y revisar su selección.

Figura 7. Voto "Marca&Vota" de revisión

- **Separar las mitades del voto:** Alicia separa las dos mitades del voto. Un receptáculo está disponible para que ella deseche la mitad izquierda del voto. Cabe destacar que la mitad descartada no lleva información sobre la identidad del votante, sólo un orden aleatorio de candidatos (véase Figura 8).

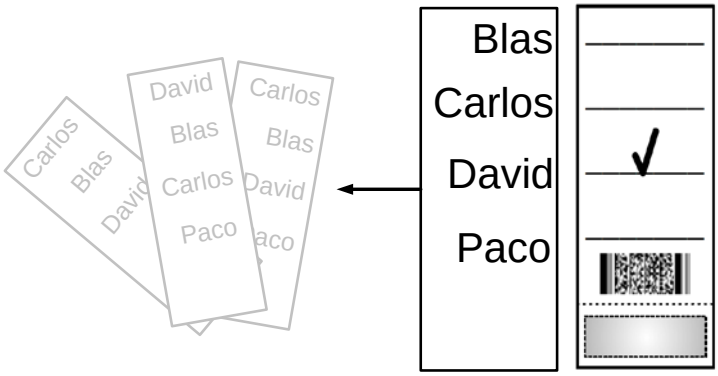

Figura 8. Separación del Voto "Marca&Vota"

- **Emitir:** Alicia presenta la mitad derecha de su voto a un oficial de las elecciones, quien inspecciona la superficie de marcado para asegurarse de que está intacta. Como la mitad izquierda se ha desechado, el oficial de las elecciones no puede decir por quién votó Alicia. Después, el oficial separa la superficie de marcado y la desecha a la vista de todos los observadores, incluyendo a Alicia. Alicia entonces pone en el escáner óptico lo que queda del voto –la marca y el código de barras–. Esto es su voto encriptado de forma efectiva. Alicia se lo lleva a casa con ella a modo de recibo.

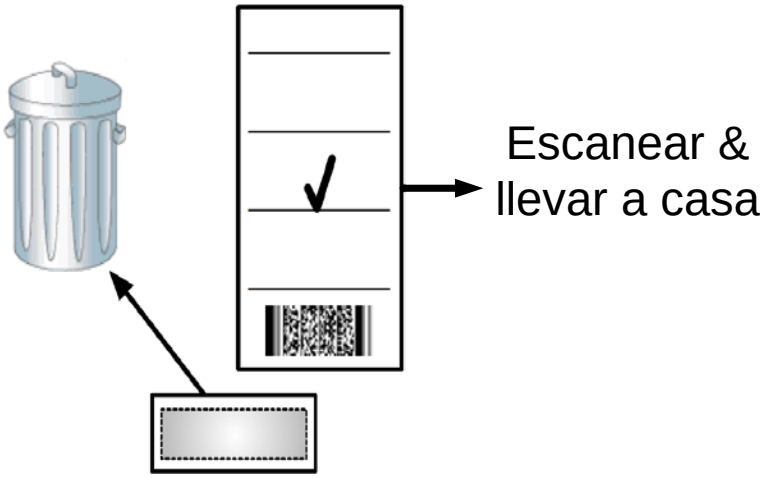

Figura 9. Emisión del Voto "Marca&Vota"

- **Verificación:** Alicia puede loguearse en el sitio web de las elecciones para verificar que su voto, incluyendo el código de barras y el marcado, ha sido correctamente cargado al tablón de anuncios. Si no ha sido así, ella se puede quejar con su recibo en mano. Alicia también puede verificar el proceso de escrutinio completo, incluyendo la suma de todos los votos en uno único encriptado, y la desencriptación verificable llevada a cabo por los oficiales de las elecciones.

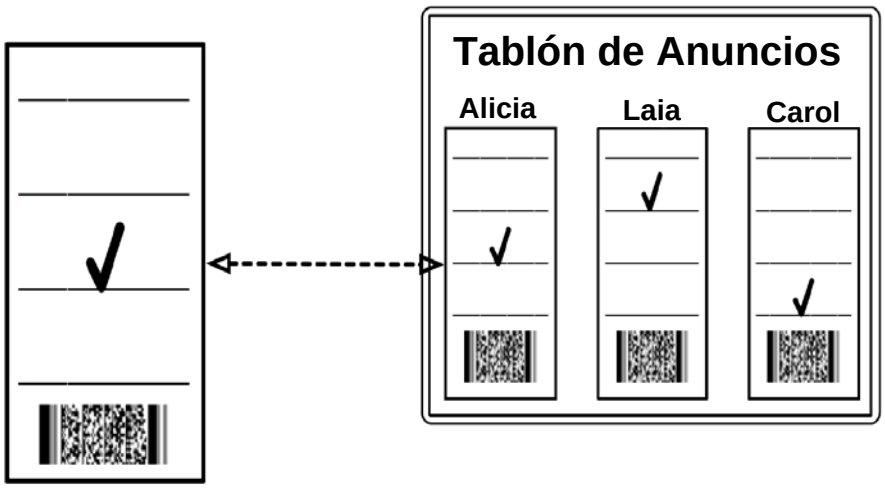

Figura 10. Verificación de que se ha contado el voto

3.1.2. Visión general de las ideas

Rivest combinó en este artículo varios ideas existentes sobre votación critográfica de una manera novedosa en su mecanismo “Marcar&Votar”.

Recuento homomórfico. Los votos criptográficos de papel no soportan por sí mismo que se pueda escribir en ellos. Generalmente, cuando Alicia quiere escribir un nombre, selecciona la opción predetermina de pseudo-candidato, y sigue un proceso separado para especificar su candidato. Así, la primera propuesta de los autores para M&V es usar la agregación homomórfica para simplificar el recuento de candidatos predeterminados [2][3]. Esta elección de diseño abre la puerta a simplificaciones futuras.

Código de barras 2-D y superficie de marcado. Con el recuento homomórfico y nombre de candidatos predeterminados, todos los textos cifrados requeridos pueden ser representados usando un único código de barras. Los valores aleatorios utilizados para generar estos textos cifrados se imprimen también en el voto, debajo de la superficie de marcado. Así, un voto está autocontenido de forma completa: mediante el marcado y la comprobación de la encriptación del orden del candidato con el texto cifrado en el código de barras, uno puede obtener inmediatamente si el voto es correcto o no. Esta revisión requiere de un simple escáner de códigos de barras, un simple ordenador, y los parámetros públicos de las elecciones.

Cortar-y-escoger. Una vez un voto es revisado, no puede usarse para votar: con los valores aleatorios revelados, el voto no protege más la privacidad. Así, la revisión se usa en un proceso de cortar-y-escoger: cada votante puede escoger dos votos, revisar uno y votar con el otro. La ventaja específica de M&V es que esta revisión no requiere de la intervención de un oficial de las elecciones: el voto y los parámetros de las elecciones públicos son suficientes. Así, revisar un voto M&V se puede realizar en vivo, en frente del votante, antes de que emita su voto. Además, los oficiales de las elecciones locales pueden revisar un número

de votos por su propia cuenta antes de que empiece la votación: una vez más, estos oficiales sólo necesitan los parámetros públicos de las elecciones para revisar con éxito.

Pruebas de corrección y lista de votos certificada. En un sistema de recuento homomórfico, los revisores quieres asegurarse de que los votos encriptados no contribuyen más que en un solo voto, de cualquier otra forma, un oficial y un votante maliciosos podrían inflar el recuento de un candidato. Con este propósito, los oficiales preparan pruebas de conocimiento cero de la corrección para cada voto oficial. Estas pruebas se publican en el tablón de anuncios para que todos las vean antes del día de las elecciones, y sólo los votos que verifican las pruebas se incluyen en el recuento.

Como resultado de esta condición de recuento, los votantes ahora necesitan la seguridad de que su voto no será descartado en el algún punto después de la emisión del voto. Desafortunadamente, el puro tamaño de la prueba descarta imprimirla en el voto junto con el texto cifrado.

Para encarar esta preocupación, los oficiales de las elecciones producen una lista de votos certificados que contienen los oficiales están preparados para garantizar como correctos. Esta lista certificado puede ser descargada fácilmente a cada recinto físico antes de que se abran las urnas. El votante puede entonces comprobar que su voto se encuentra en la lista certificada antes de empezar a votar. Además, esta certificación previene quejas falsas de votantes maliciosos que podrían inyectar votos fraudulentos en el sistema sólo con el propósito de quejarse y entorpecer el transcurso de las elecciones.

3.2. PRELIMINARES

Además de los conceptos explicados en el punto anterior, para entender el artículo de estudio [11] se proveen en esta sección una serie de conceptos criptográficos.

3.2.1. Contadores Homomórficos

La propuesta de votación homomórfica no es propia de Rivest sino que fue introducida por Baudron usando técnica introducidas por Benaloh.

Baudron describe un multicontador encriptado bajo un sistema aditivo criptográfico como el de Paillier. El espacio de bit del texto en claro se particiona en contadores separados, asegurando que se dedican suficientes bits a cada contado para que no ocurra *desbordamiento* de un contador a otro. Asumiendo un dominio del mensaje Z_n donde $\kappa = |n|$ es el número de bits de *n*, se codifica el valor t_j para el contador $j \in [1, z]$ como

$$t_j \cdot 2^{((j-1)M)}$$

y, por tanto, el conjunto de *z* contadores como:

$$\sum_{j=1}^{z} t_j \cdot 2^{((j-1)M)}$$

Así, cada contador sólo puede llegar a $2^M - 1$, y se debe asegurar que $\kappa > zM$. Para sumar 1 al contador *j* contenido dentro del multicontador *T*, debemos usar la propiedad de suma homomórfica:

$$T' = T \cdot \mathcal{E}_{pk}(2^{(j-1)M})$$

Cabe destacar que dada la seguridad semántica del criptosistema de Paillier, un observador no puede discernir, mirando a la operación homomórfica, que contador interno se incrementa. En otras palabras, dado un mensaje encriptado, la agregación homomórfica en un contador encriptado la puede hacer cualquiera, incluyendo oficiales de las elecciones y observadores que no tienen información privilegiada.

3.2.2. Pruebas de corrección

Si Alicia encripta un mensaje *m* en un texto cifrado *c* usando el criptosistema de Paillier, ella puede probar, mediante un verificador honesto de conocimiento cero, que *c* es de hecho la encriptación de *m*, usando un protocolo típico interactivo de tres rondas.

Usando las técnica de Cramer, este protocolo se puede extender para probar que el texto cifrado *c* encriptado *un* posible valor ($m_1, m_2, \ldots, m_z$), sin revelar cuál. Combinando esto con la prueba homomórfica de Jules y Jakobsson, uno puede probar, de una manera bastante eficiente y con conocimiento cero, que un conjunto de textos cifrados ($c_1, c_2, \ldots, c_z$) encriptan una permutación de $m_1, m_2, \ldots, m_z$, asumiendo que nunca dos subconjuntos de $\{m_i\}$ tienen la misma suma:

- Para cada c_i, se prueba que c_i encripta un $m_1, m_2, \ldots, m_z$.
- Se prueba que cada texto cifrado homomórfico sumado $\oplus_i c_i$ es la encriptación correcta de la suma de textos en claro $\sum_i m_i$.

Para más de un puñado de textos en claro, existen técnicas de prueba más eficientes [4].

3.2.3. Votos de papel

Actualmente existen métodos criptográficos basados en papel usando dos tipos de estructura: el voto para partir Prêt-a-Voter, y el voto a capas Punchscan. En estos preliminares revisamos estos dos tipos de estructuras necesarias para entender el artículo [11], ya que ambas estructuras se pueden adaptar para usar M&V.

Prêt-a-Voter. En este tipo de estructura, el voto es una única página de papel con una línea en la mitad vertical perforada. Los nombres de los candidatos aparecen en la izquierda en un orden aleatorio según el voto, con un especio en

la mitad derecha para que el votante pueda marcar su elección. Después de que el votante haya marcado su elección, se separan las dos mitades: la mitad izquierda (que es la que contiene los nombres de los candidatos) se desecha, y la mitad derecha es la que se emite como voto. La mitad derecha contiene información que permite a los administradores recrear la media parte izquierda del voto y determinar la elección del votante.

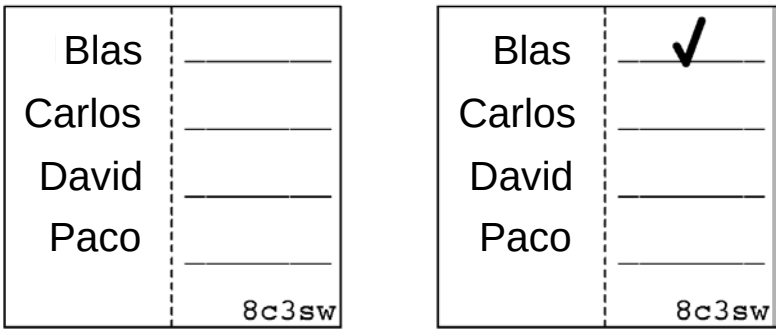

Figura 11. Voto Prêt-A-Voter

Punchscan. En esta estructura, el voto se compone de dos hojas sobrepuestas. La hoja superior contiene la pregunta, una asignación de los códigos de los candidatos (aleatorios en cada voto), y agujeros redondos físicos que revelan los códigos de la última página. Los códigos de la página inferior coinciden con los códigos de la página superior, aunque el orden en la página inferior es aleatorio. También pueden existir agujeros “tontos” y valores “tontos.

Alicia, la votante, selecciona un candidatos, determina qué código corresponde a ese candidato, y usa un “marcador de bingo” para marcar el código apropiado a través del agujero físico. El uso de este marcador grueso causa que ambar hojas se marquen. Después, Alicia separa las dos hojas, destruye una, y emite la otra. Individualmente, cada hoja muestra la elección del votante ya sea como código o como posición, pero la correspondencia de código a posición sólo es visible cuando ambas hojas están juntas. Una permutación de ambas hojas permite a los administrados reconstituir la hoja desechada y recuperar el voto.

Como es Alicia la que elige qué media parte destruye y qué media parte emite, puede eventualmente asegurarse que su voto estaba formado correctamente.

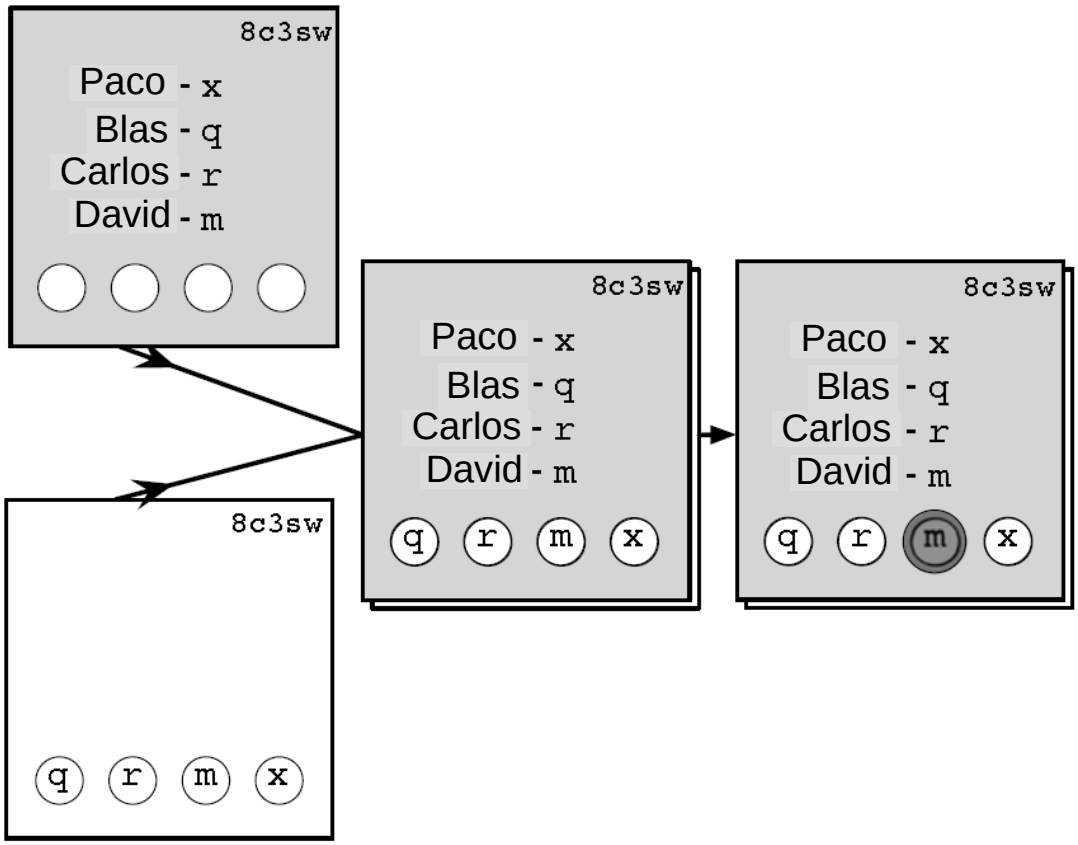

Figura 12. Voto Punchscan

Revisando Prêt-a-Voter y Punchscan. En ambas métodos, hay dos componentes a ser revisor: la correcta forma del voto, y el escrutinio correcto. En ambos esquemas, para la mitad seleccionada aleatoriamente de los votos, los oficiales de las elecciones revelan los valores aleatorios que se usan para crear los votos. Estos votos revisados se echan a perder, ya que dejan de proteger el secretismo del voto. Los votos restantes, que en este punto se ha probado de que estarán correctamente formados con alta probabilidad, se usan en las elecciones reales. Una vez se emiten los votos, estos se barajan, y la revisión post-elecciones consiste en una *Comprobación Parcial Aleatoria* sobre lo barajo y la permutación anterior. Punchscan añade una verificación adicional

del voto después de las elecciones, gracias a la decisión del votante de decidir qué parte guarda y qué parte desecha. Esto garantiza que cualquier voto de engaño que se hizo durante la revisión inicial será detectado con 50% de probabilidad.

Limitaciones. En el caso de Prêt-a-Voter, se exige un sincronismo considerable de los oficiales de las elecciones durante todas las revisiones. Esto es particularmente exigente para revelar los valores aleatorios de la mitad de los votos mientras se mantienen la otra mitad en secreto. Consecuentemente, la revisión la realizan los oficiales de antemano. Los votantes individuales deben confiar que esta revisión se realizó de forma correcta, y en particular que los oficiales de las elecciones no conspiraron para producir votos falsos. De hecho, esto es un punto débil de la propiedad de seguridad de emisión de votos deseable para sistemas de voto criptográficos. Idealmente, los votantes deberían obtener seguridad directa de que su voto fue registrado como se esperaba, sin tener que confiar en los oficiales de las elecciones.

En el caso de Punchscan, existe cierto grado de dependencia del sincronismo de los oficiales. Sin embargo, cabe destacar que la comprobación adicional que se realiza en los votos alivia esta situación: Alicia ahora obtiene seguridad directa de que su voto se formó correctamente. Desafortunadamente, esta seguridad se produce después de cerrar las elecciones. Esto puede reducir la confianza de Alicia en el sistema, ya que el protocolo de corrección de errores será probablemente oneroso.

3.3. EL MÉTODO MARCAR & VOTAR

Ahora presentamos los detalles del método Marcar & Votar que proponer Rivest en su artículo. Para hacer el proceso la más concreto posible, damos un ejemplo práctico de unas elecciones con 4 candidatos.

3.3.1. Preparación de las elecciones

En algún momento previo al día de las elecciones, la lista de los candidatos se certifica por los oficiales y se publica para que todo el mundo la conozca. Los *z* candidatos se ordena de alguna manera con el propósito de asignarles índices numéricos (el orden alfabético ya sirve). Este orden tiene que conocerlo los votantes.

Después, los oficiales de las elecciones generan de forma conjunta una paraje de claves para las elecciones, donde el oficial O_i tiene la parte $sk^{(i)}$ de la clave de desencriptación *sk,* y la clave pública combina se denota por *pk*. Un parámetro *M* se escoge, tal que 2^M es mayor que el número total de votantes con derecho a elegir. Los oficiales de las elecciones aseguran que $\kappa = |n|$ es suficiente grande para codificar un multicontador de *z* candidatos, cada uno con *M* bits. Así, un voto para el candidato *j* se codifica como $\mathcal{E}_{pk}(2^{(j-1)M})$. Cuando todo está establecido, los parámetros de las elecciones se publican:

$$params = (pk, M, \{cand_1, cand_2, \dots, cand_z\})$$

Ejemplo. Con *z* = 4 candidatos, se asigna un orden de candidatos: Paco i #1, Blas #2, Carlos #3, y David #4. Con 2×10^8 votantes (suficientes bits para toda España), escogemos *M* = 8 $> log_2(2 \times 10^8)$. Una clave pública Paillier con $\kappa = |n| = 1024$ bits es suficientemente grande para estas elecciones. De hecho, podríamos tener hasta 35 candidatos, sin tener que alterar los parámetros criptográficos.

3.3.2. Preparación de los votos

El voto. El primer voto M&V se basa en el esquema Prêt-a-Voter. Cada voto se perforada a lo largo de una línea vertical en el medio del voto. En la mitad izquierda del voto se pone la lista de los candidatos, en un orden aleatorio. En la mitad derecha del voto se ponen casillas de marcación escaneables, con un

candidato a la izquierda de cada casilla. El votante marcará una de estas casillas.

Además en la parte derecha, los votos M&V contienen una representación de los textos cifrados que codifican los votos correspondiente a cada opción ordenada. La representación de estos textos cifrados puede ser legible mediante máquina y debería ser ópticamente escaneable, por ejemplo un código de barras. Justo debajo de este código de barras, el voto incluye una superficie de rascado, debajo de la cual se encuentran los valores aleatorios usados para producir los textos cifrados en el código de barras.

Las pruebas. Los oficiales de las elecciones deben también generar pruebas NIZK para la corrección del voto. Estas pruebas no serán imprimidas en el voto, ya que son demasiado largas. A cambio, se guardan en el tablón de anuncios públicos, indexados por la secuencia de textos cifrados en el voto correspondiente, y usado en el momento del recuento para asegurar que los votos contribuyen como mucho en una unidad.

Además de estas pruebas, los oficiales de las elecciones compilan una lista de votos oficiales, la cual incluye todos los votos creados apropiadamente designados otra vez por la secuencia de textos cifrados en cada voto. Los oficiales firman digitalmente esta lista de votos, posteando la lista y la firma en el tablón de anuncios.

Esta lista de votos de oficiales es particularmente útil para ayudar a prevenir varios ataques de denegación de servicio.

Ejemplo. Asumir que el orden de candidatos de un voto dado es “Blas, Carlos, David, Alicia”, o, por índice numérico, “2, 3, 4, 1”. Del ejemplo anterior M = 28. La codificación en la mitad derecha del voto debería ser: $c_1 = \mathcal{E}_{pk}(2^{28}; r_1)$, $c_2 = \mathcal{E}_{pk}(2^{56}; r_2)$, $c_3 = \mathcal{E}_{pk}(2^{84}; r_3)$, $c_1 = \mathcal{E}_{pk}(2^{0}; r_4)$.

Esta codificación requiere 4 textos cifrados, o 8192 bits. Debajo de la superficie de rascado, en texto en claro, pondrá r_1, r_2, r_3, r_4. Los oficiales también generan $\prod_{H(c1,c2,c3,c4)}$, una NIZK de la forma correcta del voto indexada por los textos cifrados, y después postearlo en el tablón de anuncios. El mismo hash $H(c_1, c_2, c_3, c_4)$ también se incluye en la lista compilada de votos oficiales, la cual los oficiales firmal eventualmente.

3.3.3. Revisión de votos

La revisión de votos en M&V usa la propuesta cortar-y-escoger para verificar el orden de candidatos. Se revisa una mitad aleatoria de los votos, y entonces se garantiza que casi todos los votos restantes están bien construidos: la probabilidad de que más de *x* votos malos no se detecte es 2^{-x}. Una vez revisado, un voto se echa a perder y no puede usarse para emitir un voto.

Revisando un único voto. Este proceso de revisión es similar al de Prêt-a-Voter y Punchscan, con una gran diferencia: la revisión puede realizar usando sólo parámetros *públicos* de las elecciones, sin la interacción de los oficiales:

1. **Rascado:** la superficie de rascado se quita para revelar los valores aleatorios.

2. **Encriptación:** el orden de los candidatos se encripta con los valores aleatorio revelados.

3. **Comparación:** el texto cifrado resultante se compara con el texto cifrado en el código de barras 2-D.

Cabe destacar, que para automatizar este proceso in tener que escanear el orden de candidatos, uno podría realizar la comprobación en el otro sentido: leer el texto cifrado, intentar todos los candidatos en texto en claro posibles con cada valor aleatorio, y comprobar el orden de candidatos resultante con el orden de

candidatos en el voto. Comprobar el orden puede realizarlo el votante por sí mismo.

Echando a perder un voto. Una vez que los valores aleatorios se revelan, un voto no protege más la privacidad de éste. Por lo tanto, si la superficie de rascado se quita, un voto debería desecharse, y no puede emitirse. Este proceder es consistente con el uso de superficies de rascado existentes como por ejemplo las usadas en loterías.

¿Quién realiza la revisión? Aunque se suele considerar que preferible que los votantes revisen individualmente votos, algunos prefieren revisar los de manera centralizada. Marcar&Votar suporta tal método de revisión. También se puede pensar que los oficiales revisen unos pocos votos por su propia cuenta el día antes de las elecciones, además de la revisión de cada votante. M&V permite este tipo de revisiones combinadas.

Comprobar la variabilidad del orden. Oficiales de elecciones maliciosos podrían intentar romper la privacidad de Alicia presentando a todos los votantes sólo votos con el mismo orden de candidatos. Para proteger contra este ataque desaleatorizador, Alicia debería seleccionar dos votos por sí misma, asegurando que hay suficiente variabilidad entre los votes que se le ofrecen.

Alicia también debe comprobar el voto que ella realmente usa: necesita asegurarse que su voto contará, específicamente debe asegurarse que no será descalificado por alguna razón imprevista, por ejemplo una prueba NZIK inválida durante el momento del recuento. Con este propósito, Alicia comprueba la presencia de su voto en la lista oficial de votos, la cual puede obtener del tablón de anuncios. Si, en cualquier otro momento posterior, el voto de Alicia es descalificado por cualquier razón, ella puede presentar la lista oficial de votos como queja.

3.3.4. Emitiendo el voto

Un día después de las elecciones, después de haber revisado y desechado el primer voto, Alicia entra en la cabina de aislamiento con un segundo voto. Rellena éste marcando la opción que desee. Seguidamente, separa la parte izquierda del voto y la tira en el receptáculo apropiado (dentro de la cabina). Después, abandona la cabina y emite su voto de la forma siguiente:

1. **Confirmación:** un oficial de las elecciones verifica que la superficie de rascado del voto de Alicia permanece intacta. Esto es crucial para asegurar que el voto es secreto: si un votante ve los valores aleatorios del voto que realmente emite, entonces puede probar que voto a un potencial coaccionador. Entonces, el oficial separa y tira la superficie de rascado a la vista de todo el mundo.

2. **Escaneo:** Alicia introduce lo que queda de su voto en una máquina de escaneo óptico, la cual grava el código de barras y la posición marcada y los postea en el tablón de anuncios junto con el nombre de Alicia o el número de identificación del votante.

3.3.5. Recuento

Para cada voto en el tablón de anuncios, los oficiales de las elecciones y los observados comprueban su NIZK. Si está se verifica, el texto cifrado correspondiente a la posición marcada se extrae del código de barra y se suma al contador homomórfico, tal y como se hace en cualquier sistema de votación homomórfico. Cualquiera puede verificar que sólo los votos válidos se suman, ya que cualquiera puede verificar la prueba NIZK y re-realizar la suma homomórfica apropiada.

De forma similar, todos los fiduciarios de las elecciones pueden de manera independiente verificar que la suma homomórfica se ha realizado de forma de correcta. Después, el contador en texto cifrado resultante se desencripta

mediante un quórum de estos fiduciarios, a la vez que se realizan pruebas del desencriptado correcto. El texto en claro resultante revela la cuenta de votos para cada candidato. El recuento y las pruebas de los fiduciarios se postean en el tablón de anuncios para que las vea todo el mundo.

3.3.6. Estimaciones del funcionamiento

Se consideran los requisitos de cálculo y espacio, específicamente de:

- generar las pruebas NIZK para cada voto, dado que las pruebas de conocimiento cero son normalmente costosas,
- revisar un voto durante las votaciones, dados el tiempo limitado de los votantes y su paciencia, y
- el tamaño físico de los códigos de barras.

Considerando una elección con 5 vueltas, cada una con 5 candidatos.

Pruebas. Cada vuelta contiene 5 textos cifrados, uno por candidatos. Usando la técnica de prueba de conocimiento parcial CDS, se debe probar que cada texto cifrado encripta uno de los 5 candidatos. La técnica CDS simula 4 de esas pruebas y computa una quinta. Esto requiere el trabajo equivalente de 5 pruebas, tanto en términos de computación como número de bits necesarios para representar la prueba. Además, la suma homomórfica de los textos cifrados se debe probar que encripta la suma de las representaciones de los candidatos, que supone una prueba más. Así, cada vuelta requiere 26 pruebas, y por lo tanto 5 vueltas requieren 130 pruebas.

Cada una de estas pruebas, ya sea real o simulada, requiere dos exponenciaciones modulares. Así, la prueba entera requiere un total de 260 exponenciaciones modulares. De manera conservativa, un procesador moderno puede realizar una exponenciación modular de 1024 bits en aproximadamente

12ms en un máquina a 2.8Ghz. Por lo tanto, una única prueba de un voto se puede realizar en unos 3 segundos en una única CPU.

Cada una de estas pruebas está compuesta por dos elementos en el espacio de textos cifrados de Paillier, y uno en el espacio de textos en claro de Paillier. Asumiendo un módulo de 1024 bits, los elementos del texto cifrado son de 2048 bits y el texto en claro es de 1024 bits [1]. Así, cada prueba requiere 5120 bits, y, por lo tanto, el voto completo requiere 83 kilobytes de datos de prueba en el sitio web.

Verificación de votos. En las urnas, la única verificación que se necesita es la de la correcta encriptación del voto revisado. Dado 5 valores aleatorios, todos los 5 valores de r^n se puede calcular a través de la exponenciación modular, después de la cual sólo se necesitan multiplicaciones modulares, que son negligibles por comparación. Así, la verificación de los votos se puede realizar en 60ms por vuelta, o 300ms para nuestro voto considerado. El tiempo de voto y de rascado de cada votante (1-2 minutos) hace casi negligible el coste criptográfico.

Tamaño del código de barras. El código de barras 2D PDF417 puede guardas 686 bytes de datos binarios por pulgada cuadrada, usando una densidad de símbolo que es fácilmente imprimible y escaneable. En nuestro ejemplo con 25 candidatos, se requieren 25 textos cifrados de Paillier, lo que significa 6400 bytes, asumiendo un módulo de 1024 bits para Paillier. Así, 10 pulgadas cuadras son suficiente para representar todos los texto cifrados necesarios para este ejemplo de elecciones.

3.4. EXTENSIONES

Durante el artículo [11] se proponen extensiones para hacer el método Marcar & Votar incluso más práctico.

3.4.1. Organizaciones ayudantes

No se espera que los votantes se personen en las urnas sin el equipamiento requerido para la revisión de los votos y comprobar la lista oficial de votos. A cambio, las organizaciones ayudantes, por ejemplo los partidos políticos y las organizaciones activistas, pueden proporcionar este servicio a las urnas. Los votantes pueden consultar uno a más de estas organizaciones según su consideración. Además, a ninguna de estas organizaciones se les confía información privada acerca de las elecciones.

3.4.2. Múltiples vueltas & Múltiples Candidatos

Cuando las elecciones contienen más de una vuelta, puede causa que el multicontador se quede pequeño, ya que sólo puede contener $|n|/z$ contadores. Una solución es usar un grado de encriptación más elevado usando la generalización de Damgård-Junik, para que el espacio de cada contador sea de $s|n|$ en vez de $|n|$, con una longitud del texto cifrado correspondiente $(s+1)|n|$. Desafortunadamente, el tamaño de este texto cifrado puede sobrepasar la codificación del código de barras, que se espera que no sea de más de uno pocos kilobytes.

Otra opción es designar, en los parámetros públicos de la elección, multicontadores separados, uno para cada vuelta. En este caso, los parámetros deben también indicar las asignaciones de vuelta/multicontador. Con un código de barras separado para cada vuelta, se pueden alcanzar casos más prácticos.

En casos extremos, no hay más remedio que usar la generalización de Damgård-Junik, ya que los textos cifrados para una vuelta dada podrían ser no distinguibles y por lo tanto no se les podría asignar diferentes multicontadores. Si un único código de barras no puede aguantar todos los textos cifrados requeridos para una vuelta, se puede designar como última opción un código de

barras diferente para cada candidato. Bajo estas condiciones extremas, es inevitable que la complejidad del proceso de revisión aumente.

3.4.3. Reduciendo la superficie de rascado

Dado que el material impreso bajo la superficie de rascado puede dañarse durante el rascado, y no se puede esperar que almacene de forma fiable cantidades de datos considerables bajo esta superficie de rascado. De hecho, es fácil reducir estos datos mediante la designación de una función pseudo-aleatorio como parámetros de la lección. Esta función generadora (PRF) genera todos los valores aleatorios a partir de una semilla pequeña, que necesita sólo 128 bits. Tal longitud de datos puede ser fácilmente codificada como caracteres alfanuméricos o como un código de barra de una sólo dimensión, ambos ofreciendo suficiente redundancia para resistir a unos pocos rascados.

3.4.4. Cadena de votación y superficies de rascado

Se sabe desde hace mucho tiempo que todos los sistemas de votación basados en papel son susceptibles a ataques de cadena de votación. En estos ataques, un coaccionador da a Alicia un voto premarcado antes de que entre en la zona de votación, esperando que Alicia emita este voto premarcado y le devuelva un voto en blanco cuando salga.

La contramedida para estos ataques de cadena en votaciones sugiere tener unos identificadores únicos de votos en un papelito arrancable pegado al voto. Un oficial escribe el identificador de voto para Alicia antes de que entre en la cabina de aislamiento. En el momento de emitir el voto, el oficial comprueba que el identificador de voto todavía está presente y comprueba el valor. Después, para conservar el anonimato, el identificador es arrancado y desechado.

Este proceso es, de hecho, casi idéntico a la superficie de rascado que se sugiere en el método M&V. Además de comprobar el identificador, en el método M&V, el oficial de las elecciones debe simplificar comprobar la integridad de la

superficie de rascado. Así, la sobrecarga de este método en el momento de emitir el voto es mínima.

3.5. Adaptando Punchscan

El método de Punchan permite más vuelta por voto que el método Prêt-a-Voter, porque el voto entero puede usarse sin una línea de separación. Sin embargo, Punchscan es también más complicado, porque el votante puede emitir cualquier hoja. Esto obliga la destrucción de la otra mitad más delicada, ya que Alicia podría fácilmente vender su voto si ella presenta la segunda mitad.

Por lo tanto, el artículo [11] propone un nuevo voto que combina las propiedades de Punchscan y Prêt-a-Voter, y añade el método M&V. Como este voto se origina a partir de Punchscan, se le denomina voto "Punchscan Marca&Vota". Sin embargo, cabe destacar que heredad algunas propiedades del Prêt-a-Voter.

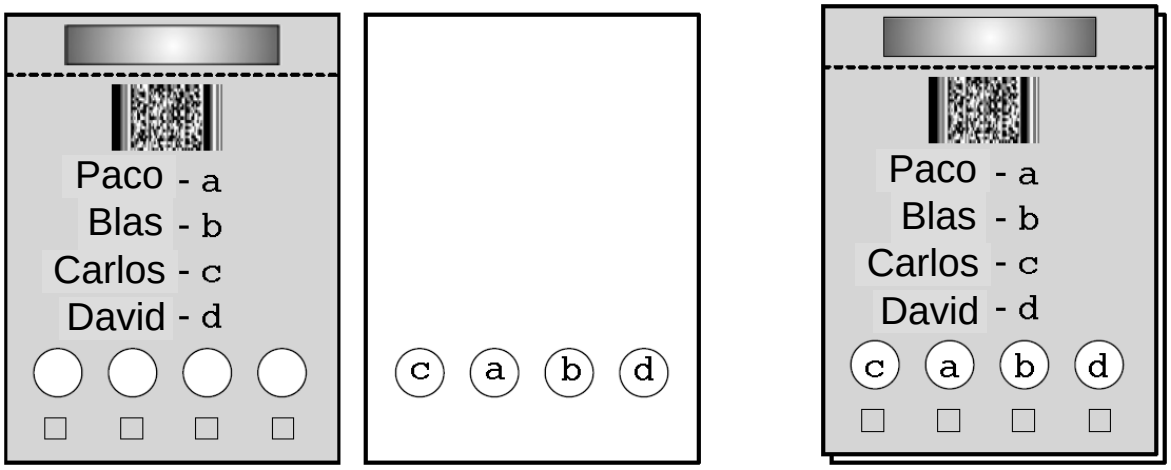

Figura 13. Variante de voto Punchscan Marcar&Votar

Esta variante de voto, está compuesta por dos hojas superpuestas. A diferencia del Punchscan original, las dos hojas sirven para diferentes propósitos. Alicia, la votante, separará ambas mitades y emitirá la hoja superior en todos los caso. La mitad inferior es genérica y su destrucción no necesita ser verificada. Este cambio es seguro porque el proceso cortar-y-escoger se realiza ahora por dos

votos verificadores, en vez de partir uno. Además, mientras que el método cortar-y-escoger Punchan original sólo puede verificarse después de haber emitido el voto, esta variante permite al votante verificar antes de emitir el voto.

La hoja superior contiene las vueltas y los candidatos en orden estándar, con un código estándar asignado a cada candidato (por ejemplo, “S” para los socialistas y “P” para los populares). Otra vez, esto difiere del voto Punchscan, donde códigos aleatorios se asignan a los candidatos. Esta hoja superior ofrece casillas de marcado para cada vuelta, y un agujero encima de cada casilla que revela el código mostrado en la mitad inferior en esa posición. También incluida en la hoja superior están los componentes M/V: el texto cifrado en un código de barras 2D, y los valores aleatorios escondidos bajo una superficie de rascado.

La hoja inferior contiene sólo los códigos de los candidatos estándar en un orden aleatorio. Los textos cifrados en la primera hoja deberían coincidir con este orden aleatorio. Destacar, otra vez, que la hoja inferior es completamente genérica: no contiene ningún identificador de ningún tipo, ningún texto cifrado, y ningún valor aleatorio.

Verificación prevotación. Igual que en el método original, Alicia escoge dos votos. Comprueba uno mediante el rascado y ayudad de una organización para verificar que los valores aleatorios coinciden con el orden de candidatos del voto. Alicia entonces vota con el segundo voto, ya que el ahora el voto rascado es nulo.

Emitiendo el voto. Alicia marca su selección en aislamiento. Dado que, a diferencia del método Punchscan original, las marcas en la hoja superior no calcan a la hoja inferior. Cuando está lista, Alicia separa la hoja inferior del voto y la tira en un receptáculo adecuado.

Después, Alicia presenta la hoja superior de su voto al oficial de las elecciones, quien la trata exactamente como si fuera un voto del esquema Prêt-a-Voter

Marcar&Votar: verifica la superficie de rascado, la separa y la tira, y escanea el resto. Como anteriormente, esta mienta restante no revela cómo Alicia votó, lo que significa que el oficial puede examinar con cuidado el voto sin poner en riesgo la privacidad de voto. Como siempre, lo que queda del voto le sirve a Alicia como recibo.

3.6. MODELADO DE LA AMENAZA

En el artículo [11] se consideran varias amenazas y cómo el método propuesto las tiene en cuenta. En este capítulo se describen estas amenazas según la temática de éstas en vez de tratarlas en un orden cronológico como sucederían durante un proceso electoral, ya que existen amenazas que atañen a diferentes partes de este proceso.

3.6.1. Atacando a los votos

Como resulta obvio, el primer objetivo de ataque en cualquier método de votación es la creación de votos maliciosos. Se consideran varias entidades que podrían verse envueltas en este ataque.

Oficiales de las elecciones: Un oficial de unas elecciones podría crear votos malos de dos maneras: un voto completamente inválido, o todavía más perjudicial, un voto válido que no coincide con un orden de candidatos legible por el hombre. En cualquier caso, la primera manera de defenderse es por medio del esquema de cortar-y-escoger: sólo un pequeño número de tales votos pueden pasar indetectables, ya que la mitad de ellos serán revisado aleatoriamente. En el caso de votos completamente inválidos, la verificación es mucho más rigurosa: los oficiales serían responsable de votos certificados que no tuvieran una prueba NIZK, y sólo votos válidos podrían tener una prueba NIZK.

Entidades externas. Entidades externas podrían querer introducir votos fraudulentos, presumiblemente como ataque de denegación de servicio contra

votantes, o, simplemente incrementando el número de quejas, como ataque de denegación de servicio contra oficiales de las elecciones. Estos problemas se frustran mediante una lista de votos certificados. En el momento en que un oficial de unas elecciones descubre un voto nocertificado, comienza una investigación. Si los oficiales no captaran el problema, las organizaciones que ayudan a los votantes lo harían. Consecuentemente, votos fraudulentos deberían ser captados antes de que los votantes entrasen en la cámara de aislamiento.

Conspiración entre oficiales y votantes. Un único voto maliciosamente creado podría alterar el recuento de la agregación homomórfica de manera significativa. Estos es particularmente problemático si los oficiales colaboran de manera secreta con un votante quien no ejecutaría correctamente la revisión del esquema cortar-y-escoger porque quiere usar un voto fraudulento. Una vez más, la prueba NIZK en el tablón de anuncios hace que esto nunca pase, proporcionando una garantía universal verificable que cada voto emitido sólo contribuye en una unidad a un único candidato en cada vuelta.

3.6.2. Atacando la condición de voto secreto

Otro ataques obvio en una elecciones es atacar el secretismo garantizado por los votos, especialmente ya que toda la protección de un voto está contenido en el voto en sí mismo.

Votos agujereados. Los oficiales de las elecciones podrían filtrar información acerca del orden de los candidatos en un voto usando la aleatoriedad del texto cifrado. Esta amenaza se alivia en cierta manera mediante el uso de aleatoriedad basada en semillas, una que en al medida en que una porción de la semilla es pública y seleccionar después de que las semillas de los votos individuales son escogidas.

Alterando la superficie de rascado. Eva, una usuaria maliciosa, podría intentar quitar la superficie de rascado de su voto, leer los valores aleatorios, y reemplazar la superficie de rascado sin que se notase. Eso permitiría a Eva vender su voto, dado que su voto encriptado será posteado en el tablón de anuncios, junto con el nombre entero de Eva, para que todos lo vean y lo revisen. Se debe asumir, como defensa contra esta amenaza, que la superficie de rascado está a prueba de falsificaciones para prevenir un reemplazo tan fácil que engañaría a un administrado de las elecciones. En el mundo real harían falta experimentos para determinar qué nivel de superficie no falsificable es factible.

Grabación de la apariencia de los votos. Una debilidad considerable de todo criptosistema de votación pre-impreso basado en papel, como es el propuesto en el artículo [11], es que los oficiales de las elecciones que hacen los votos podrían grabar los textos cifrados de los votos y los órdenes de los candidatos, violando de esta forma el carácter de voto secreto.

Incluso en Prêt-a-Voter y Punchscan, que usa múltiples autoridades en un entorno de *mixnet*, el último servidor mezclador sabe el orden final de candidatos. Podría valer la pena explorar modos de distribución del mecanismo generador de votos. La mejor solución podría estar basada en procesos, donde las máquinas producen los votos e inmediatamente olvidan el mecanismo aleatorio usado.

Por otra parte, el riesgo de grabación no sólo existe por parte de los oficiales de las elecciones: aquéllos que transportan los votos podrían tener una oportunidad de grabar la correspondencia entre el orden de candidatos y el código de barras. Cabe destacar que tanto Prêt-a-Voter como Punchscan sufren de este mismo problema. Una defensa prometedora en todos los casos sería simplemente esconder cierta parte del voto de tal manera que no pueda ser identificado de manera única durante el transporte. Por ejemplo, los votos podrían ser sellados

individualmente en envoltorios opacos. O de forma alternativa, los códigos de barras podrían ocultarse debajo de un plástico opaco que puede ser extraído antes de la votación. Si el código de barra fuera resistente, se podría incluso ocultar bajo otra superficie de rascado.

Emisión. En el momento de la emisión del voto, los administrados de las elecciones deben asegurarse de que el voto emitido no tiene los valores aleatorios en claro. Por supuesto, esta verificación debe realizarse sin violar el carácter de secreto del voto en el proceso. La propuesta de los autores trata esta amenaza: un oficial sólo ve la mitad encriptada del voto. Así, el oficial puede hacer lo que sea necesario para verificar que la superficie de rascado está intacta sin hacer que el voto deje de ser secreto.

Sin embargo, queda una amenaza: la conspiración entre oficial y votante. Si un oficial y un votante conspiran en secreto para preservar, en vez de descartar, la superficie de rascado, el votante podría revelar su selección al oficial. Este problema debería ser tratado mediante la suficiente observación de los interventores durante el proceso electoral. M&V mitiga el riesgo al crear votos donde la mitad que se desecha es genérica.

Aleatoriedad coaccionada. El artículo [11] también trata una nueva amenaza de los sistemas basados en papel: la aleatoriedad coaccionada [5]. En este ataque, un coaccionador quiere reducir el voto de Alicia a aleatorio. Considerar, por ejemplo, la situación donde Alicia vota en un distrito donde históricamente tiene más peso un partido político por un amplio margen. Tales distritos son muy comunes en muchos países. Un coacciones puede pedir a Alicia que vote por el primer candidato de la lista, no importa quien sea. El coaccionar puede comprobar esto mirando en el tablón de anuncios bajo el nombre de Alicia o el número identificador. Por supuesto, el identificador no sabrá seguro por quién votó Alicia pero él habrá reducido el voto de Alicia a un voto aleatorio. Con

suficientes votantes, el coacciona podría estadísticamente reducir el número de votos para el partido favorecido normalmente en ese distrito.

Un modo de defenderse contra este ataque, es ofrecerle al votante suficientes ordenaciones diferentes de modo que pueda escoger un voto donde el comportamiento que le ha indicado el coaccionador coincida con su elección personal. Desafortunadamente, el ataque puede hacerse mucho más complejo: por ejemplo, en vez de que el coaccionador sugestionara la votación al primer candidato de la lista podría seleccionar el candidatos dependiendo del identificador de voto.

3.6.3. Atacando el tablón de anuncios y el escrutinio

Gran parte de la seguridad del escrutinio depende del tablón de anuncios. Un atacante podría querer insertar datos fraudulentos, por ejemplo cambiar los parámetro de las elecciones o reemplazar el voto emitido por un ciudadano honesto. Mediante firmas digitales en todos los datos publicados se pueden prevenir estos ataques, asumiendo que una mínima PKI se despliega para certificar las claves públicas de los oficiales. Los observadores del contenido del tablón de anuncios, incluyendo las organizaciones que ayudan, pueden entonces detectar datos erróneos.

Alternativamente, un atacante podría querer suprimir información del tablón de anuncios. En particular, el servidor del tablón de anuncios podría suprimir información por sí mismo. Para protegerse contra este ataque, el tablón de anuncios debe desplegarse en múltiples servidores. Estos servidor pueden correr un protocolo de acuerdo Bizantino distribuido para asegurar la consistencia del contenido, o los observadores podrían simplemente depender de las firmas digitales del contenido y de la redundancia de los servidores para captar si algún servidor eliminar contenido.

Dado un tablón de anuncios implementado con las medidas de seguridad comentadas, los ataques en el proceso de recuento se pueden prevenir, porque cada paso es comprobado mediante pruebas. Los votos se prueban que están bien formados mediante la prueba NIZK en el tablón de nuncios, y cualquier observador puede rerealizar la suma homomórfica, y la última desencriptación del recuento también se prueba que es correcta.

4. Conclusiones

La utilización de tecnología en la captación de los votos abre un nuevo abanico de posibilidades que va mucho más allá del incremento de velocidad en el recuento.

Así mismo, la naturaleza digital de los votos, conjuntamente con la existencia de actores privilegiados en el nuevo escenario electrónico, plantean una serie de cuestiones esenciales de seguridad a resolver. Básicamente debemos considerar las siguientes:

1. La disponibilidad de los sistemas de captación de votos: protección ante ataques de denegación de servicio.

2. La garantía de la privacidad de los votantes, aun cuando éstos deben ser identificados adecuadamente.

3. La protección de los votos digitales en cuanto a su manipulación, eliminación o la adición de votos falsos.

4. La verificabilidad respecto del tratamiento de los votos, una vez los resultados son publicados.

Este último punto es de vital importancia puesto que la emisión de votos por vía electrónica supone una falta de transparencia para el votante que debe ser compensada adecuadamente. Es esencial asimismo que la verificabilidad no abra las puertas a la coacción o a la venta de votos.

La criptografía es el núcleo central de seguridad sobre el que deben pivotar todas las medidas de seguridad de índole tecnológica en un sistema de voto electrónico. Esta juega un papel central en la protección de los sistemas de voto

electrónico. La primera propuesta de sistema criptográfico para la protección de la privacidad de los votantes llegó en 1981 de parte del gran criptógrafo David Chaum. Siguieron multitud de propuestas, básicamente en el ámbito académico.

En este marco, los autores del artículo bajo estudio [11] han propuesto el método Marca & Votar, un sistema simple de votación mediante el uso de técnicas criptográficas que puede ser implementado con los medios tecnológicos que existen en la actualidad a un bajo coste y con una complejidad mínima.

Comparado con los sistemas de votación modernos mediante pantallas táctiles, puede parecer que este método es poco complejo. Pero de acuerdo con los autores, el sistema de Marcar & Votar es el mejor modo de permitir a los votantes por sí mismo comprobar que sus votos de papel se han contado como es debido.

Actualmente, incluso en los sistemas más modernos es necesario de un máquina que imprima recibos a los votantes conforme ejercieron su derecho a voto. Este método mejora esta parte del proceso de votación, aunque los votantes aún dependen de más gentes y de más procedimientos. Este método permite asegurar al votante que su voto es contado y de la forma en que el votante escogió.

Con los sistemas de votación basados en criptografía, la verificabilidad extremo a extremo es posible, porque cualquier votante debería ser capaz de revisar el proceso de votación por completo. Al mismo tiempo, tales procesos de revisión deberían ser balanceados contra la necesidad de anonimato.

Como los sistemas tradicionales basados en papel no proporcionar suficiente anonimato ya que el número único que aparece impreso en el voto para asegurar que es legítimo puede ser rastreable hasta el nombre del votante. En este ámbito, los autores usan técnicas criptográficas para mantener la identidad

del votante en secreto mientras aseguran que todos los votos emitidos son legítimos.

La propuesta de los autores se basa en esta idea y puede ser usada conjuntamente con otros esquemas de votación ya existentes. Un esquema recientemente propuesto, llamado Prêt-à-Voter, implica que en una mitad del voto aparezcan los nombres de los votantes en un orden aleatorio, y en la otra mitad las casillas para que el votante elija el candidato que quiera. Después de que el voto sea emitido, un votante parte el voto por la mitad separando los nombres de los candidatos y las casillas. Este sistema depende de un código criptográfico en la parte de las casillas para codificar el nombre de los candidatos en el orden que aparecen en el voto original.

La preocupación de este sistema es cómo asegurar que la información encriptada coincide con el orden de los nombres de los candidatos. Esto se resuelve en el artículo de estudio [11] a cada votante dos votos de papel. Los votantes escogen qué voto se revisa y cuál usan para emitir su elección. El proceso de revisión en sí mismo no les dice nada acerca del la validez del voto original, pero les proporciona un 50% de probabilidad de descubrir un voto de papel amañado. Y dada una probabilidad tan alta, los votos de papel ilegítimo serán detectados rápidamente en el proceso.

El método Marcar & Votar hace este proceso de revisión seguro porque permite que un voto de papel sea revisado sin tener que involucrar a un oficial (que podría ser corrupto). Este método introduce la novedad al sistema de Prêt-à-Voter de una superficie de rascado en la parte donde aparecen el nombre de los candidatos, mientras que el orden de estos nombres se codifica criptográficamente debajo de las casillas.

Para comprobar que un voto no está trucado, el votante simplemente tiene que rascar la superficie del voto correspondiente para ver un número que guarda una correspondencia con el orden de los nombres de los candidatos. En teoría, los

autores dicen que los votantes podrían usar software criptográfico en la sede electoral correspondiente para verificar este orden, pero explican que es más práctico que terceras entidades de confianza proporcionen los medios necesarios para comprobar los votos. Si el código coincide, entonces el voto revisar es legítimo, de lo que se induce que el otro voto es válido para votar.

Lo que es más importante y novedoso de esta propuesta, es que los votos con autocontenidos. El hecho de que sean autocontenido quiere decir que cualquier organización colaboradores de las elecciones, o el votante por sí mismo, puede revisar el voto antes de emitirlo y sin tener la necesidad de interactuar con los oficiales de las elecciones. Dado la superficie de rascado, que los autores la introducen como intuitiva, este nuevo esquema de votación puede ser especialmente útil para dar una explicación plausible del poder de la verificación criptográfica en los sistemas de votación.

El éxito de este sistema dependerá no tan sólo de sus características de seguridad. El sistema propuesto es considerablemente complicado, aunque toda esta complejidad no es gratuita, sino necesario para asegurar que el voto es secreto.

El principal determinante para la adopción de este esquema de votación, y en general de cualquier otro que use técnicas criptográficas, es que los oficiales de las elecciones lo entiendan y lo acepten, y que por otra parte los votantes se den cuenta de sus beneficios. Aunque los legisladores ya están poniendo su fe en software que no entienden todavía son reticentes. Existe la ironía de que para hacer unas elecciones más transparentes sea necesario el uso de criptografía haciendo que los procesos subyacentes sean más complejos.

BIBLIOGRAFÍA Y REFERENCIAS

[1] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, y Guillaume Poupard. *Practical multi-candidate election system*. En PODC, páginas 274–283. ACM, 2001

[2] C. Andrew Neff. *Coerced Randomization*, abril 2006. Manuscript no publicado de Andy Neff.

[3] David Chaum. *Punchscan*. http://punchscan.org visto el 13 de agosto, 2006.

[4] David Chaum. *Secret-Ballot Receipts: True Voter-Verifiable Elections*. IEEE Security and Privacy, 02(1):38–47, 2004.

[5] Ronald Cramer, Ivan Damgard, y Berry Schoenmakers*. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols*. En Yvo Desmedt, editor, CRYPTO, volume 839 del Lecture Notes in Computer Science, páginas 174–187. Springer, 1994.

[6] A. Fiat and A. Shamir. *How to prove yourself. practical solutions to identification and signature problems*. In Andrew M. Odlyzko, editor, CRYPTO, volume 263 of Lecture Notes in Computer Science, páginas 186–189. Springer, 1987.

[7] Jerome H. Saltzer, David P. Reed, y David D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*: 277-288, noviembre 1984.

[8] Michael Shamos. *Paper v. Electronic Voting Records-An Assessment*, Abril 2004. http://www.electiontech.org/downloads/Paper%20vs%20Electronic.pdf

[9] Whitfield Diffie y Martin E. Hellman. New Directions in Cryptography. *IEEE Trans, Inform. Theory,* IT-221644-654, noviembre 1976.

[10] Taher El Gamal. *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*. *IEEE Trans. Inform. Theory*\ 31:469-472, 1985.

[11] Ronald L. Rives. *Self-Contained Paper-Based Cryptographic Voting.* In Roger Dingledine y Ting Yu, editores, *ACM Workshop on Privacy in the Electonic Society.* ACM, 2006.

1