J. A. Onieva, J. Lopez, and J. Zhou, "Mejorando Servicios de Correo Electronico Certificado con Prontitud Temporal y Multicasting", *VIII Reunión Española sobre Criptología y Seguridad de la información (RECSI04). Avances en Criptologia y Seguridad de la Informacion*, pp. 537-546, 2004.
NICS Lab. Publications: https://www.nics.uma.es/publications

Mejorando Servicios de Correo Electrónico Certificado con Prontitud Temporal y Multicasting

No Author Given

No Institute Given

Resumen El correo electrónico certificado es un servicio añadido al correo electrónico estándar, en el cual el remitente desea obtener un recibo procedente del destinatario. Para este servicio, encontramos que los protocolos de intercambio (justo) son un componente principal para asegurar la corrección en la ejecución de los servicios de correo electrónico certificado, ya que los ítems que ambas partes presentan (en este caso específico, el mensaje de correo y el recibo del mismo) deben ser intercambiados sin que ninguna de las partes obtenga una ventaja durante el proceso sobre la otra. Podemos encontrar en esta línea de investigación protocolos optimistas eficientes para el intercambio electrónico, y mas concretamente para Correo Electrónico Certificado (CEC) y Firma Electrónica de Contratos (FEC). Realizando un estudio adecuado hemos observado que algunos aspectos de dichos protocolos podrían ser mejorados. En este artículo proponemos una solución que permite a ambas entidades terminar el protocolo de forma asíncrona. También extendemos el protocolo a múltiples usuarios.

Keywords: CEC, Intercambio Justo, Protocolo Multiparte.

1. Introducción

En el escenario de Comercio Electrónico (CE) actual, los usuarios pueden acceder a Internet para realizar compras de libros, entradas de cine o pujar por artículos en distintos sitios de subasta, pero el éxito de este tipo de plataformas electrónicas depende en gran medida de una propiedad muy importante para sus usuarios: *confianza*. Véase sino el caso de eBay, que se trata posiblemente del sitio de Internet más popular para la subasta de distintos artículos. Quizás, la razón principal de su éxito estriba en la existencia de un sistema de reputación entre los distintos usuarios que participan, sistema que se traduce en un modelo de confianza.

Sin embargo, esto no siemrpe será así. Dado que cada vez más compañías participan en actividades relacionadas con CE y redes peer to peer (P2P) se extienden en Internet, los usuarios demandarán más seguridad en sus conexiones y transacciones electrónicas. Por consiguiente, dado que el pilar sobre el que se

asienta una transacción económica es el intercambio de información, protocolos que aseguren la correción de ésta son necesarios.

Así, los Protocolos de Intercambio Justo (PIJ) son un mecanismo que asegura un intercambio de información entre dos entidades (computadoras, usuarios, etc..) sin que ninguna de ellas obtenga en el proceso una ventaja sobre la otra. Las aplicaciones más importantes de este tipo de protocolos son los CEC y FEC.

Este tipo de protocolos serían triviales si se pudiera presuponer que las entidades participantes se van a comportar siguiendo las reglas del protocolo. Pero esto no es realista en un mundo tan diverso como el CE. Por ello se recurre a Terceras Partes Confiables (TTP). Sin embargo, los protocolos que cuentan con su participación on-line son ineficientes, debido a que ésta ha de formar parte de cada ejecución. Adicionalmente, desde el punto de vista legal, la TTP estaría involucrada en la legalidad de cada intercambio (un peso demasiado caro).

El objetivo de este artículo son aquellas soluciones en las que la TTP solo ha de intervenir en caso de que se produzca alguna excepción entre las entidades participantes (por ejemplo un fallo en la red durante la ejecución del protocolo). En este campo, estos protocolos han sido denominados como PIJ optimistas [1,2,3,4,5], ya que presuponen el comportamiento honesto de las entidades participantes, sacando provecho de ello para involucrar a la TTP solo en caso de ser necesario. En este artículo utilizamos como base las soluciones propuestas por Micali en [6] y las extendemos con *prontitud temporal* asíncrona y un escenario multiparte.

El articulo comienza con una sección de introducción a los escenarios de intercambio y sus propiedades. En la sección 3 explicamos de forma breve la solución aportada por Micali para CEC. En la siguiente sección extendemos el protocolo con prontitud temporal asíncrona y con un escenario multiparte en la sección 5. La sección 6 concluye este artículo.

2. Introducción a los PIJ

2.1. Intercambio Electrónico

Muchas transacciones electrónicas pueden ser modeladas como un intercambio de información digital que involucra a dos o más usuarios que han acordado los items a intercambiar[1]. Por tanto, el problema de seguridad al que nos enfrentamos radica en como intercambiar estos items sin que ninguna de las entidades obtenga ventaja en el proceso. Como hemos mencionado en la sección anterior, la utilización de una TTP podría ser una solución ineficiente, mientras que por otro lado, como se sugiere en [7], al mismo tiempo es imposible lograr este objetivo sin la presencia de la TTP.

Podemos ver un esquema básico del Intercambio Electrónico en la figura 1, en el que, inicialmente, como ya hemos mencionado, ambas partes (los ya tradicionales usuarios Alice y Bob) poseen alguna información de los items a

[1] Ninguno de ellos conoce el item del otro extremo a priori, pero pueden verificar su autenticidad haciendo uso de mecanismos como las funciones hash

intercambiar (por ejemplo, el formato). Desde el instante en que Alice envía el item a Bob y hasta el último paso del protocolo nos encontramos en un estado no seguro. Así pues, el protocolo debe ser diseñado de tal forma que se reduzca esta situación incluso en caso de un fallo en la red o comportamiento inadecuado de cualquiera de las entidades. Será en estos casos en los que la participación de la TTP sea inevitable.

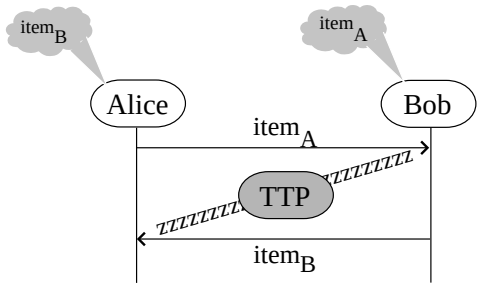

Figura1. Escenario Básico de Intercambio Electrónico

2.2. Propiedades del Intercambio Electrónico

En este apartado describimos los requerimientos que deben estar presentes en un PIJ. Algunos de ellos pueden ser opcionales, dependiendo de la aplicación en la que se integra el protocolo. Aunque *justeza* es probablemente el más importante, en algunas aplicaciones, todos ellos son importantes para el correcto funcionamiento del escenario.

- *Justeza*: Se dice que un Intercambio Electrónico proporciona justeza si ninguna de las entidades puede obtener ventaja terminando prematuramente el protocolo o modificando de alguna forma su consecución. Al final del protocolo (ver figura 1); bien Alice obtiene el $item_B$ y Bob recibe $item_A$, o ninguno de ellos obtiene ninguna información determinante.
- *Prontitud Temporal*: Se dice que un Intercambio Electrónico proporciona prontitud temporal si cualquiera de las entidades puede alcanzar el estado final del protocolo en una cantidad finita de tiempo (sin perder la propiedad anterior).
- *Confidencialidad*: Se dice que un Intercambio Electrónico proporciona confidencialidad si ninguna de las entidades puede tener acceso al contenido de los items durante el transcurso del protocolo.
- *No repudio*: Se dice que un Intercambio Electrónico proporciona no repudio si ninguna de las entidades puede negar al final de la ejecución del protocolo el haber enviado el item correspondiente.

Teniendo en cuenta que tanto Alice como Bob pueden actuar de forma deshonesta, el canal de comunicación entre ellos no es importante. Asumimos que el canal de comunicación entre la TTP y los usuarios no se encuentra pernamente roto; es decir, los mensajes enviados a y desde la TTP alcanzarán su destino en un tiempo finito.

3. Protocolo Optimista con tiempo límite para CEC

Explicamos brevemente el protocolo optimista que aparece en [6] para CEC. Cada usuario en el sistema tiene un identificador único. A, B y PO para Alice, Bob y la Oficina Postal (o TTP) respectivamente. Asumimos que todos ellos pueden firmar digitalmente mensajes haciendo uso de mecanismos de firma no-existencialmente falsificable a través de un ataque de texto elegido. La firma digital de una entidad X sobre un mensaje M se denota por $SIG_X(M)$, y asumimos por conveniencia que M es extraible de la firma digital. También asumimos que todas las entidades pueden cifrar mensajes haciendo uso de un algoritmo de cifrado asimétrico resistente a un ataque de texto cifrado elegido. Con $E_X(M)$, denotamos el cifrado de M con la clave pública de X [2].

A pesar de que Micali propone varios protocolos con diferentes propiedades, aquí explicamos el más completo que proporciona las propiedades de prontitud temporal, confidencialidad y justeza. Dado que se trata de un protocolo optimista, si ambas entidades se comportan de forma honesta, la TTP (referida como PO en el CEC) no se verá involucrada durante el desarrollo del mismo. Alice calcula el secreto $Z = E_{PO}(A, B, E_B(M))$ protegido con la clave de la TTP antes de enviar el mensaje de correo a Bob. Para preservar la propiedad de prontitud temporal, Micali propone una solución basada en un tiempo límite, en la que Alice elige un tiempo t después del cual la TTP no debería intervenir en la conclusión del protocolo.

1. $A \rightarrow B : SIG_A(t, Z)$
2. $B \rightarrow A : SIG_B(Z)$
3. $A \rightarrow B : E_B(M)$

Bob debe verificar la firma digital procedente de Alice, extrayendo ese límite de tiempo t y estimando si dispone de tiempo suficiente para contactar con la Oficina Postal en caso de que sea necesario (por fallo en la comunicación con Alice o mal comportamiento de ésta). Si definimos t_D como la máxima discrepancia de tiempo entre el reloj local de Bob y la PO, y éste recibe el paso 1 en un tiempo t_B de forma que $t_B + t_D$ es mayor o igual que t, entonces Bob detiene la ejecución del protocolo; en otro caso procede con el paso 2. Después de verificar la firma de Bob, Alice envía el mensaje M en el paso 3.

[2] Se asume que los mensajes se cifran directamente con un algoritmo de clave pública, pero de acuerdo con los estándares prácticos, M debería ser cifrado simétricamente con alguna clave y finalmente encriptar esa clave con un algoritmo de clave pública

Si después de responder, Bob no recibe el paso 3 en una cantidad razonable de tiempo, o $Z \neq E_{PO}(A, B, E_B(M))$, Bob ejecuta un subprotocolo de *resolución* con la PO:

$B \rightarrow PO$: $SIG_A(t, Z), SIG_B(Z), t$
IF ($t_{PO} < t$) AND (firmas validas)
  $PO \rightarrow B$: Y
  $PO \rightarrow A$: $SIG_B(Z)$

En este subprotocolo, la PO verifica las firmas y si la petición de Bob se produce con anterioridad al tiempo límite marcado por Alice (que puede ser extraido de su firma). En caso afirmativo, PO descifra Z con su clave privada y si el resultado de tal operación es una tupla que consiste en A, B y una cadena desconocida Y, envía ésta a Bob y reenvía la firma digital de Bob a Alice.

Como podemos ver, en este protocolo, incluso cuando la TTP interviene su participación es *transparente*; es decir, las evidencias obtenidas no se ven afectadas por esta participación.

4. Extensión de un protocolo CEC con Prontitud Temporal Asíncrona

Creemos que un tiempo límite no es una solución apropiada para asegurar la característica de prontitud temporal, por lo que proponemos una solución diferente, *prontitud temporal asíncrona*. En la propuesta que hemos visto hasta ahora, incluso si Bob calcula de forma aproximada, como se ha indicado, el tiempo necesario para acceder a la Oficina Postal, puede darse la situación en la que ésta sea inaccesible por un período de tiempo mayor. En ese caso, Bob podría no recibir el mensaje de correo de Alice, mientras ésta tiene en su poder el recibo de Bob. Esta situación sería tan complicada de resolver como dictaminar que entidad soporta la responsabilidad (legal en una aplicación de CE real) de lo sucedido.

Por ello introducimos un nuevo protocolo de *cancelación*, el cual junto al subprotocolo de *resolución* permiten a Bob finalizar el protocolo en cualquier instante. De la misma forma, Alice puede abortar el protocolo si no desea esperar a la resolución del mismo por parte de Bob.

El protocolo *principal* modificado (y el único que se ejecutará en caso de que ambas entidades se comporten de forma honesta y no se produzca un error en la red de comunicaciones existente entre ambos) quedaría de la siguiente forma:

1. $A \rightarrow B : Z$
2. $B \rightarrow A : SIG_B(Z)$
3. $A \rightarrow B : E_B(M)$

Así pues, el subprotocolo de resolución modificado que será lanzado por Bob bajo las mismas condiciones que el original es el siguiente:

$1'$. $B \rightarrow PO : h(Z), SIG_B(Z)$
$2'$. $B \leftarrow PO$: IF cancelado THEN $SIG_A(cancel, Z)$
  ELSE $E_B(M)$

El nuevo protocolo de cancelación es como sigue:

$1'.\ A \rightarrow PO : h(Z), SIG_A(cancel, Z)$
$2'.\ A \leftarrow PO :$ IF resuelto THEN $SIG_B(Z)$
ELSE *confirmado*

Usamos una función hash $h(Z)$ como una etiqueta para facilitar la búsqueda de Z por parte de la PO cuando uno de los usuarios participantes en el protocolo requiere su servicio. La Oficina Postal proporciona el acceso a los items de información que las entidades necesitan. Las entidades podrán acceder en cualquier instante a este servicio que presta la PO y no es su responsabilidad si los usuarios acceden y obtienen los items esperados. La Oficina Postal debe mantener un estado y necesita administrar (dentro de un período de tiempo razonable de acuerdo a una política de servicio) una tabla con las tuplas $(h(Z), estado)$ y las firmas correspondientes, lo que provoca que el servidor que implemente este servicio debe proporcionar la seguridad necesaria para esta información almacenada.[3].

Aunque en [6] no existe una mención específica al proceso de resolución de disputas, éste debe ser definido para cualquier PIJ. En este proceso ambas partes deben acordar un juez o árbitro (presumiblemente electrónico) que, basado en la evidencia recibida, evalúe el resultado de la disputa. De esta forma, si Bob niega haber recibido un mensaje de correo M haciendo uso de este protocolo, Alice debe proporcionar $(Z, SIG_B(Z))$ y el árbitro resuelve que Alice envió un mensaje M a Bob si:

1. $Z = E_{PO}(A, B, E_B(M))$;
2. La firma de Bob sobre Z es válida;
3. Bob no puede proporcionar $SIG_A(cancel, Z)$.

Asumimos la utilización de un algoritmo de cifrado público determinista, o en otro caso, Alice no podría deshacerse de la semilla aleatoria durante el protocolo (por ejemplo, si usamos un criptosistema como ElGamal [8]).

5. Extensión a un protocolo CEC Multiparte

En la actualidad podemos encontrar escenarios en los que la participación de distintas entidades en una determinada aplicación puede ser de gran utilidad. CEC y FEC son dos ejemplos. Enviar un mail a varios usuarios o tener un contrato firmado de forma digital por un grupo de usuarios son extensiones útiles en dichas aplicaciones. De esta forma proponemos una solución basada en el protocolo de la sección anterior en la que el remitente puede distribuir de forma certificada un mensaje M a distintas entidades. Notación adicional necesaria para la explicación es la siguiente:

- $Header$: una cabecera que indica en que posición el destinatario del mail tiene que buscar su información (ejm. $[B_i, i]$).

[3] Al igual que no podemos tener acceso a las diferentes cartas que los distintos usuarios intercambian en la actualidad haciendo uso de los sistemas de correo tradicionales

- B : conjunto de destinatarios.
- M : mensaje que se envía desde Alice a los recipientes B.
- $A \Rightarrow B : Y$: Alice realiza un broadcast de la cadena Y al grupo B.
- B' : un subgrupo de B que responde a Alice con la evidencia de recibo.
- $B'' = B - B'$: un subgrupo de B con el que Alice quiere cancelar el protocolo.
- $B''\_finished$: un subgrupo de B'' que ha finalizado el intercambio con un subprotocolo de resolución.
- $B''\_cancelled = B'' - B''\_finished$: un subgrupo de B'' con el que PO ha cancelado el intercambio.
- $u_B = u_{B_1}, u_{B_2}, ...$: grupo de claves públicas de B.
- $E_B(M) = E_{B_1}(M), E_{B_2}(M), ...$: concatenación de cifrados de M para el grupo B.
- $Z = E_{PO}(A, B, E_B(M))$: un secreto Z con la clave pública de PO.

La Oficina Postal electrónica (actuando como TTP en este caso) participará en los subprotocolos a petición de forma mutuamente exclusiva; es decir, la ejecución de los subprotocolos por parte de la PO se hace de forma atómica para cada usuario. A continuación, describimos el protocolo (ver figura 2):

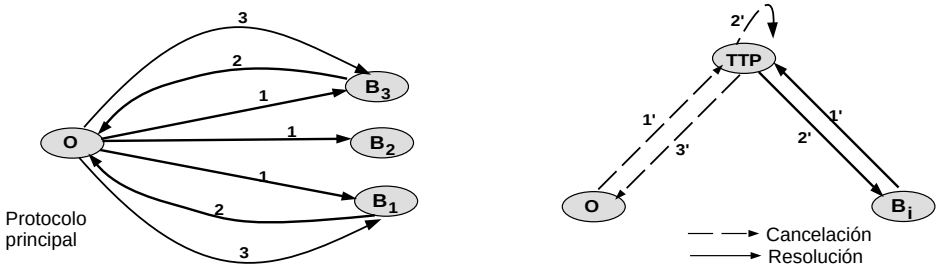

Figura2. Protocolo CEC Multiparte

El protocolo *principal* ejecutado por los usuarios es el siguiente:

1. $A \Rightarrow B \ : Header, u_B, Z$
2. $B_i \rightarrow A : SIG_{B_i}(u_{B_i}, Z)$ donde cada $B_i \in B$
3. $A \Rightarrow B' : E_{B'}(M)$

En el paso 1, Alice envía el secreto Z así como las claves públicas de todos los destinatarios participantes, de forma que si cualquier participante no está de acuerdo con su clave seleccionada (debido, por ejemplo, a que el correspondiente certificado fue revocado recientemente) puede detener la ejecución del protocolo. Si no es así, tras verificar los datos obtenidos, cada destinatario envía evidencia

de recibo a Alice en el paso 2; la cual en respuesta, envía el mensaje M (cifrado) al subconjunto de destinatarios que respondieron en el paso anterior.

En el caso en el que Alice no reciba un mensaje válido procedente de algunos de los destinatarios B'' en el paso 2, puede iniciar el siguiente subprotocolo de cancelación:

$1'.\ A \rightarrow PO : h(Z), SIG_A(cancel, B'', Z)$
$2'.\ PO$ FOR $(\bigvee i\ B_i \in B'')$
IF $(B_i \in B''\_finished)$ THEN obtiene $SIG_{B_i}(u_{B_i}, Z)$
ELSE añade B_i a $B''\_cancelled$
$3'.\ A \leftarrow PO : \bigvee i\ SIG_{B_i}(u_{B_i}, Z),\ SIG_{PO}(B''\_cancelled, Z)$

En este caso Alice comunica a la Oficina Postal su intención de revocar el protocolo con las entidades que aparecen en B'' y tras verificar la petición de Alice, la PO comprueba que entidades resolvieron previamente el protocolo y obtiene sus evidencias de recibo. Entonces, la PO genera una evidencia de cancelación para el resto de destinatarios e incluye toda la información en un mensaje destinado a Alice.

En caso de que algun destinatario B_i no reciba el mensaje 3, o éste no sea válido, B_i iniciará el siguiente subprotocolo de *resolución*:

$1'.\ B_i \rightarrow PO : Header, h(Z), u_{B_i}, SIG_{B_i}(u_{B_i}, Z)$
$2'.\ B_i \leftarrow PO :$ IF $(B_i \in B''\_cancelled)$ THEN $SIG_{PO}(B''\_cancelled, Z)$
ELSE $\{E_{B_i}(M)$
añade B_i a $B''\_finished$ y almacena $SIG_{B_i}(u_{B_i}, Z)\}$

El destinatario envía a la Oficina Postal electrónica toda la información que posee procedente de Alice junto a su evidencia de recibo. Si esta entidad no pertenece al grupo de entidades con el que Alice ha cancelado el protocolo, entonces la PO verifica las firmas digitales y descifra Z al mismo tiempo que almacena $SIG_{B_i}(u_{B_i}, Z)$. No debe ser posible para el destinatario inducir a la PO para que revele el mensaje de correo si el protocolo ha sido cancelado en su caso. Por esta razón la PO debe verificar la firma de los destinatarios así como la integridad de Z en el primer paso.

En otro caso, la PO envía una evidencia de cancelación al destinatario de forma que éste pueda fácilmente demostrar, en caso de disputas, a un árbitro (electrónico) que el intercambio fue cancelado.

Así pues, si B_i niega haber recibido M, Alice puede presentar la evidencia $Z, E_B(M), SIG_{B_i}(u_{B_i}, Z), SIG_{PO}(B''\_cancelled, Z)$ y el árbitro resolverá que el destinatario recibió el mensaje M de Alice si:

1. $E_{B_i}(M)$ computado con u_{B_i} pertenece a $E_B(M)$;
2. $Z = E_{PO}(A, B, E_B(M))$;
3. La firma de B_i en Z y su clave pública de cifrado es valida;
4. La firma de PO en $SIG_{PO}(B''\_cancelled, Z)$ es válida y $B_i \notin B''\_cancelled$.

Alice resolverá satisfactoriamente la disputa si todos los chequeos anteriores son positivos. En el caso en que todos sean afirmativos excepto el último

y Alice no pueda presentar evidencia de cancelación, entonces el árbitro debe interrogar a B_i. Si éste no puede presentar $SIG_{PO}(B''\_cancelled, Z)$ en el que $B_i \in B''\_cancelled$, Alice también ganará la disputa. En otro caso, B_i puede repudiar el haber recibido el mensaje M. Así pues, podemos observar que la evidencia proporcionada por la Oficina Postal es *autocontenida*; es decir, no es necesario contactar a la PO para la resolución de una disputa.

6. Conclusiones

Hemos descrito brevemente la solución de Micali a correo electrónico certificado destacando que la propiedad de prontitud temporal propuesta en el artículo original puede no ser suficiente para que la termincación del protocolo para ambas entidades se realice de forma segura. Aunque añadimos un nuevo subprotocolo de cancelación, el protocolo principal (el único que se ejecuta en el caso de que aparezcan situaciones anormales) no aprecia apenas cambios. Los mensajes adicionales que aparecen en el flujo de información del nuevo diseño mejoran la propiedad de prontitud temporal de forma elegante y eficiente.

También se ha propuesto una extensión a correo electrónico certificado multiparte para la distribución del mismo mensaje (mail) a distintas entidades, de forma que solo aquellas que contestan con evidencia de recibo pueden recibir el mensaje en claro. La propiedad de prontitud temporal también se ha considerado y aunque la PO tiene que actuar con múltiples entidades en caso de que un subprotocolo se inicie, no se ha introducido una complejidad considerable.

En ambos casos se proporciona un proceso de resolución de disputas entre las entidades participantes, ya que este proceso debe estar suficientemenete definido para que un árbitro electrónico pueda resolver a favor de alguna de las entidades haciendo uso de las evidencias presentadas por ellas.

Referencias

1. Asokan, N., Schunter, M., Waidner, M.: Optimistic protocols for fair exchange. In: Proceedings of the 4th ACM conference on Computer and communications security, ACM Press (1997) 7–17
2. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures. IEEE Journal on Selected Areas in Communications **18** (2000) 593–610
3. González-Deleito, N., Markowitch, O.: An optimistic multi-party fair exchange protocol with reduced trust requirements. In: Proceedings of the 4th International Conference on Information Security and Cryptology. Volume 2288 of Lecture Notes in Computer Science., Springer-Verlag (2001) 258–267
4. Zhou, J., Gollmann, D.: An efficient non-repudiation protocol. In: PCSFW: Proceedings of The 10th Computer Security Foundations Workshop, IEEE Computer Society Press (1997)
5. Zhou, J.: Non-repudiation in electronic commerce. Computer Security Series. Artech House (2001)
6. Micali, S.: Simple and fast optimistic protocols for fair electronic exchange. In: Proceedings of the twenty-second annual symposium on Principles of distributed computing, ACM Press (2003) 12–19

7. Pagnia, H., Gärtner, F.C.: On the impossibility of fair exchange without a trusted third party. Technical Report TUD-BS-1999-02, Darmstadt University of Technology, Department of Computer Science, Darmstadt, Germany (1999)
8. ElGamal, T.: A public-key cryptosystem and a signature scheme based on discrete logarithms. In: IEEE Transactions on Information Theory. Volume IT-31. (1985) 469–472