

MICROMINT

di

Aibangbee Pamela

Zugarini Chiara

MicroMint è un protocollo atto a effettuare transazioni a basso costo (micropagamenti), sviluppato sull'idea di Millicent da Rivest e Shamir.

Il risultato più rilevante di questo metodo è la completa eliminazione delle operazioni a chiave pubblica (chiave crittografica utilizzata in un sistema di crittografia asimmetrica) dando più importanza alla velocità a discapito della sicurezza.

La scelta di dare meno importanza alla sicurezza del sistema è giustificata dal fatto che MicroMint è un sistema di micropagamenti (operazione o servizio che consente il trasferimento di piccole quantità di denaro), quindi la perdita di piccole somme di denaro in piccola scala non è un grave danno.

La "moneta" di MicroMint è prodotta da un broker che la vende a degli utenti, gli utenti danno questa moneta ai venditori come pagamento, i venditori restituiscono le monete al broker che ne rimborsa l'ammontare attraverso altri mezzi.

Monete

Una moneta è una stringa di bit la cui validità può essere facilmente constatata da ognuno, ma che è difficile la produrre (eccetto che dal broker). Generare più monete risulta più conveniente di generarne poche, è necessario un cospicuo investimento iniziale per coniare la prima moneta ma le successive vengono prodotte con molta convenienza (simile all'economia per una coniazione reale, per la quale si investe molto per acquistare macchinari costosi che consentono poi di produrre monete in modo economico).

Le monete di MicroMint sono rappresentate da collisioni di funzioni hash, per alcune specifiche funzioni hash ad una via che trasformano stringhe x di m -bit in stringhe y di n -bit.

Diciamo che x è la contro immagine di y se $h(x)=y$.

Data una coppia distinta di stringhe (x_1, x_2) a m -bit è detta collisione a due vie se $h(x_1)=h(x_2)=y$

Per qualche y di n -bit. Per aumentare ulteriormente la difficoltà di conio da parte di possibili falsari si propone di utilizzare collisioni a k -vie.

Una collisione a k -vie è un insieme di valori distinti x , (x_1, \dots, x_k) che hanno lo stesso valore hash y .

Il numero di valori x che devono essere esaminati prima di vedere la prima collisione è circa $2^{n(k-1)/k}$. Se si esaminano c volte questi valori di x , per $1 \leq c \leq 2^{n/k}$, ci si aspetta di vedere circa c^k collisioni a k -vie.

Quindi scegliendo $k > 2$ si ha l'effetto di accelerare la velocità di generazione delle collisioni una volta trovata la prima collisione, oltre ad incrementare la sicurezza del sistema.

Coniatura

Produrre monete è come lanciare casualmente delle biglie in 2^n contenitori: k biglie in un contenitore costituiscono una moneta. Quindi per battere moneta il broker creerà 2^n contenitori e lancerà approssimativamente $k2^n$ biglie e creerà una moneta da ogni contenitore che conterrà k biglie.

N.B: con questa scelta di parametri ogni biglia ha probabilità pari a $1/2$ di far parte di una moneta!

Se un contenitore ha più di k biglie al suo interno il broker può estrarre in principio un sottoinsieme di k elementi in vari modi e produrre parecchie monete. Si consiglia di non produrre più di una moneta dallo stesso contenitore in quanto un falsario potrebbe combinare due monete e crearne una da uno stesso contenitore per crearne altre, in questo modo si semplifica il compito del broker di identificare monete a spesa multipla.

In questa descrizione di base esiste però un problema: la memorizzazione dei dati è di gran lunga più onerosa della computazione.

Il numero di biglie che possono essere lanciate supera abbondantemente il numero di quello che può essere memorizzato da un hard-disk e il numero di quelle di cui ha realmente bisogno il broker. Per trovare un giusto equilibrio possiamo pensare di rendere inutilizzabili alcune biglie per la coniazione. Per fare ciò diciamo che una biglia è “buona” se i bit di maggior peso del valore hash y hanno un valore z specificato dal broker.

Per essere più precisi:

siano t ed u per cui $n=t+u$, dove n è il numero di bit del valore hash y .

Allora se gli t -bit più significativi di h hanno valore z allora il valore y è buono e gli u -bit di y determinano il valore del contenitore nel quale la biglia x è stata lanciata.

In questo modo il broker lancia $k2^n$ biglie, ne memorizza $k2^u$ buone e genera da queste $\frac{1}{2} 2^u$ monete valide.

Vita di una moneta

- **Distribuzione di monete:** verso la fine di ogni mese il broker cominci a vendere monete agli utenti per il mese successivo. All’inizio di ogni mese il broker rivela il nuovo criterio di validità delle monete. Il broker tiene traccia delle monete distribuite ai singoli utenti. Un utente acquista monete addebitando l’acquisto su carta di credito. Le monete non utilizzate vengono restituite al broker alla fine di ogni mese in favore di altre valide.
- **Pagamenti:** un utente che deve pagare un acquisto al venditore spedisce a questo una moneta (x_1, \dots, x_k) non spesa in precedenza. Il venditore verifica la validità della moneta controllando l’hash su ogni valore della serie e verificando che sia uguale per tutti.
- **Riscatto delle monete:** ogni giorno (ha comunque facoltà di farlo quando vuole) il venditore restituisce le monete che ha accumulato al broker, questo controlla che le monete non siano già state riscattate: per le monete valide paga il venditore (meno la tassa di brokeraggio), per quelle non valide sceglie di pagare uno solo dei venditori da cui ha ricevuto la stessa moneta. In questo modo si penalizzano i venditori, ma si scoraggiano gli stessi ad effettuare truffe ai danni del broker restituendo monete che hanno raccolto altri venditori.

Sicurezza

Distinguiamo tra attacchi a piccola e larga scala. Si suppone che utenti e venditori non abbiano interesse ad imbrogliare per guadagnare pochi centesimi anche perché l’attacco risulta dispendioso se paragonato all’obiettivo. I meccanismi di sicurezza sono studiati per scoraggiare tre tipi di attacchi:

- Falsificazione
- Furto di monete
- Spesa doppia

Falsificazione

La falsificazione a piccola scala è così costosa da non destare interesse nel falsario.

La falsificazione a larga scala può essere rintracciata e contrastata nel seguente modo:

- Le monete false diventano automaticamente non valide alla fine di ogni mese.
- Le monete false possono essere generate solo dopo che il broker annuncia il nuovo criterio di validità mensile della moneta all'inizio del mese.
- L'utilizzo dei predicati nascosti che forniscono un intervallo di tempo più adeguato a respingere monete false senza compromettere la validità legale delle monete già in circolazione.
- Il broker può rintracciare la presenza di un falsario verificando che le monete ricevute corrispondano effettivamente a contenitori da cui ha generato moneta.
- Il broker può dichiarare qualsiasi momento che il periodo corrente è concluso, richiamare tutte le monete del periodo corrente e emettere nuove monete utilizzando una nuova procedura valida.
- Il broker può generare simultaneamente monete per molti mesi successivi; il tentativo di falsificazione può cominciare solo una volta che le condizioni di validità del mese vengono annunciate, il falsario così avrà il doppio problema di cominciare tardi e di essere troppo lento rispetto al broker pur utilizzando le stesse risorse. Produrre monete in anticipo non crea problemi di memoria o risposta al broker, in quanto le monete prodotte in anticipo possono essere lasciate temporaneamente in un nastro magnetico non costoso perché non ha bisogno di rispondere velocemente a utenti che richiedono già queste monete.

Furto di monete

Il furto durante la distribuzione iniziale di monete agli utenti e il riscatto da parte dei venditori si combatte con la criptazione delle monete in quanto le relazioni utente/broker e venditore/broker sono relativamente stabili e quindi è possibile concordare delle chiavi.

Per evitare l'utilizzo di tecniche a chiave pubblica si utilizzano monete *specifiche per l'utente*, e per evitare che due venditori si mettano d'accordo in modo da riscattare le stesse monete si possono rendere le monete *specifiche per il venditore*.

- Monete specifiche per l'utente: l'utilità delle monete specifiche sta nel fatto che se rubate non sono utilizzabili da terzi. Una prima tecnica consiste nel dividere gli utenti in gruppi e consegnare ai membri del gruppo monete la cui validità dipende dall'identità del gruppo (applicare alle monete del gruppo un'opportuna funzione hash ausiliaria h' specifica per il gruppo, cosicché il venditore può verificare la giusta provenienza della moneta). Questa tecnica può creare problemi se il gruppo è troppo grande (è facile trovare acquirenti per monete rubate) o troppo piccolo (il broker è costretto a preparare una quantità di monete superiore alle sue reali esigenze). Una seconda tecnica consiste nell'acquisto da parte dell'utente di un supercontenitore, preparato dal broker in maniera non specifica per ogni utente. A questo punto

sarà l'utente che al momento della creazione di moneta applicherà una sua specifica funzione hash; una caratteristica vantaggiosa di questo schema è che è facile produrre un numero elevato di monete per un dato utente.

- Monete specifiche per il venditore: inizialmente si è pensato che l'utente potesse dare le monete ai venditori in modo tale che ogni moneta potesse essere riscattata solo da una piccola frazione dei fornitori. Questa tecnica rende una moneta rubata meno desiderabile, dal momento che è improbabile che venga accettata da un fornitore diverso da quello dove è stato speso. L'ulteriore controllo di validità può essere effettuato sia dal venditore e dal broker. L'ovvia difficoltà è che né il broker né l'utente possono prevedere in anticipo di quale venditore l'utente sarà cliente, ed è irragionevole forzare l'utente ad acquistare in anticipo monete specifiche per ogni possibile venditore. Una seconda idea mette in relazione le monete specifiche del venditore a quelle specifiche per l'utente. In pratica il venditore non fa altro che applicare una sua specifica funzione hash alle monete dell'utente, come una sorta di firma.

Spesa doppia

Cos'è? Tentativo di pagamento o riscatto multiplo con la stessa moneta.

Una truffa a piccola scala di questo tipo è difficile da identificare, ma a causa del basso valore delle singole monete, non è poi così importante se sfugge all'identificazione.

Per quanto riguarda le truffe a larga scala ricordiamo che il broker tiene traccia delle monete assegnate, quindi di quante monete a spesa doppia sono associate ad ogni utente e venditore. Con queste informazioni il broker può far decadere un sospetto truffatore dal sistema (il servizio non è più disponibile per il dato utente o venditore).

BIBLIOGRAFIA:

- *“PayWord and MicroMint: Two simple micropayment schemes”* di Ronald L. Rivest e Adi Shamir;
- <http://theory.lcs.mit.edu/~rivest>