J. A. Montenegro, J. Lopez, and R. Peralta, "Computacion Segura Multiparte Aplicada a Subastas Electrónicas", *IX Jornadas de Ingeniería Telemenatica (JITEL 2010)*, 2010.
NICS Lab. Publications: https://www.nics.uma.es/publications

Computación Segura Multiparte Aplicada a Subastas Electrónicas

José A. Montenegro, Javier López
Dpto. Lenguajes y Ciencias de la Computación
ETSI Informática Málaga. Universidad de Málaga
monte@lcc.uma.es, jlm@lcc.uma.es

Rene Peralta
Computer Security Division
National Institute of Standards and Technology.
peralta@nist.gov

***Resumen*—La confidencialidad ha pasado de ser un requisito de seguridad a ser considerado como requisito funcional y de obligado cumplimiento e inclusión en todos los sistemas de comunicaciones. Un inconveniente que presenta las técnicas criptográficas, utilizadas para obtener la confidencialidad de la información, surge cuando varias entidades se ven forzadas a compartir información secreta para realizar tareas puntuales de colaboración, ya que las primitivas tradicionales utilizadas para conseguir la confidencialidad resultan poco flexibles. La situación ideal permitiría hacer posible dicha colaboración sin que ninguna de las partes revele la información aportada. En este escenario entra en juego la tecnología de Computación Segura Multiparte (CSM) que posibilita realizar operaciones con la información compartida sin tener que hacerla pública. Este trabajo muestra una solución CSM aplicada a una subasta electrónica que permite la realización de la subasta sin que las apuestas sean reveladas a ningún participante, incluyendo el subastador, por lo que no necesita el estableciendo de ninguna autoridad confiable. Aunque la literatura ofrece una amplia variedad de propuestas teóricas de CSM desde su creación en la década de los ochenta, no es común su aplicación práctica en situaciones reales.**

***Palabras Clave*—Pruebas Discretas, Protocolo Conocimiento Cero, Cifrado Probabilístico, Computación Multiparte, Subastas Electrónicas.**

I. Introducción

La Computación Segura Multiparte (CSM), es un tipo de protocolo de comunicación que permite a una serie de participantes ponerse de acuerdo con el valor resultante de una función pública aplicada a sus datos privados, sin hacerlos públicos. Una amplia serie de problemas en e-commerce y e-government pueden ser resueltos aplicando CSM. Como ejemplo concreto de escenarios de aplicación destacamos los dos siguientes:

- Subastas públicas donde una organización gubernamental muestra el interés en subastar un bien público y las empresas participantes no quieren desvelar información sobre sus apuestas. Numerosos estudios establecen la convicción que la revelación de las cantidades subastadas podría perjudicar a las empresas en sucesivas subastas, así como revelar importante información sobre el estado financiero de la empresa.
- Voto electrónico donde los datos confidenciales a preservar son las decisiones de los votantes sobre los candidatos y el valor de la función representa el ganador de la elección.

La solución tradicional aplicada a ambos casos, ha sido la utilización de sobres de papel para preservar la información confidencial. Para realizar una transición de estas aplicaciones a un mundo electrónico, es necesario establecer un modelo criptográfico que simule el sobre tradicional. Aunque existen actualmente varias soluciones genéricas CSM, y que cumplen con una complejidad polinómica, desde el punto de vista de utilización de recursos computacionales pueden ser catalogadas de poco prácticas. Este es el principal motivo por el cual planteamos un nuevo modelo que pretende establecer una solución acorde con requisitos reales a este tipo de problemas.

Como marco de aplicación hemos seleccionado una subasta electrónica pública, concretamente la denominada subastas Vickrey. Este tipo de subasta recibe el nombre del ganador del premio Nobel de Economía en 1996, William Spencer Vickrey. Su característica más representativa, y que la distingue de los otros modelos de subastas, es que el ganador de la subasta es el usuario que realiza la mayor apuesta, sin embargo el precio del objeto subastado es el valor de la segunda apuesta más elevada. Esta simple característica tiene un impacto regulador en el precio de los elementos subastados, evitando que los postores paguen más por el elemento subastado que su valor real [24].

Aunque las subastas que siguen el modelo Vickrey aportan propiedades deseables en las subastas públicas, el diseño original no contempla la necesidad de mantener confidencial las apuestas de los usuarios. Por este motivo hemos considerado la necesidad de desarrollar una variación denominada como Subasta Vickrey a sobre cerrado, donde la apuesta de un usuario es desconocida por los restantes usuarios del sistema, revelando solamente las dos apuestas más elevadas de la subasta. Para conseguir dicha tarea haremos uso de CSM así como de elementos criptográficos adicionales que serán detalladas durante el trabajo.

El artículo es estructurado de la siguiente forma, la sección II establece una revisión en la literatura de las propuestas realizadas sobre subastas electrónicas. El diseño del sistema de subasta electrónica es descrito en la sección III, detallando los componentes del sistema que forman parte de él.

La sección IV realiza una breve introducción a la Computación Segura Multiparte y define la función utilizada para comparar dos apuestas confidencialmente. Debido a que las apuestas no son reveladas es necesario establecer un proceso de generación y verificación de retos sobre la función de comparación, así como también es necesario el establecimiento de una estructura de datos y un proceso de verificación de los retos. Para completar esta sección será presentado el concepto de Pruebas Discretas así como los distintos modelos de pruebas discretas que es posible

desarrollar: Pruebas Manuales, Pruebas No Interactivas y Pruebas No Interactivas Reducidas.

Los detalles sobre la implementación del sistema son especificados en la sección V, así como el rendimiento del sistema traducido al tiempo de creación y verificación de las pruebas en sus distintas modalidades. Para finalizar, la sección VI ofrece las conclusiones sobre el trabajo desarrollado y las líneas de trabajo futuras.

II. Trabajo Previo sobre Subastas a sobre cerrado

El diseño de una subasta segura o subasta a sobre cerrado ha sido una área de trabajo activa durante la última parte de los noventa y en la actual década. Esta situación fue estimulada por la decisión de la Comisión Federal de Comunicaciones NorteAmericana (FCC) de asignar las licencias en 1994 para el espectro electrónico mediante subastas competitivas y la necesidad de cumplir principalmente los requisitos de confidencialidad.

Desafortunadamente la mayoría de las soluciones propuestas hacen un mayor hincapié en la creación de nuevas primitivas criptográficas que en el diseño de un sistema seguro que mantenga las propiedades de seguridad de una subasta.

El artículo [9] puede ser considerado como el primer acercamiento para abordar los requisitos de las subastas electrónicas. El trabajo básicamente está orientado a la creación de una nueva primitiva criptográfica denominada *verifiable signature sharing*. A grandes rasgos, la primitiva habilita al titular de un mensaje firmado para compartir la firma entre un grupo de usuarios. Sólo los miembros de un grupo pueden reconstruir la firma. El principal inconveniente de esta propuesta es que todas las apuestas son reveladas a la finalización del periodo de apuesta.

Los autores en [17] diseñan una nueva primitiva denominada *oblivious transfer*, pero el diseño del sistema resultante puede ser catalogado como poco práctico. Esta primitiva fue mejorada en el trabajo [14] con la modificación de la primitiva y la creación de *verifiable proxy oblivious transfer*. Aunque la nueva propuesta fue diseñada para resolver un fallo de seguridad de la propuesta anterior, los autores hacen una declaración implícita de que la nueva primitiva requiere una cantidad excesiva de cómputo y de comunicación entre las partes involucradas.

Omote et al. presentan en [19] un esquema que hace uso de dos tipos de administradores de la subasta, uno que es el encargado del registro de los postores, y otro que administra la fase de apuestas. Únicamente la cooperación de ambas entidades y el ganador pueden decidir determinar cuál es la apuesta ganadora. La propuesta es una mejora del trabajo [2] que hace uso de cifrado homomórfico y de la técnica *mix match*. Aunque el trabajo ofrece una descripción técnica muy considerable, no incluye ninguna información sobre el diseño.

Durante el desarrollo de nuestra propuesta, Damgard et al. hicieron público el desarrollo de una subasta pública [6] basada en Computación Multiparte y *pseudorandom secret sharing* [5].

III. Diseño de un Sistema de Subasta Seguro

La creación de un sistema de subastas seguras requiere del diseño y desarrollo de determinados servicios, y los protocolos que conectan dichos servicios. El trabajo [15] determina un modelo de objetos y un diagrama de flujo de datos de una subasta que hemos considerado muy útiles para la realización de nuestro sistema.

Los componentes de la subasta deben cumplir los requisitos funcionales y de seguridad que establece la naturaleza de una subasta segura. En nuestro caso, además de los requisitos de seguridad nos centraremos en cumplir con los estándares, incluso si alguno de los servicios han de ser diseñados desde cero.

- Servicio de Aletoriedad: Tal y como conocemos los protocolos criptográficos actuales sería prácticamente imposible concebirlos sin la existencia de números aleatorios. El documento [10] es una buena referencia que demuestra el papel que juegan los números aleatorios en la criptografía. Además de la necesidad de este servicio como base para la creación de primitivas y protocolos, varios sistemas son diseñados bajo la asunción de la existencia de un generador de números aleatorios.
  La generación de números aleatorios puede llevarse a cabo mediante el uso de técnicas software o hardware. Normalmente los componentes hardware necesitan de máquinas dedicadas por lo que su utilización no resulta adecuada para los usuarios. Desafortunadamente no existe ningún estándar que define los métodos necesarios para llevar a cabo un servicio, tal como un protocolo de comunicación o la estructura de datos del protocolo.
  Por otro lado, la creación de números aleatorios basado en software es considerado como un proceso pesado desde el punto de vista computacional, por lo que usualmente hacemos uso de algoritmos más livianos como son los algoritmos pseudo aleatorios. Algunas vulnerabilidades están originadas debido a una incorrecta implementación, un uso impropio o un ataque al generador de número aleatorio o un ataque [13], [1], [11].
  Para la inclusión en nuestro sistema proponemos la creación de un servicio de aleatoriedad confiable que pueda ser creado mediante software, hardware o mediante un uso híbrido de componentes, basándonos en las recomendaciones de [7], [22].
  El servicio proporcionará números aleatorios certificados así como una prueba de la calidad del servicio. Es importante resaltar que los números que ofrece el servicio deben ser certificados para asegurar su origen.
- Servicio de Tiempo: La secuencia temporal de los eventos en una subasta tiene una influencia significativa en el rendimiento del sistema y en el desarrollo de la subasta. Normalmente, las apuestas son eventos controlados por los participantes, pero ciertos eventos importantes como la finalización del periodo de subasta implica la necesidad de incluir una fuente de tiempo confiable para determinar y verificar que las acciones han sido realizadas en un instante concreto de tiempo.
  El trabajo [25] establece un estudio en profundidad de los requisitos de tiempo real de las subastas

electrónicas. Además de la citada importancia que tiene el establecimiento de un servicio de tiempo en las tareas de la subasta, juega un papel muy importante en los protocolos criptográficos, debido a que cada paso debe ser ejecutado en un instante de tiempo preciso.
Al contrario que en el caso del servicio de aleatoriedad, el servidor y el protocolo no ha tenido que ser desarrollado, sino que específicamente hacemos uso el servicio de tiempo estándar proporcionado por National Institute of Standard and Technology (NIST)[1]. Actualmente existen tres protocolos disponibles para acceder a información relativa al tiempo. Time Protocol [21] y Daytime Protocol [20] proporciona un servicio básico y eficiente pero están considerados como obsoletos y además sus especificaciones no incluyen el soporte para los mecanismos de certificación. El tercer estándar Network Time Protocol (NTP) [18] mejora los dos anteriores estandares e incluye las propiedades de seguridad que lo hacen adecuado para nuestros requisitos.

- Servicio Clave Pública: Nuestra propuesta está basada en criptografía de clave pública, por lo que cada postor posee su par de claves criptográficas. Es bien conocido que la criptografía asimétrica necesita de la existencia de una autoridad de certificación. Estas entidades confiables vinculan la clave pública con la identidad del usuario. Varias soluciones de certificación han sido expuestas y principalmente difieren en el modelo de confianza, jerárquico o anárquico, y la estructura de los datos del certificado. En nuestro caso, siguiendo los estándares, hemos implementado solamente un nodo raíz de una Infraestructura de Clave Pública (PKI).
- Servicio de Tablón de Anuncios: Todos los servicios anteriores tienen una vinculación directa o indirecta con las propiedades de seguridad. El tablón de anuncios es un servicio impuesto por los requisitos del sistema de subasta, estando su diseño dirigido por sus propiedades.Básicamente, el tablón de anuncios es un repositorio confiable donde todos los datos relativos a la subasta son publicados. El principal objetivo es que todos los postores tengan acceso a la información generada en la subasta.
  Un requisito indispensable es que todas las apuestas sean publicadas en el tablón de forma cifrada en el momento que los postores la emitan y antes que el periodo de apuesta finalice, ya que, en caso contrario, el usuario no es considerado como un postor y no puede participar en las fases finales de resolución. Además, la apuesta publicada en el tablón será la información utilizada posteriormente como entrada de la función de comparación. El tablón de anuncio tiene una dependencia directa con el servicio de tiempo, debido a que es tan importante publicitar la información como su correcta datación.
  Una vez que la resolución de la subasta finaliza y son publicados los resultados en el tablón, comienza la fase de verificación donde las pruebas de veracidad de las funciones de comparación generadas por cada postor serán también publicadas en el tablón. A modo de resumen, la función principal del tablón es que todos los elementos de la subasta sean públicos a todos los participantes, para que todos los participantes posean la misma información y una posición igualitaria en la subasta.

[1]El servicio puede ser accedido en http://tf.nist.gov/service/its.htm y además, en la dirección http://tf.nist.gov/tf-cgi/servers.cgi podemos consultar los servidores de tiempo disponibles así como su estado en cada momento.

A. *Etapas de la Subasta*

Aunque normalmente una subasta tiene dos fases, la fase de apuesta y de resolución, hemos decidido incluir nuevas fases para hacer un mayor énfasis en cómo la seguridad incide en la subasta, de esta forma el sistema queda configurado con cinco fases:

- Configuración: El subastador establece los parámetros de la subasta y envía todos los datos al usuario después de que cada uno de ellos realice su proceso de autenticación. Básicamente, hay dos parámetros relativos a la subasta y otros dos relativos a los algoritmos criptográficos. Los dos primeros parámetros son relativos a la subasta, mientras que los restantes son parámetros de seguridad:
  - Intervalo del Precio: Establecemos el límite inferior (T1) y superior (T2) así como la cantidad de incremento (S). Esta información es utilizada, además de en la subasta, para optimizar la entrada de la función de comparación, reduciendo los bits necesarios para su representación y optimizando el circuito de comparación. Por tanto, las entradas válidas quedan reducidas al intervalo definido entre (T2-T1)/S.
  - Duración Subasta: Determina la finalización del proceso de apuesta. Después de este punto no es posible realizar más apuestas y los protocolos de seguridad comienzan a transmitir la información relativa a la seguridad.
  - Alfa: Este parámetro tiene relación con las pruebas criptográficas y por ende con la seguridad del sistema. Cuanto más elevado sea el valor de Alfa más seguridad tendrá el sistema resultante pero por contra más tiempo de cómputo es necesario. Por ello, es necesario configurar el valor apropiado de Alfa basándonos en los recursos del sistema. A modo de ejemplo, un valor de Alfa igual a 20 significa que el sistema es seguro con una probabilidad de $1 - 2^{-20} \sim 99,99\%$.
  - Método Matriz: Si seleccionamos esta opción, se utilizará una matriz booleana para reducir las pruebas criptográficas. Una descripción más detallada del proceso de creación y verificación de pruebas discretas reducidas es realizada en la sección IV-C3.
- Apuestas: Una vez que han sido establecidos los parámetros de la subasta por parte del subastador, los usuarios deben pasar un proceso de autenticación. Para ello, se han desarrollado dos métodos, un método básico basado en el par (usuario, palabra de paso) y un método de autenticación avanzado haciendo uso de las claves criptográficas del usuario y del servicio de números aleatorios.

- Finalización del tiempo de subasta: Después de este momento ningún usuario puede realizar una apuesta y pasamos a las fases de determinación y verificación. Como mencionamos anteriormente es extremadamente importante la sincronización de los relojes de los integrantes del sistema, y de ahí la importancia del servicio de tiempo.
- Determinación del Ganador y del Precio: Consideramos que nuestro sistema es una subasta a sobre cerrado ya que las apuestas son enviadas de forma cifrada. Debido a que hemos seleccionado la subasta Vickrey, la segunda apuesta ganadora es la que establece el precio de la subasta por lo que necesitamos dos rondas para determinar el ganador y el precio de la subasta. El proceso de determinación es un proceso de muestreo donde el servidor realiza una consulta a todos los usuarios sobre los apuestas válidas y recibe una respuesta de cada uno de ellos de forma positiva o negativa. El proceso de muestreo comienza en el valor más elevado (T2) que fue establecido previamente en el proceso de configuración, hasta encontrar los valores ganadores o finalizar con el valor más pequeño (T1), decreciendo la cantidad en cada paso el valor configurado determinado como (S). En el caso que un usuario conteste de forma afirmativa, el postor debe revelar la apuesta, abriendo así el sobre electrónico. Por tanto, en este punto, el usuario ejecuta la fase de revelación del protocolo de compromiso de bit para probar que el valor enviado anteriormente era correcto. Obtendremos un ganador cuando el proceso de verificación del compromiso de bit sea válido, y en ese momento comenzará el proceso de determinación del precio de venta. En caso contrario, el postor ha mentido por lo que el proceso de determinación de un ganador continúa sin el postor que ha mentido. Si estamos en la situación de empate, la subasta Vickrey se convierte en una subasta normal donde el ganador es el postor que realizó primero la apuesta y su apuesta es el precio de venta.
- Publicación y Verificación de las Pruebas: Mientras que la apuesta del ganador y el precio de venta son revelados en el etapa anterior, las apuestas restantes nunca lo son. Para probar la veracidad de las apuestas de los restantes participantes deben publicar sus pruebas discretas. La definición, contenidos, así como los procesos de creación y verificación de las pruebas discretas, son ampliamente descritos en la sección IV-C. Básicamente, en esta fase, los postores deben publicar la información necesaria para reconstruir los circuitos de comparación y realizar la verificación de las puertas del circuito sin revelar la cantidad apostada.

IV. Aplicación práctica de Computación Segura Multiparte

El protocolo de los Millonarios de Yao [27] puede ser considerado como el punto de inicio de la Computación Segura Multiparte. Todos los desarrollos teóricos sobre CSM fueron realizados en la década de los ochenta y actualmente sigue siendo un campo activo aunque ha sido retomado de forma más práctica. Ejemplos de ello son el trabajo sobre el cual escribimos, así como los resultados obtenidos en los trabajos [3], [6]. Como hemos mencionado anteriormente, el principal objetivo de nuestra propuesta es que las apuestas no sean reveladas en ningún momento, haciéndose solamente pública las apuestas que obligatoriamente requiere el proceso de subasta, como son la ganadora y la apuesta que establece el precio de venta. Para llevar a cabo este requisito aplicamos los conceptos de CSM. La principal virtud de CSM es que permite a los usuarios realizar tareas computacionales de forma distribuida sobre su información sin revelarla. El diseño de CSM debe cumplir los requisitos de confidencialidad y correctitud incluso si existe un ataque de una entidad externa o por un conjunto de participantes.

La confidencialidad es obtenida mediante la utilización previa de un protocolo de compromiso de bit donde cada usuario realiza un cifrado de su apuesta bit a bit. Los bits cifrados son utilizados como la entrada del circuito de comparación que describiremos en la siguiente sección IV-A. La propiedad homomórfica del criptosistema elegido permite que los valores de entrada sean distribuidos por el circuito de comparación sin perder en ningún momento la confidencialidad. Por otro lado, las pruebas discretas definidas en la sección IV-C permitirán verificar la corrección de la función de comparación de cada postor.

A. Creación de un Circuito que muestra que un número es menor que otro dado

Supongamos que un postor *B* ha realizado una apuesta *X*, siendo un número positivo. La representación binaria de *X* utilizando *n* bits es $x_{n-1}2^{n-1} + ... + x_1 2^1 + x_0 2^0$, por tanto $(x_{n-1}, ..., x_1, x_0)$ es un vector donde $x_i \in \{0, 1\}$. Dado un número *S*, también en representación binaria $s_{n-1}2^{n-1} + ... + s_1 2^1 + s_0 2^0$, el postor quiere construir una función de comparación tal que $S \geq X$. Para resolver este problema, primero definiremos un circuito cuya salida es 1 sii $S \geq X$.

Siendo $A = 2^n + S - X$ y su representación binaria como $(a_n, a_{n-1}, ..., a_1, a_0)$. Entonces a_n es 1 sii $S \geq X$. Además, *A* puede ser $A = S + (2^n - 1 - X) + 1 = S + \bar{X} + 1$ donde $\bar{X}$ es el complemento a bit de *X*. Por ello, a_n es simplemente el bit de acarreo n^{th} cuando sumamos S, $\bar{X}$ y 1. La representación de *A* utilizando puertas booleanas AND ($\wedge$), XOR ($\oplus$), y NOT ($\neg$) es:

$$F(S => X) = C_k \text{ donde } C_k = (S_k \oplus C_{k-1}) \wedge (\neg\ X_k \oplus C_{k-1}) \oplus C_{k-1}) \text{ y } C_0 = 1$$

aunque existe una ecuación equivalente, que ha sido la escogida, ya que causa un mejor rendimiento.

$$F(S => X) = \neg\ C_k \text{ donde } C_k = (\neg\ S_k \oplus C_{k-1}) \wedge (X_k \oplus C_{k-1}) \oplus C_{k-1}) \text{ y } C_0 = 0$$

B. Generación y Verificación de los Retos

La función de comparación es la piedra angular de la seguridad de nuestra propuesta. Cada postor tendrá su propio circuito $(PCircuit)$ cuya misión es básicamente demostrar a los demás participantes que no ha mentido en su apuesta y, por otro lado, tendrá acceso a los circuitos $(VCircuit)$ de los demás postores para verificar su veracidad.

Los nodos en el $(PCircuit)$ tendrán sus valores binarios y sus correspondientes compromisos de bit (en la práctica pueden tomarse como valores cifrados), mientras que en el $(VCircuit)$ solamente conoceremos los valores cifrados de los nodos. Por consiguiente, un circuito de comparación tendrá una instancia como $(PCircuit)$ y tantas instancias $(VCircuit)$ como usuarios quieran probar la función de verificación, sin riesgo a revelar la información.

Cada postor debe proveer a los verificadores los valores cifrados de las puertas AND, que junto a las apuestas cifradas que se enviaron en la fase de apuesta, permiten completar los valores cifrados del circuito. A nadie escapa la posibilidad de que el postor mienta sobre los valores cifrados de las puertas AND, por lo que es necesario establecer un mecanismo para conocer si esos valores son válidos y, por ende, el circuito de comparación construido. El mecanismo creado es un par de desafíos que permiten verificar si el postor conoce el valor binario de la salida y las entradas de las puertas AND del circuito, basándose en la elección de tres números aleatorios (A, B, C) que deben cumplir las propiedades detalladas en ambos retos.

El reto 1 se centra en la demostración de la salida de la puerta AND utilizando su compromiso de bit (bO) que permite que no se revele su información binaria. El postor X debe ejecutar los siguientes pasos para generar el desafío 1 que será verificado por el usuario Y:

- (A, B, C) debe cumplir, sin seguir ningún orden específico, que dos de ellos $(E0_1, E0_2)$ deben ser equivalentes al compromiso de bit bO de la salida de la puerta AND que estamos procesando.
- X envía (A, B, C) a Y.
- X envía las raíces a Y $\sqrt{E0_1 \times b0\, mod N}$ y $\sqrt{E0_2 \times b0\, mod N}$ donde N es la clave publica de X.

El usuario Y debe por tanto realizar el siguiente proceso para verificar el reto 1:

- Y calcula $U = (\sqrt{E0_1 \times b0 mod\, N})^2$ y $V = (\sqrt{E0_2 \times b0 mod\, N})^2$
- Y elige un número (γ) en (A, B, C) y calcula $\delta = bO \times \gamma$. Si δ es igual a U o V realiza el siguiente paso. En caso contrario X ha mentido.
- Y elige otro número distinto (β) en (A, B, C) y calcula $\theta = bO \times \beta$. Si θ es igual a U o V X no ha mentido.

Haciendo uso del reto 1 cualquier usuario puede verificar si la salida de la puerta AND era correcta, pero existe la posibilidad que el postor mintiera sobre las entradas de la puerta AND, enviando previamente el compromiso de bit inverso. Para tal caso debemos definir un nuevo reto.

El reto 2 verifica si el postor no mintió en las entradas de la puerta AND, siguiendo una técnica similar a la realizada en el reto 1, pero utilizando los compromisos de bit de la entrada (bI_1,bI_2). De esta forma, para establecer el reto 2, el usuario X debe realizar los siguientes pasos:

- (A, B, C) debe cumplir, sin seguir ningún orden específico, que dos de ellos (EI_1, EI_2) serán equivalente al compromiso de bit de las entradas bI_1 y bI_2 de la puerta que estamos procesando.
- X envía (A, B, C) a Y.
- X envía las siguientes raíces a Y $\sqrt{EI_1 \times bI_1 mod\, N}$, $\sqrt{EI_2 \times bI_2 mod\, N}$ y $\sqrt{EZero}$ donde N es la clave pública de X.

Los pasos utilizados para verificar el reto 2 son básicamente similares a aquellos que utilizamos en el reto 1 con algunas modificaciones.

- Y calcula $U = (\sqrt{EI_1 \times bI_1 mod\, N})^2$ y $V = (\sqrt{EI_2 \times bI_2 mod\, N})^2$
- Y elige un número (γ) en (A, B, C) y calcula $\delta = bI_1 \times \gamma$. Si δ es igual a U o V iremos al siguiente paso. En otro caso, Y repite el proceso asignándole a γ alguno de los otros dos número en S. Si al menos uno de los casos es válido iremos al siguiente paso. Si no podemos realizar ninguna asignación el usuario X ha mentido sobre la entrada bI_1.
- Y elige otro número diferente del paso anterior (β) en (A, B, C) y calcula $\theta = bI_2 \times \beta$. Si θ es igual a U o V (uno de los valores que no ha sido seleccionado previamente), podemos concluir que las entradas son correctas. Si la asignación no puede realizarse entonces X ha mentido en la entrada bI_2.

C. Pruebas Discretas

Los trabajos [12], [4] establecen el concepto de *Pruebas Discretas*. Este tipo de pruebas es considerado como un tipo nuevo de pruebas de conocimiento cero debido a que no puede clasificarse en ninguna de las categorías anteriores. Esta afirmación está basada en la creación de un conjunto de certificación, que contiene un vector de compromisos de bit que codifica un vector de bits en un conjunto dado sin revelar el vector de bits. Es por ello que las pruebas discretas constituyen una herramienta perfecta para establecer las verificaciones de los circuitos de comparación establecidos anteriormente.

Es posible establecer tres modelos de pruebas, las cuáles serán especificadas en detalle en las siguientes secciones, incluyendo en cada caso sus ventajas y desventajas de cada propuesta y qué contexto es más apropiada su aplicación.

1) Pruebas Manuales: La primera de las opciones disponibles son las pruebas Manuales. Este tipo de prueba es recomendable realizarla si existe una representación visual del circuito de comparación, ya que el usuario que realiza las comprobaciones debe interactuar con la información como veremos a continuación.

Las ventajas de este método es que el usuario que está realizando la verificación define la seguridad del sistema ya que la prueba puede ser ejecutada cuantas veces desee. Como inconveniente principal, además de la necesidad de

implementar un interfaz visual, este método requiere que los postores o un agente que lo represente este siempre online.

La figura 1 es una representación visual del protocolo que es necesario realizar para llevar a cabo las pruebas manuales. En la figura podemos observar que el verificador lanza el protocolo de verificación cuando selecciona una puerta AND del circuito. Básicamente, se basa en la ejecución de los retos 1 o 2 definidos anteriormente, según la elección del usuario.

Cada solicitud tiene asignada un identificador de sesión para evitar los deadlocks del protocolo ya que debemos recordar que es posible que cada usuario pueda actuar en distintos roles en el protocolo o incluso que varios usuarios verifiquen un mismo circuito a vez.

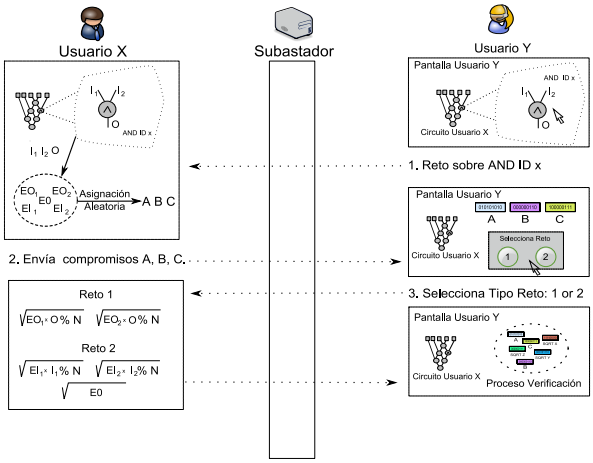

Fig. 1. Secuencia de mensajes generados en las pruebas manuales

La estimación de los mensajes intercambiados que son necesarios para probar la veracidad del circuito pueden ser calculados utilizando la siguiente ecuación:

$$mensaje = N\_user \times 4 \times ANDs\_circuit$$

donde N_user es el número de postores en la subasta, $ANDs\_circuit$ es el número de puertas AND del circuito de comparación y cuatro mensajes son los necesarios para completar cada prueba. Esta fórmula sólo tiene en cuenta la ejecución de un reto, por lo que sería el doble de mensajes si el verificador decide ejecutar ambos retos.

2) Pruebas No Interactivas: Las pruebas manuales hacen posible que el usuario establezca la seguridad del sistema, por tanto el nivel de confianza es administrado por el usuario a su merced. Aunque esta propiedad es la deseada por cualquier sistema de seguridad, su ejecución produce una considerable cantidad de mensajes en la red y además requiere que todos los participantes estén conectados para interactuar en los retos.

La pruebas No Interactivas han sido diseñadas con el objetivo de eliminar la dependencia en la interacción con los postores. Esta decisión implica que las pruebas tengan que ser automatizadas y por ello es necesario simular el comportamiento aleatorio del verificador. La solución viene por utilizar el servidor de aleatoriedad como fuente confiable de números aleatorios para simular que el usuario selecciona un reto especifico. Además, el proceso de automatización implica que el usuario no debe seleccionar cuántas veces el reto será aplicado a cada puerta AND, por lo que depende del parámetro Alfa (α) definido durante la fase de configuración.

El sistema será probablemente seguro con una probabilidad equivalente a $1 - 2^{-\alpha}$, por lo que grandes valores α darán lugar a un sistema más seguro, mientras el proceso de creación y verificación de las pruebas consumirá más tiempo de computación y por ende las pruebas tendrán un tamaño mayor. Teniendo en cuenta esta situación es necesario estimar su valor apropiado, ya que tendremos que tener en cuenta la capacidad de cómputo del dispositivo utilizado.

La figura 2 es un esbozo del proceso de creación de las pruebas interactivas. A grandes rasgos, el algoritmo para crear las pruebas no interactivas es el procesado de todas las puertas AND del circuito α veces y seleccionaremos de forma aleatoria el reto a aplicar en cada ocasión.

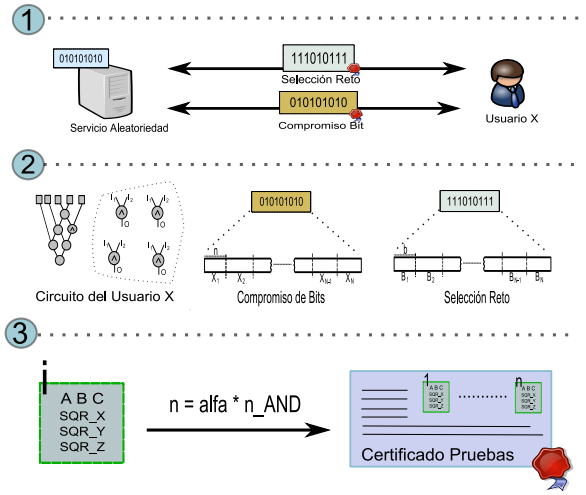

Fig. 2. Proceso de creación de pruebas no interactivas

3) Pruebas No Interactivas Reducidas: La complejidad de las pruebas no interactivas dependen directamente del número de puertas AND del circuito y del valor del parámetro de seguridad α. Introduciendo una pequeña modificación en las pruebas no interactivas la complejidad computacional y la carga en las comunicaciones puede ser reducida y además eliminar la dependencia con el tamaño del circuito. La solución pasa por la utilización de una matriz aleatoria de valores binarios para reducir el número de compromisos de bits generados. La figura 3 muestra un esquema del proceso de creación de las pruebas discretas no interactivas reducidas.

Inicialmente el objetivo de este método era reducir drásticamente la longitud de las pruebas criptográficas, pero su implementación nos constató el hecho que los procesos de creación y verificación reducidos consumen menos cómputo que el caso de las pruebas no reducidas. Esta situación se debe a que el tiempo consumido por la multiplicación de la matriz y el vector de retos es incluso menor que la ejecución de una raíz cuadrada.

V. Detalles de Implementación

El sistema de subasta descrito en el trabajo ha sido implementado utilizando la tecnología Java. La figura 4 muestra una captura de la aplicación. La selección de Java es justificada ya que hace posible un rápido desarrollo del prototipo, es un lenguaje multi-plataforma y posee una implementación eficiente de números grandes, requisito

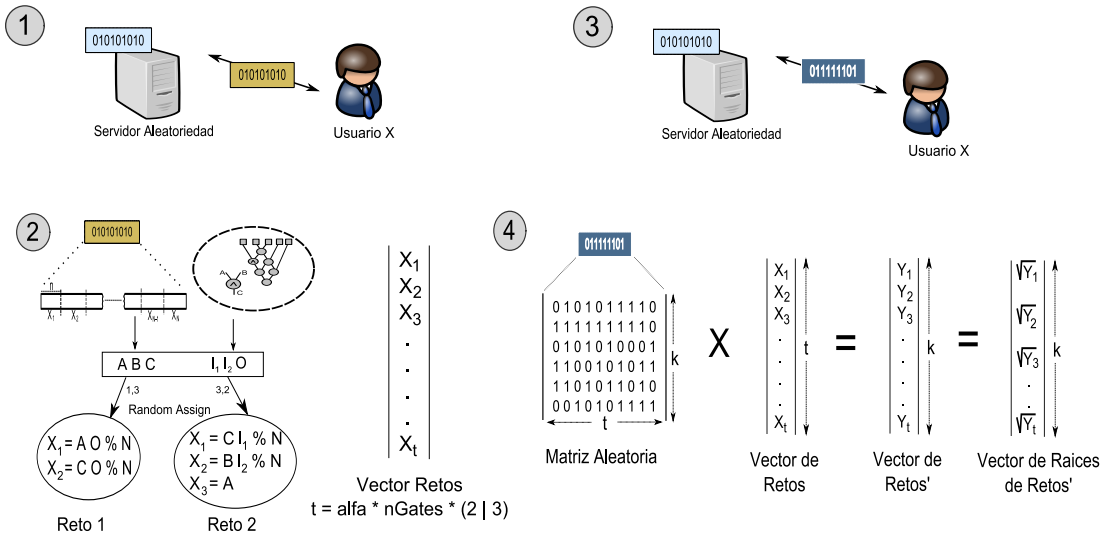

Fig. 3. Proceso de creación de pruebas no interactivas reducidas

necesario para el establecimiento de los protocolos y primitivas criptográficas.

El desarrollo del sistema cuenta además con la utilización de librerías de código abierto como es el caso de: $Derby$ para añadir soporte de base de datos SQL; $iText$ es utilizado para almacenar la información criptográfica en un documento con formato portable como es el caso de PDF; $jGraph$ es una librería gráfica que hace posible la descripción visual de la función de comparación, así como permitir la ejecución de las pruebas manuales (sección IV-C1) y $Barcode4J$ utilizada para representar grandes números en forma de matriz visual, así como facilitar la tarea de comparar dos grandes números.

El principal objetivo de la fase de desarrollo era cumplir con unos requisitos computacionales aceptables y un ancho de banda bajo. Después de un estudio detallado de los requisitos del sistema, determinamos que la carga del sistema está principalmente basada en los procesos de creación y verificación de pruebas. Más concretamente, podemos determinar que el proceso de creación es 13.52 veces más costoso computacionalmente que el proceso de verificación.

A modo de ejemplo nos centramos en una de las funciones más utilizadas en el sistema, que resulta ser el cálculo del número de Jacobi. Por desgracia, Java no incluye una implementación propia del número de Jacobi, por lo que ha sido necesaria su implementación teniendo en consideración un uso intensivo y la necesidad de que la aplicación cumpla con un rendimiento de tiempo real. Para su elaboración fue necesario comparar las distintas implementaciones existentes en la literatura. El trabajo [8] selecciona y compara los tres algoritmos más eficientes para calcular el número de Jacobi: Williams [26], Lesbegue [16] y Modified Binary [23]. Tenemos que remarcar que las implementaciones realizadas en el trabajo original no eran realizadas en Java, sino en la tecnología accesible en la época que el trabajo fue realizado, por lo que aunque en el trabajo se consideraba que el método *Modified Binary* era el más eficiente, la posterior implementación en Java de los tres algoritmos y la comparación de tiempos, mostró que la propuesta de *Williams* era la solución más eficiente para su implementación en Java.

Anteriormente hemos constatado el hecho de que el proceso de creación de pruebas depende del número de puertas *AND* en el circuito de comparación y del parámetro de seguridad *Alfa*. La figura 5 muestra los valores obtenidos en un test de rendimiento utilizando un circuito con 20 puertas, una clave de 1024 bits y diferentes valores de α entre el intervalo de 1 y 100. El mismo test ha sido realizado con una clave de 2048 bits y el tiempo computacional es incrementado *6.858* veces con respecto a la utilización de claves de 1024 bits. Como conclusión del test de rendimiento un incremento de la seguridad utilizando una clave de doble longitud, produce un sistema siete veces más lento.

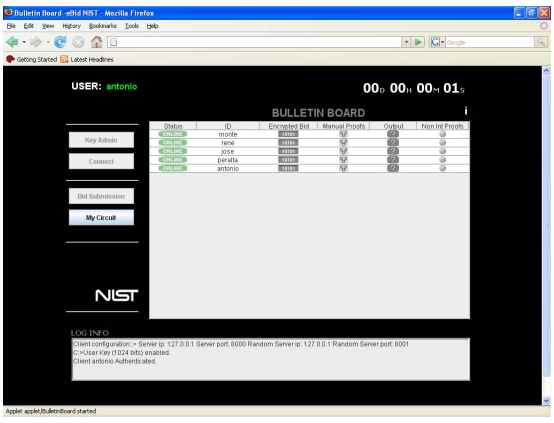

Fig. 4. Captura de pantalla de la aplicación del postor

Por otro lado, si optamos por utilizar las pruebas no interactivas reducidas, el tiempo de computación se reduce considerablemente. Concretamente, en la situación que requiere más computo en nuestro test (α = 100, puertas AND = 20), el método de pruebas no interactiva tarda aproximadamente unos 153,675 milisegundos, mientras que el método reducido sólo necesita unos 38,954 milisegundos, por lo que la utilización de la matriz consigue una reducción del tiempo de computo en casi cuatro veces.

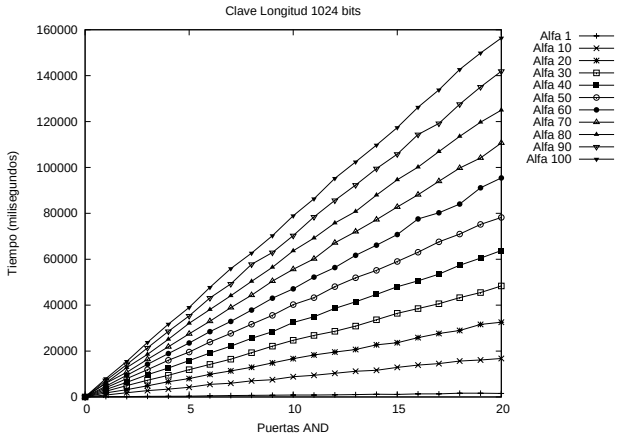

Fig. 5. Tiempo de procesamiento de las pruebas con una clave 1024 bit

VI. Conclusiones

El principal objetivo de esta investigación ha sido el desarrollo de un sistema de computación multiparte que cumpla con requisitos técnicos realistas, debido a que los sistemas de computación multiparte que existen en la literatura, desde la definición inicial de Yao, son soluciones difíciles de llevar a la práctica. Como banco de pruebas de nuestra propuesta nos hemos centrado en el diseño e implementación de una subasta electrónica a sobre cerrado, concretamente la denominada *sealed-bid Vickrey*.

Además, la inclusión de las pruebas criptográficas discretas permite verificar la veracidad de las apuestas de forma cifrada sin revelar su valor. La propiedad de discreta permite que la apuesta ganadora y el precio de venta puedan ser procesados sin que las apuestas perdedoras sean hechas públicas (incluso por el subastador). Los principales objetivos de la propuesta han sido utilizar la menor cantidad de recursos, que por su naturaleza son escasos, como tiempo de computación, ancho de banda de comunicación, número de bits comunicados, utilización de memoria, aleatoriedad e interacción (los protocolos con muchas rondas de comunicación son menos prácticos que aquellos que realizan pocas rondas).

Los tests realizados al sistema desarrollado muestran que las pruebas discretas son realmente prácticas y debido a la generalidad que prestan dicho método, nuestra investigación abre las vías a un gran número de aplicaciones sobre Internet en el terreno del *e-commerce* y *e-goverment* (elecciones, administración segura de informes médicos distribuidos, etc.).

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación a través de la concesión de una beca postdoctoral Fulbright (FU-2006-1527) al primer autor en Computer Security Division de National Institute of Standards and Technology y por el proyecto de investigación ARES (CSD2007-0004) del Ministerio de Ciencia e Innovación.

Referencias

[1] M. Szydlo A. Juels. A two-server, sealed-bid auction protocol. In *Financial Cryptography 2002*, pages 72–86, June 2002.

[2] M. Abe and K. Suzuki. M+1st price auction using homomorphic encryption. In *5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC*, volume 2274, pages 115–124, February 2002.

[3] P. Bogetoft, D. Christensen, I. Damgard, M. Geisler, T. Jakobsen, M. Krigaard, J. Nielsen, J. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft. Multiparty computation goes live. 2008.

[4] J. Boyar and R. Peralta. Short discrete proofs. In *EUROCRYPT*, pages 131–142, August 1996.

[5] I. Damgard and R. Thorbek. Non-interactive proofs for integer multiplication. In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515, pages 412–429, May 2007.

[6] I. Damgard and T. Toft. Trading sugar beet quotas - secure multiparty computation in practice. *Ercim News*, 73:32–33, 2008.

[7] D. Eastlake, S. Crocker, and J. Schiller. *Randomness Recommendations for Security*. IETF Network Working Group, request for comments: 4086 edition, June 2005.

[8] S. M. Eikenberry and J. P. Sorenson. Efficient algorithms for computing the jacobi symbol. *Source Journal of Symbolic Computation*, 26(1):509 – 523, 1998.

[9] M. Franklin and M. Reiter. The design and implementation of a secure auction service. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 302–312, May 1995.

[10] R. Gennaro. Randomness in cryptography. *IEEE Security & Privacy*, 4(2):64 – 67, March 2006.

[11] I. Goldberg and D. Wagner. Randomness and the netscape browser. *Dr. Dobb's Journal*, pages 66–70, January 1996.

[12] I. Damgard J. Boyar and R. Peralta. Short non-interactive cryptographic proofs. *Journal of Cryptology*, 138(4):449–472, 2000.

[13] D. Wagner J. Kelsey, B. Schneier and C. Hall. Cryptanalytic attacks on pseudorandom number generators. In *Fifth International Fast Software Encryption*, pages 168–188, March 1998.

[14] A. Juels and M. Szydlo. A two-server sealed-bid auction protocol. In *Financial Cryptography*, volume 2357, pages 72–86, March 2002.

[15] M. Kumar and S. Feldman. Internet auctions. In *Proceedings of the 3rd USENIX Workshop on Electronic Commerce*, pages 49–60, September 1998.

[16] V.A. Lebesgue. Sur le symbole (a/b) et quelques unes de ses applications. *J. Math. Pures Appl*, 12(1):497–517, 1847.

[17] B. Pinkas M. Naor and R. Summer. Privacy preserving auctions and mechanism design. In *1st ACM Conference on Electronic Commerce*, pages 129–139, November 1999.

[18] D. Mills. *Network Time Protocol (Version 3) Specification, Implementation and Analysis*. IETF Network Working Group, request for comments: 1305 edition, May 1992.

[19] K. Omote and A. Miyaji. A second-price sealed-bid auction with public verifiability. *Transactions of Information Processing Society of Japan*, 43(8):2405–2413, 2002.

[20] J. Postel. *Daytime Protocol*. IETF Network Working Group, request for comments: 867 edition, May 1983.

[21] J. Postel and K. Harrenstien. *Time Protocol*. IETF Network Working Group, request for comments: 868 edition, May 1983.

[22] A. Rukhin and et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Computer Security Division, NIST, special publication 800-22 edition, May 2001.

[23] J. Shallit and J. Sorenson. A binary algorithm for the jacobi symbol. *SIGSAM Bulletin*, 27(1):4 – 11, 1993.

[24] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8 – 37, 1961.

[25] M. Wellman and P. Wurman. Real time issues for internet auctions. In *1st IEEE Workshop on Dependable and Real Time E-commerce System (DARE)*, pages 54–56, June 1998.

[26] H. Williams. A modification of the rsa public-key encryption procedure. *IEEE Transactions on Information Theory*, 26(6):726 – 729, 1980.

[27] A. Yao. Protocols for secure computations. In *Symposium on Foundations of Computer Science*, pages 160–164, November 1982.