J. Areitio, and J. Lopez, "Mecanismo de Seguridad de Bajo Costo para Micropagos", *VI Reunion Española de Criptología y Seguridad de la Información (VI RECSI)*, pp. 193-200, 2000.
NICS Lab. Publications: https://www.nics.uma.es/publications

MECANISMO DE SEGURIDAD DE BAJO COSTO PARA MICROPAGOS

Areitio Bertolín, Javier (jareitio@orion.deusto.es); López Muñoz, Javier (jlm@lcc.uma.es)

Dpto. Telecomunicaciones, Facultad de Ingeniería. ESIDE. Universidad de Deusto

E.T.S. de Ingeniería Informática. Universidad de Málaga

Resumen

La presente comunicación presenta un mecanismo de micropagos flexible, de bajo costo que puede utilizarse para realizar pagos en línea entre el cliente y el vendedor y fuera de línea con el agente de negocios. Este mecanismo evita grandes almacenamientos de datos y cálculos largos. Se puede implantar en software para el cliente y en hardware/software para el vendedor.

Palabras clave: *Criptografía, Micropagos, Claves, Funciones hash, MAC.*

1. INTRODUCCION

En la actualidad existe un creciente interés en torno a los pagos electrónicos sobre redes como Internet. La definición del protocolo SET por parte de un grupo de proveedores de tarjetas de crédito demuestra este interés. Entre la variedad de mecanismos de pago que se han propuesto recientemente varios abordan la cuestión muy específica de los micropagos [1],[2],[3],[4],[6]. Dichos pagos surgen en el contexto de Internet cuando un usuario que navega desea acceder a recursos que requieren de un pequeño pago. Según Rivest y Shamir [4], los micropagos precisan gran eficiencia para ser viables económicamente. Esto significa que el empleo directo de la criptografía de clave pública no es lo más adecuado, incluso los criptosistemas convencionales como el DES/TDES/IDEA pueden ser cuestionables. Por tanto, la mejor elección parece apuntar hacia el empleo de las funciones resumen/unidireccionales (ó "hash"), posiblemente con clave que se utilizarán como MACs (Message Authentication Codes). Las tres partes de un mecanismo genérico de micropagos son:

1. El cliente (C) que desea acceder al recurso contra el pago del mismo.

2. El proveedor de servicios ó vendedor (V) que ofrece el servicio y necesita que se le pague.

3. El agente de negocios (A) que ofrece un soporte para la transacción

En el contexto de micropagos, se suele asumir que las comunicaciones con el agente de negocios son de largo período (diarias ó semanales con los vendedores y mensuales con los clientes). Por tanto, la criptografía "de alto costo" puede utilizarse en este nivel (por ejemplo, TDES/IDEA, RSA, etc.). Por su parte, las transacciones entre clientes y vendedores son frecuentes y para estas comunicaciones en este mecanismo sólo se utilizarán MACs (Códigos de Autenticación de Mensajes) que son "de bajo costo"; de hecho, se utiliza una función unidireccional con clave (hash). Cuando se adopta una criptografía donde sólo se permiten algoritmos "de bajo costo" (en términos de potencia de computación, carga de comunicación, ...) es necesario examinar el nivel de seguridad resultante. Existen dos cuestiones principales:

a) En el lado del vendedor, el riesgo de gastar por exceso: un cliente puede utilizar los derechos concedidos por el agente de negocios para comprar más de lo que se le concedió originalmente.

b) En el lado de cliente, el riesgo de que un atacante robe sus derechos utilizando p.e. un

“sniffer” y/o ser impropiamente facturado por el agente de negocios.

También debería considerarse el criterio de robusto en el caso de que un cliente pierda su clave secreta. El esquema presentado guarda algo de similitud con el mecanismo de Rivest y Shamir (que utiliza criptografía de clave secreta, tanto en el lado del cliente como en el lado del vendedor) sin embargo ofrece mayores garantías en el lado del cliente introduciendo un protocolo de pago basado en “desafios” que aunque muy simple, protege de posibles copias fraudulentas de los tokens del cliente. Este mecanismo sólo necesita MACs y Funciones hash por ello es de bajo costo.

2. DESCRIPCION DE COMPONENTES

Supondremos que los vendedores disponen de un dispositivo a prueba de falsificación para validar los micropagos, por ejemplo, una tarjeta inteligente, una tarjeta PCMCIA, etc.. Cada agente de negocios puede decidir distribuir su dispositivo específico ó compartirlo con otros (por ejemplo, utilizar un dispositivo proporcionado por un consorcio de bancos). Supondremos que las comunicaciones entre el agente de negocios y el dispositivo del vendedor y entre el agente de negocios y el cliente son seguras, tanto en confidencialidad como desde el punto de vista de la integridad. Esto puede realizarse con esquemas criptográficos fuertes como por ejemplo algoritmos de firma digital y de cifrado. Supongamos que el dispositivo posee una memoria permanente (pequeña) interna y una memoria externa (mayor) que no necesita estar físicamente segura. La memoria interna posee dos registros S(A,c) y S(A,d) que son las sumas globales de crédito y débito de todas las transacciones que el agente de negocios A autoriza. Además contiene información relativa a la transacción previamente fallida ó abortada para alertar de los posibles intentos de acceso fraudulentos. La memoria externa posee muchos registros r(c,T) y r(d,T) para todo token T que se utilice para pagar al vendedor V. Cada registro tienen la forma: r(i, T)=(fecha, I(d,T), S(i,T), MAC(KA)[fecha, I(d,t),S(i,T)]), donde "fecha" representa la fecha en que se ha actualizado el registro y KA es la clave del agente de negocios A para el token T. La figura 1 muestra la representación gráfica del protocolo de pago para una cantidad "z". Los elementos del mismo son:

(1) Inicialización del vendedor. El agente de negocios A fija su propia clave secreta KA y la comunica de forma segura al dispositivo de cada vendedor. El dispositivo inicializa sus registros a cero y limpia la memoria externa.

(2) Protocolo de gastar. Para permitir que pague un cliente C, A genera un token T que es una cadena de bits con la forma: I(d,T)=[número de token][fecha de expiración][I(d,A)] junto con una clave utilizada para gastar KT=MAC(KA)[token, I(d,T)], donde "token" representa una cadena de bits fija que se utiliza para evitar las interacciones malas entre varios cálculos de MAC con tipos diferentes. La cadena I(d,A) es un identificador del agente de negocios. Haciendo esto, el agente de negocios autoriza al cliente a gastar una cantidad de dinero dada con la clave KT. Como ya se ha indicado, la relación entre el agente de negocios A y el cliente C es confiable, de modo que el control de la cantidad gastada se deja al cliente C. Además, debe ser tal que el fraude sistemático pueda detectarse y el cliente deshonesto debe recogerse en una lista negra. De este modo, el token T será asociado privadamente al cliente C en la base de datos del agente de negocios A.

(3) Protocolo de pago. Cuando se desea gastar una cantidad "z" a favor de un vendedor V, el cliente C se presenta al vendedor con un token T. El cliente también elige un número aleatorio (denominado "desafío") "nc" y lo comunica al vendedor. Entonces, el vendedor proporciona (I(d,T), z, nc) a su dispositivo que genera un "desafío" (ó número aleatorio) "nv" que se envía a C. El dispositivo registra que una transacción sucedió empezando con (I(d,T), z, nc, nv) para más tarde detectar si se abortó ó no. El vendedor también envía la cadena de identificador I(d,D) de su dispositivo. El cliente entonces revela una "compra": g=MAC’(KT) [compra, I(d,T), I(d,D), z, nc, nv] donde “compra” es una cadena de patrón fijo y el vendedor V interroga a su dispositivo con [I(d,t),g] que verifica si: g = MAC’(MAC(KA)<token,I(d,t)>[compra,I(d,t),I(d,D),z,nc,nv] ó no, los valores z, nc y nv se leen de la memoria. Si el pago tiene éxito, el registro S(c,A) se incrementa por “z” y el

registro r(c,T) se pide de la memoria. Si no existe, el dispositivo creará uno nuevo inicializado a cero. El registro S(c,T) lo incrementa en "z" el dispositivo y lo guarda en memoria externa.

(4) Cancelación. De forma similar, una compra a partir del token T puede cancelarla el vendedor utilizando una clave de cancelación. En este caso el dispositivo incrementa S(d, A) y S(d,T) por "z" de una forma similar.

(5) Autorizar el pago. El vendedor V regularmente envía al agente de negocios A la cantidad gastada por su cliente. Pregunta al dispositivo para que le envíe S(c,A) y S(d,A) y el vendedor envía todos los registros r(c,T) y r(d,T). El agente de negocios A verifica la consistencia de los contadores, es decir, que los valores MAC son correctos, que las fechas de los registros se encuentran entre la última fecha de autorización y la fecha actual y que la suma de todos los S(c,T) es S(c,A) y la suma de todos los S(d,T) es S(d, A). Entonces, el agente de negocios A paga por S(c,A)-S(d,A) e incrementa cada contador S(T) por S(c,T)-S(d,T). El contador S(T) es el dinero gastado por el token T. Si el token T ha gastado por exceso, se contacta con el cliente correspondiente para que proporcione las explicaciones oportunas.

3. CUESTIONES DE VULNERABILIDAD

Suponiendo que la transmisión de KA entre el agente de negocios y los vendedores es segura, la única forma de recuperar KA que permite crear dinero es deducirla de las demás informaciones. Por ejemplo, un cliente deshonesto puede intentar obtener KA de la ecuación KT=MAC(KA)[token, T]. La función MAC debe resistir este tipo de ataques Se recomienda una longitud de clave de 100 a 160 bits para KA. Un atacante puede intentar gastar dinero sobre otra cuenta de cliente T. Si C guarda su clave secreta KT, el atacante sólo puede escuchar clandestinamente la comunicación de pagos. Por tanto criptoanalizará KT de la ecuación: g= MAC'(KT)[compra, I(d,T), I(d, D), z, nc, nv] ó pretende gastar sin conocer KT, pero debe responder a un "desafío" actualizado "nv". Como el cliente real escogió un número aleatorio (ó desafío) nc antes de obtener nv, el impostor debe remitir un número nc' y el vendedor se dará cuenta. Para el MAC' la salida puede ser menor que la de MAC y la longitud de clave para KT puede ser de menos de 128 bits. Si un cliente intenta gastar por exceso la clave, será detectado por el protocolo de autorización. Si un atacante intenta reencaminar la comunicación entre C y V hacia otro vendedor V' haciendo que C pague, esto puede evitarse utilizando el identificador I(d, D) del dispositivo del vendedor real en la respuesta g.

4. CONCLUSIONES

Se ha sintetizado un mecanismo de pagos flexible que puede implementarse utilizando primitivas criptográficas sencillas. Se ha supuesto que los proveedores de servicios disponen de un dispositivo a prueba de falsificaciones seguro por los agentes de negocios. El mecanismo presentado es una solución de bajo costo al problema de los micropagos realizando un nivel de seguridad significativo para el cliente.

REFERENCIAS

[1] AREITIO, J. *Transacciones Electrónicas Seguras: Consideraciones sobre Certificados Digitales, Autoridades de Certificación y Terceras Partes Confiables.* VIII Congreso Securmática. Abril (1997). 386-396.

[2] AREITIO, J. *Elementos de Seguridad para el Comercio Electrónico sobre Internet.* REE. Ediciones Técnicas REDE. Septiembre (1997). 86-96.

[3] AREITIO, J. *Aplicación de la Criptografía para Proteger el Comercio Electrónico.*

Vco. GM2 Publicaciones Técnicas. Noviembre (1997). 42-46.

[4] RIVEST, R.L. SHAMIR, *A. PayWord and Micromint: two simple micropayments schemes.* CryptoBytes. Vol 2, Num. 1. (1996). 7-11.

[5] SET *(Secure Electronic Transactions Specifications).* Draft July (1996).http:// *www.mastercard.com/set*.

[6] JUTLA,C.S. JUNG,M *Paytree Amortized signature for Flexible Micropayments.* AsiaCryp'96. (1996). 751-758.