1. Introduzione

Negli ultimi decenni abbiamo assistito ad un costante e progressivo sviluppo delle tecnologie multimediali che, rapidamente , sono entrate a far parte di moltissimi aspetti della nostra vita quotidiana.
Uno dei campi dove le tecnologie multimediali si stanno affermando sempre di più è quello degli scambi commerciali, infatti, un numero sempre maggiore di utenti, famiglie e imprese ha iniziato ad utilizzare internet, che è diventato un potenziale mercato in cui acquistare o vendere informazioni e prodotti.
Al fine di rendere possibile la conclusione di transazioni commerciali tramite questa rete si sono sviluppati vari sistemi di pagamento su internet.

I vari pagamenti elettronici

I sistemi di pagamento elettronici possono essere classificati come segue:

- sistemi di pagamento tramite carta di credito:

Si appoggiano sull'infrastruttura di gestione delle transazioni tramite carta di credito. La transazione avviene trasmettendo un numero di carta di credito al commerciante, utilizzando algoritmi crittografici per gestire riservatezza, integrità e non ripudio allatransazione;

- assegni elettronici e borsellino elettronico:

Si basano entrambi sul sistema bancario corrente.
Nel primo il cliente stipula una convenzione con l'istituto emittente ed emette quindi un
assegno sottoscritto con firma digitale, che può così essere presentato alla banca per l'incasso.

Con il borsellino elettronico si intende invece una carta prepagata e ricaricabile, dalla quale viene sottratta di volta in volta la somma spesa per effettuare una transizione;

• e-cash (o moneta elettronica in senso stretto):
Si tratta di un sistema tramite il quale una società emette crediti spendibili in rete,
dietro pagamento della somma equivalente da parte dell'acquirente. Il contante digitale (e- cash) è stato pensato per unire i vantaggi dei comuni sistemi di pagamento elettronici (sicurezza e convenienza) con quelli del denaro di carta (la riservatezza e l' anonimato).

2. La sicurezza delle reti

La sicurezza nella archiviazione e nella trasmissione dei dati richiede di adottare adeguate misure per proteggere i dati da intrusioni o da un loro utilizzo diverso da quello previsto dai legittimi possessori e/o operatori. Questo problema, ha acquisito una connotazione particolare grazie all'uso dei computer e delle reti di trasmissione, e si è esteso alla protezione di dati di qualsiasi natura. La tecnica di base adottata a tale scopo viene denotata con il termine generale di crittografia. Attualmente le due forme più diffuse di crittografia sono dette crittografia simmetrica e crittografia asimmetrica, cui sono strettamente legati argomenti quali la firma digitale e la certificazione. Il problema ha di recente assunto maggior peso con riferimento ai sistemi di pagamento in rete.

L'utilizzo della rete per lo scambio di informazioni sensibili, come quelle relative alle transazioni per il commercio elettronico, deve tenere in conto il problema della sicurezza, al fine di garantire che le transazioni avvengano con un livello di sicurezza paragonabile a
quella delle transazioni convenzionali. Questo è possibile tramite l'implementazione di un set di funzionalità nel punto di accesso tra il

sistema che offre i servizi (rete locale o server) e la rete esterna, che, secondo lo standard ISO Security Architecture, possono essere ricondotte alle seguenti cinque classi:

a) Autenticazione
b) Controllo degli accessi
c) Riservatezza
d) Integrità
e) Non ripudio

Questo set di funzionalità è necessario e sufficiente a garantire che gli scambi di informazioni tra il sistema e la rete sia protetto.

a) Autenticazione

L'autenticazione si preoccupa di verificare che chi richieda un servizio è effettivamente quello che dichiara di essere. Tutti gli strumenti utilizzati per l'autenticazione si basano sul concetto che chi deve autenticarsi debba dimostrare di avere qualcosa che è a lui riservato. Questo qualcosa può essere o un'informazione o un oggetto in suo possesso esclusivo.

b) Controllo degli accessi

Una volta che, grazie all'autenticazione, si è certi dell'identità di chi accede al sistema, è possibile limitare l'accesso degli utenti alle sole risorse che sono state ad esse assegnate.

• Nel caso di computer, il controllo degli accessi si riferisce tipicamente a risorse del filesystem della macchina (directory su disco, file, driver di periferiche), permettendo di controllare l'accesso in termini di blocco totale dell'accesso, permesso di sola lettura, permesso di scrittura, permesso di esecuzione, ecc.

• Nel caso di una rete, invece, il controllo dell'accesso si applica alla visibilità delle macchine connesse in rete e dei servizi da queste offerti. E'

possibile, tramite firewall, decidere verso quali macchine lasciar transitare i pacchetti dati, verso quali programmi in esecuzione su queste macchine, quali comandi permettere verso i programmi raggiungibili (es.: solo scrittura di dati, lettura di dati, ecc.).

c) Riservatezza

Una volta che è stato possibile accedere ad un servizio, è necessario che le informazioni che transitano tra l'utente ed il sistema non siano utilizzabili da terzi. Sebbene sia importante evitare che nessun tipo di informazione sia estraibile dal traffico di rete (es.: la sola trasmissione di un pacchetto da parte di un sistema di anti-intrusione potrebbe far trapelare l'informazione che è stata rilevata qualche violazione del sistema), la riservatezza viene generalmente interpretata come protezione crittografica dei dati scambiati.

d) Integrità

Indipendentemente da aspetti di riservatezza, i dati scambiati debbono poter raggiungere la destinazione senza modifiche da parte di terzi. Infatti, anche per informazioni che possono rimanere pubbliche (come un listino prezzi), è necessario poter verificare la coincidenza di quanto ricevuto con quanto trasmesso e rilevare qualsiasi alterazione dei dati. Queste problematiche sono state ampiamente affrontate nel settore delle telecomunicazioni e la soluzione più comune è quella di trasmettere, oltre ai dati utili, informazioni aggiuntive codice di controllo) che possano rilevare alterazioni sui dati scambiati, tipicamente per effetto di errori di trasmissione. C'è, però, una sostanziale differenza tra l'uso del controllo di integrità nelle trasmissioni e nella protezione delle informazioni. Nelle trasmissioni, il codice di controllo può viaggiare non protetto. Nella sicurezza, invece, è necessario fare in modo che solo il mittente possa generare questo codice proteggendolo opportunamente. Se così non fosse, chiunque potrebbe modificare il messaggio e ricalcolare un nuovo codice, eliminando quelli originali. Anche per l'integrità, la soluzione è nell'utilizzo di algoritmi crittografici.

e) **Non ripudio**

Per transazioni di rete, è necessario garantire che le parti che intervengono in un determinato scambio non possano poi negare di aver preso parte allo stesso. L'esigenza è la stessa che si ha per transazioni convenzionali e che si può risolvere tramite l'utilizzo di firme (su contratti, distinte, ecc.). Nel caso di transazioni elettroniche, la soluzione è nell'uso della firma digitale, anche questa basata su algoritmi crittografici.

Sicurezza delle comunicazioni

Delle cinque classi (sopra elencate) corrispondenti alle funzionalità richieste per la sicurezza di un sistema, l'autenticazione, la riservatezza e l'integrità riguardano aspetti generali di protezione delle comunicazioni, indipendentemente dalle applicazioni interessate dallo scambio dati. E' infatti possibile implementare queste funzionalità proteggendo l'avvio di una sessione ed i singoli pacchetti dati, ignorando completamente il campo di applicazione ed il contenuto dello scambio informativo. Di conseguenza, queste funzionalità possono essere fornite da apparati di rete (es.: firewall) o da programmi generici (es.: browser web).

Sicurezza delle applicazioni

Delle cinque classi (sopra elencate) corrispondenti alle funzionalità richieste per la sicurezza di un sistema, il controllo degli accessi ed il non ripudio sono legate alle applicazioni interessate dallo scambio dati. Le modalità con le quali eseguire il controllo degli accessi sono molto variegate e dipendono dal sistema che si vuole controllare (computer, rete,...), dal sistema operativo utilizzato (Windows, Unix, ...), dall'applicazione, ecc. Per quanto riguarda il non ripudio, le modalità con le quali eseguire la firma elettronica sono ormai standardizzate e le infrastrutture necessarie per il suo utilizzo ampiamente diffuse. I campi di applicazione vanno dalla protezione delle comunicazioni interpersonali (con il supporto da parte dei principali software di posta elettronica) alla

protezioni delle transazioni (come i pagamenti con carta di credito o moneta elettronica).

Certificati digitali

I certificati digitali, rappresentano quello che i documenti d'identità costituiscono nella vita reale; servono per stabilire con esattezza, in una comunicazione, l'identità delle parti. Essi sono dei file, con una validità temporale limitata, usati per garantire l'identità di un soggetto, sia esso un server o una persona, vengono utilizzati ogniqualvolta ci siano problemi di sicurezza del tipo:

1. Quando si forniscono o si utilizzano servizi on line come pagamenti e consultazione di dati riservati.

2. Quando si scambiano massaggi di posta elettronica: il mittente che compare su una e-mail non ci assicura riguardo all'identità di chi ha spedito veramente quel messaggio.

3. Quando vogliamo verificare la validità di documenti in formato elettronico scaricati da internet o vogliamo garantire l'autenticità di documenti da noi pubblicati.

Tutte queste problematiche vengono affrontate e risolte con l'utilizzo dei certificati digitali e di altre tecnologie ed essi collegate.
I certificati digitali vengono rilasciati dalle cosiddette Autorità di Certificazione (Certification Authority, solitamente abbreviato con C.A.), oppure alcuni enti detti “Autorità di rilascio certificati autonoma" possono rilasciare direttamente i propri certificati. Una Autorità di Certificazione rilascia i certificati a chi ne fa richiesta dopo averne attestato l'identità. Svolge il ruolo di garante dell'identità di chi usa il certificato da lei rilasciato, così come le autorità di pubblica sicurezza (prefettura, comune, etc…) che emettono documenti di identificazione quali il passaporto o la carta d'identità. Chiunque può verificare la validità

di un certificato, in quanto le C.A. devono mantenere un pubblico registro dei certificati emessi e una Lista dei Certificati Revocati (Certification Revocation List) disponibile per la verifica per via telematica da parte di tutti gli utenti.

Protocollo SSL

Il protocollo SSL provvede alla sicurezza del collegamento garantendo tre funzionalità fondamentali:
• **Privatezza del collegamento**. La crittografia è usata dopo un handshake iniziale per definire una chiave segreta. Per crittografare i dati è usata la crittografia simmetrica (e.g. DES,RC4,etc.).

• **Autenticazione.** L'identità nelle connessioni può essere autenticata usando la crittografia asimmetrica, o a chiave pubblica (per es. RSA,DSS,etc). In questo modo i clients sono sicuri di comunicare con il corretto server, prevenendo ogni interposizione. E' prevista la certificazione sia del server che del client.

• **Affidabilità**. Il livello di trasporto include un check dell'integrità del messaggio basato su un apposito MAC (Message Authentication Code) che utilizza funzioni hash sicure (e.g. SHA, MD5, etc.). In tal modo si verifica che i dati spediti tra client e server non siano stati alterati durante la trasmissione. SSL è un protocollo aperto e non proprietario; è stato proposto da Netscape Communications al W3 Consortium come un possibile futuro approccio standard alla sicurezza per i browsers WWW e per i servers.
Lo scopo primario del Protocollo SSL è fornire riserbo ed affidabilità alle comunicazioni.

3. Protocollo ikp

Che cos'è?
Il protocollo iKP (*Internet Keyed Payment Protocol*), sviluppato dalla IBM, è un prototipo di sistema di pagamento su Internet basato su carta di credito. È stato la base dello standard SEPP di Mastercard, poi abbandonato in favore del nuovo standard SET, in cooperazione con VISA. iKP può facilmente essere usato per implementare un sistema di assegni elettronici.

iKP è stato progettato per:

• Ottenere un alto livello di integrità per tutte le parti coinvolte, tenendo conto delle differenze di rischio e di esigenze tra una parte e l'altra.
• Fornire riservatezza nelle transazioni economiche.
• Lavorare con il minimo impatto sui sistemi finanziari esistenti.

Pur fornendo quanto necessario per pagamenti sicuri, iKP:

• Non consente alcuna trattativa su modalità di pagamento, prezzo ecc.: contiene una semplice procedura di contratto ("offerta/ordine").
• Non fornisce la non tracciabilità dei pagamenti (ma protegge dal venditore i dati del compratore).
• Non fornisce mezzi per una distribuzione sicura di informazioni: fornisce ricevute di pagamento ma non le protegge.

Impostazioni di base

Come in tutti i sistemi di pagamento elettronico le parti interessate alle transazioni economiche sono: compratore, venditore, acquirente, fornitore.
Tuttavia, nel sistema iKP, le parti direttamente coinvolte sono tre: il compratore, il venditore ed il *gateway dell'acquirente*.

Il sistema di pagamento è gestito da un'organizzazione tipo Mastercard, VISA. Tali enti hanno relazioni fisse di affari con certe banche che agiscono da fornitore di carta di credito per il compratore e da acquirente dei pagamenti per il venditore. Ogni fornitore ha un BIN (Bank Identification Number), che riceve al momento in cui stipula il contratto con l'organizzazione che gestisce il sistema, e che è in rilievo su ogni carta di credito fornita, come parte del numero di carta di credito. Il BIN identifica inoltre l'organizzazione che gestisce il sistema. È molto importante notare la presenza del *gateway* (dell'acquirente): tale entità funziona da interfaccia tra il "mondo elettronico" e l'infrastruttura per pagamenti già esistente. Il gateway autorizzerà le transazioni usando proprio tale infrastruttura: la rete commerciale di compensazione/autorizzazione per carte di credito.
Il protocollo sfrutta le solite primitive crittografiche.

Per l'autenticazione di un messaggio iKP usa:

- Firma digitale;
- Codifica dei segreti con protocollo a chiave pubblica di tipo *plaintext-aware*;

La codifica di tipo *plaintext-aware* è un modo di operare che assicura l'integrità dei messaggi, ed è sicuro sotto ragionevoli ipotesi.
Al contrario della firma digitale, tale metodo non consente la risoluzione di contestazioni.

Per la riservatezza**,** iKP sfrutta due meccanismi:

- I dati segreti che devono essere verificati dal destinatario ma che non necessitano di essere trasmessi, sono nascosti sfruttando funzioni *salted hash* (per esempio un numero *N* può essere nascosto nella funzione *h(N,x)*, dove *x* è un valore random noto a mittente e destinatario).
- Codifica a chiave pubblica di tipo *plaintext-aware.*

Esistono tre varianti del protocollo iKP, identificate dal valore dell'indice *i* presente nel nome.

• In 1KP solo l'acquirente può firmare i messaggi (cioè solo l'acquirente possiede una coppia di chiavi pubblica e privata).
• In 2KP anche il venditore può firmare (esistono due proprietari di coppie di chiavi).
• In 3KP anche il compratore può firmare (esistono tre proprietari di coppie di chiavi).

Tutti i protocolli iKP possono essere implementati sia via software che via hardware. In
1KP e 2KP il cliente non necessita di un dispositivo di pagamento personalizzato: per completare un pagamento bastano il numero di carta di credito e il PIN (se presente).
Comunque, per garantire maggiore sicurezza, è raccomandabile l'utilizzo di dispositivi anti-frode che proteggano il PIN e, nel caso 3KP, la chiave segreta del cliente.
È importante sottolineare ancora che lo scopo dei protocolli iKP è quello di abilitare ai pagamenti. iKP non si preoccupa di gestire come l'ordine venga inoltrato; iKP assume che l'ordine, incluso il prezzo, sia già stato concordato fra compratore e venditore. Inoltre, iKP non consente alcuna codifica dei dati relativi all'ordine. Si suppone che tale tipo di protezione sia fornita da altri protocolli esistenti, come SHTTP e SSL . Il prototipo iKP supporta solo i protocolli 2KP e 3KP; questo perché 1KP, pur essendo un protocollo molto semplice, non permette la risoluzione di contestazioni tra compratore e venditore.

4. Protocollo SET

VISA e Mastercard hanno sviluppato congiuntamente il protocollo SET (*Secure Electronic Transaction*) come metodo per il pagamento sicuro su reti aperte.

Gli obiettivi che SET si prefigge riguardo la sicurezza sono:

- Garantire la riservatezza dell'informazione.
- Assicurare l'integrità dei pagamenti.
- Autenticare compratore, venditore e acquirente.
- Definire algoritmi e protocolli necessari per tali servizi.

Gli obiettivi che SET si prefigge riguardo l'interoperabilità sono:

- Definire informazioni dettagliate per assicurare che applicazioni sviluppate da un venditore lavorino con applicazioni sviluppate da altri venditori.
- Creare e supportare uno standard aperto per pagamento con carte di credito.
- Sfruttare gli standard esistenti, quando possibile.
- Consentire l'implementazione su ogni combinazione di piattaforme hardware e software come Power PC, Intel, Sparc, UNIX, MS-DOS, OS/2, Windows, Macintosh.

Gli obiettivi che SET si prefigge riguardo l'accettazione di mercato sono:

- Ottenere un'accettazione globale, tramite una facile implementazione e un impatto minimo su venditore e compratore.
- Sfruttare le applicazioni per clienti già esistenti.
- Minimizzare lo scambio di relazioni tra acquirente e venditore, e tra compratore e fornitore.

- Fornire un protocollo efficiente dal punto di vista delle istituzioni finanziarie.

Il principio fondamentale che ha guidato gli ideatori del SET è stato quello di rendere sicure le transazioni con carta di credito su Internet, senza la necessità di modificare i circuiti bancari esistenti per le autorizzazioni. Le reti bancarie hanno dei server per l'autorizzazione che filtrano le transazioni illecite in accordo a ben specificati criteri, come ad esempio raggiungere un determinato limite di spesa o effettuare un numero eccessivo di transazioni in un dato intervallo di tempo. Così, prima che sia autorizzata la transazione, il commerciante deve chiedere autorizzazione al server. Quindi nella fase di pagamento, il commerciante riceve il pagamento corrispondente al prezzo dei beni/servizi.

Il SET introduce due nuove entità:

1. la *certification authority*, che certifica i partecipanti;

2. il *payment gateway*, che fa da filtro tra Internet e la rete bancaria.

Nel SET, ci sono sei partecipanti:

1. *il cardholder (*il *possessore della carta di credito)*, la cui carta è conforme alle specifiche SET è stata emessa da una istituzione preposta, tipicamente banche affiliate con Visa e MasterCard;

2. il *server del Commerciante;*

3. il *payment gateway*; (*Gateway di pagamento);*

4. l'issuing institution (l'*istituzione* che emette la carta di credito);

5. la *Certification Authority (CA);*

6. l'*Acquiring Institution*, che è la banca del commerciante.

Il possessore della carta, il commerciante, la certification authority ed il payment gateway sono connessi attraverso la rete. Ognuno dei partecipanti deve prima ottenere un certificato dalla CA conforme alle specifiche SET. Questi certificati sono inclusi in ognuno dei messaggi scambiati tra il cardholder , il commerciante ed payment gateway. L'issuing institution e l'acquirent institution sono collegate mediante una rete bancaria sicura e chiusa. Il gateway è il ponte fra queste reti, e protegge l'accesso alla rete bancaria. Inoltre ha due interfacce, una sul lato Internet conforme alle specifiche SET ed una dal lato della rete bancaria conforme al protocollo proprietario. Il SET garantisce la sicurezza degli scambi sia tra cliente e commerciante che tra il commerciante ed il gataway.
Tuttavia, la sicurezza offerta usa mezzi complessi e questo si traduce in un eccessivo sovraccarico computazionale, per cui il tempo di risposta può essere inadeguato. Inoltre la pesantezza delle operazioni fa sì che il SET non venga usato nei pagamenti piccoli.
Altri fattori che fanno decrescere la popolarità del SET sono:
• Gli aspetti legali, in particolare le leggi che limitano l'uso della crittografia;
• I dati riguardanti il cardholder sono memorizzati nell'hard disk di un computer e questo non da garanzie sufficienti.

5. Digital cash

Digital Cash, noto come "electronic cash" o "e-cash", è una modalità, basata su un protocollo di crittografia a chiave asimmetrica, per creare e trasmettere moneta elettronica (l'equivalente elettronico di denaro contante e assegni sulla rete delle reti: Internet).Ciò è reso possibile con l'introduzione di protocolli che consentono di effettuare transazioni.

Tra le proprietà desiderabili per un sistema Digital Cash abbiamo:

1. SICUREZZA: si cerca di trovare un modo per evitare la falsificazione della moneta elettronica, rappresentata da stringhe di bit. La copia ed il riutilizzo di moneta elettronica è computazionalmente impossibile;

2. ANONIMATO: è importante per chi utilizza moneta elettronica proteggere la propria privacy, in particolare evitare tracce sul percorso della moneta che facciano risalire a chi l'ha usata. I protocolli che vengono utilizzati garantiscono diversi livelli di anonimato, in genere tutti ragionevolmente buoni;

3. ACCETTABILITA': è importante che la moneta elettronica emessa da una banca sia accettata dalle altre banche; quando tra le banche si effettua un scambio di valuta, la loro riconciliazione avviene in modo automatico;

4. TRASFERIBILITA': è desiderabile che la moneta elettronica sia accettata da terzi, senza prima contattare la banca. In questo modo, viene valorizzata la proprietà di anonimato, anche se ciò complica il meccanismo che garantisce la sicurezza;

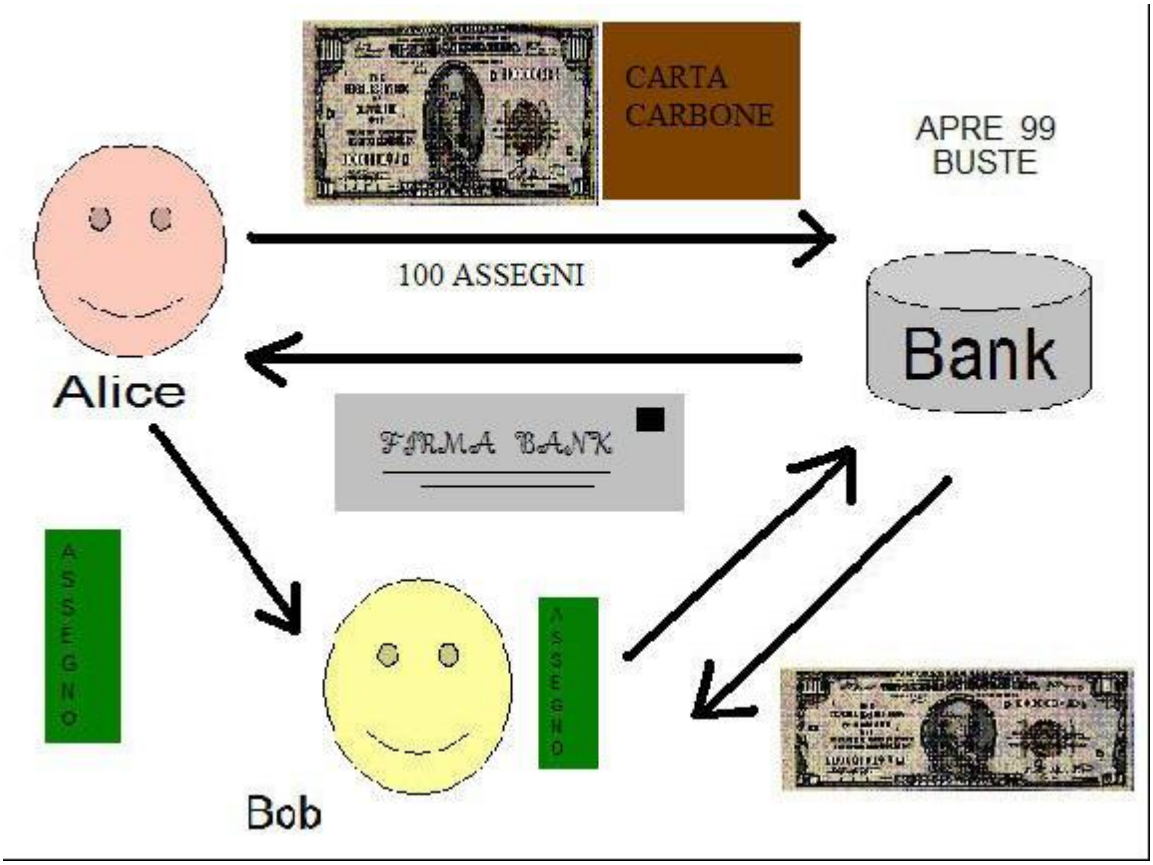
5. INDIPENDENZA DELL'HARDWARE: per il riutilizzo delle monete durante le operazioni off-line, alcuni protocolli sfruttano hardware particolari che garantiscono la protezione da possibili intrusioni;

6. TIPOLOGIE DI PAGAMENTO: le modalità di pagamento sono diverse, come l'utilizzo di moneta elettronica opportunamente "coniata", assegni elettronici, smart card e carte di credito;

7. COSTI DI GESTIONE: i costi di gestione variano fondamentalmente in funzione del grado di sicurezza che il sistema offre. Le differenti problematiche da affrontare per gestire l'e-commerce hanno portato alla creazione di diversi protocolli, ciascuno dei quali, in analisi dettagliata, si

mostra più o meno efficace nel soddisfare determinate funzionalità. Analizziamo quattro protocolli generici per un sistema digital cash. L'obiettivo è mostrare come avviene una transazione economica tra due parti generiche che chiameremo A e B. In tutti e quattro i protocolli esiste una banca che funge da mediatrice, il cui obiettivo consiste nel generare e convalidare la moneta che A invia a B. La denominazione progressiva dei protocolli è dovuta al fatto che ogni singolo protocollo risolve i problemi esistenti nel protocollo precedente. In questo modo si arriva all'ultimo protocollo, il quarto, che rappresenta la soluzione ottimale. Inoltre all'interno della sezione analizzeremo anche un attacco "ideale" a questi protocolli noto come “crimine perfetto”.Nei confronti di questo attacco i protocolli non hanno difesa e sono quindi vulnerabili.

Protocollo digital cash 1



A prepara 100 assegni con l’importo di $1000 l'uno;

A mette questi 100 assegni, insieme a della carta carbone, in 100 buste sigillate dandole alla banca;

la banca apre 99 di queste buste, scelte a caso fra le 100, spedite da A e verifica che ci siano assegni da $1000;

la banca firma l'unica busta rimasta sigillata e grazie alla carta carbone tale firma si trasferisce sull'assegno.

La banca restituisce ad A tale busta detraendo $1000 dal suo conto;

A apre la busta e spende l'assegno da un commerciante B;

B controlla che la firma della banca apposta sull'assegno sia autentica;

B incassa l'assegno dalla banca;

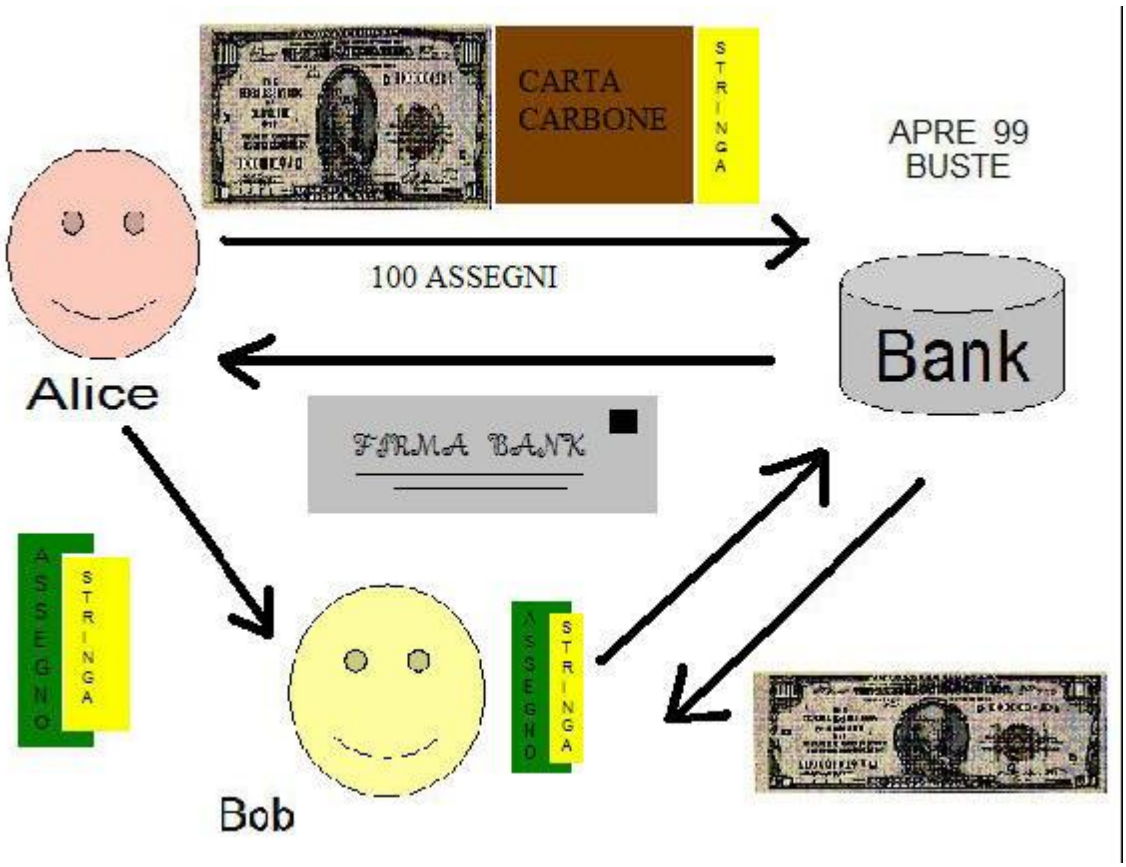
la banca verifica la sua firma e accredita $1000 sul conto di B.

Con questo protocollo viene mantenuto l'anonimato di A in quanto la banca firma l'assegno senza vederlo e si tutela sull' effettivo importo dell'assegno grazie alle 99 verifiche che precedono la firma. Risulta chiaro che A ha solo l'1% di probabilità di frodare la banca, ma resta comunque il fatto che A può copiare la moneta e spenderla più volte.

Protocollo digital cash 2

Il protocollo precedente impedisce ad A di farsi firmare dalla banca un assegno con un importo superiore ma nn le inpedisce di copiarlo e spenderlo due volte. Il seguente protocollo risolve il problema.

I suoi passi sono:



A prepara 100 assegni con l'importo di $1000 ciascuno. Ogni assegno include una stringa, significativamente lunga, scelta casualmente che lo distingue univocamente;

A mette questi 100 assegni, insieme a della carta carbone, in 100 buste sigillate dandole alla banca;

la banca apre 99 di queste buste, scelte a caso fra le 100, spedite da A e verifica che ci siano assegni da $1000 e che ogni assegno abbia una stringa univoca;

la banca firma l’unica busta rimasta sigillata e grazie alla carta carbone tale firma si trasferisce sull’assegno.

la banca restituisce ad A tale busta detraendo $1000 dal suo conto e mettendoli in un conto temporaneo;

A apre la busta e spende l’assegno da un commerciante B;

B controlla che la firma della banca apposta sull’assegno sia autentica;

B incassa l’assegno dalla banca;

la banca verifica la propria firma e controlla nel suo data-base che un assegno con quella stringa non sia già stato depositato, in tal caso accredita l’importo sul conto di B prendendoli dal conto temporaneo e memorizza la stringa di unicità in un data-base.

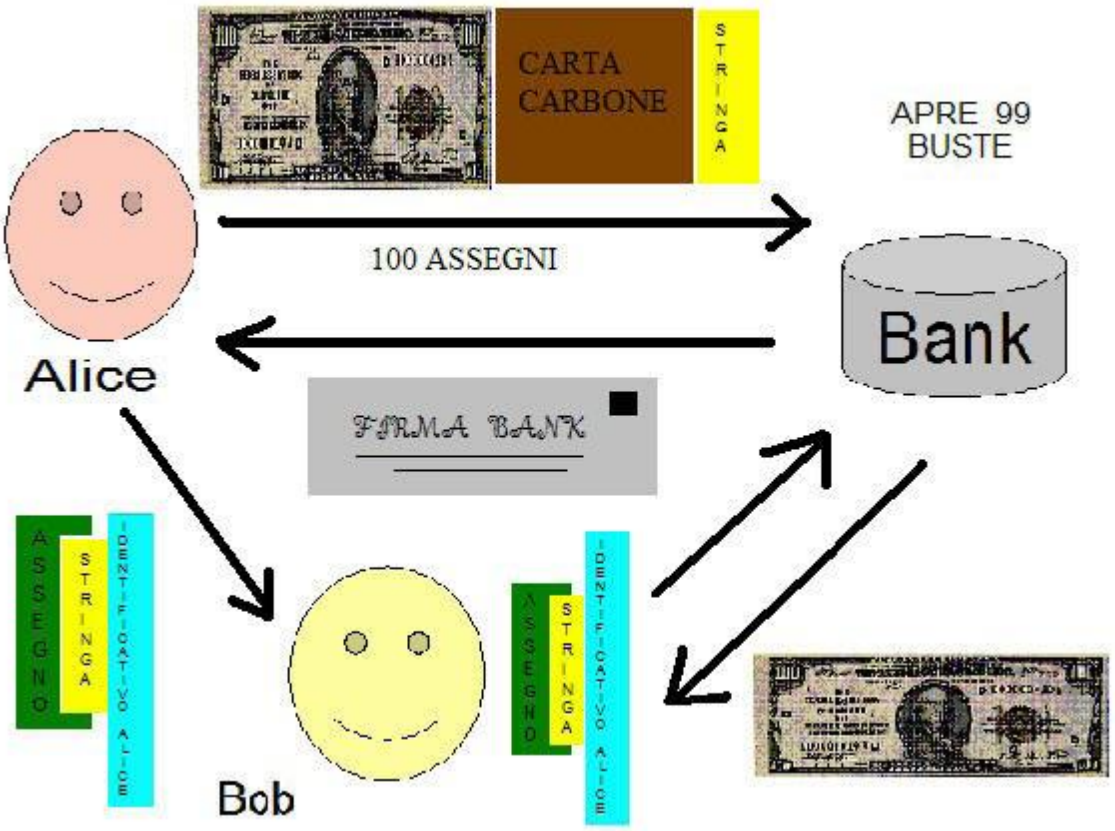
Se l'assegno è stato già pagato la banca nn lo accetta.

Con questo protocollo la banca si tutela nel caso A cercasse di spendere più volte l’assegno e nel caso in cui B cercasse di depositarlo più volte.

Protocollo digital cash 3

Il protocollo 2 tutela la banca dai possibili imbrogli ma non riesce comunque ad identificare un eventuale falsificatore (A oppure B). Il protocollo 3 risolve questo problema.

I suoi passi sono:



A prepara 100 assegni con l'importo di $1000 ciascuno. Ogni assegno include una stringa significativamente lunga, scelta casualmente, che lo distingue univocamente;

A mette questi 100 assegni, insieme a della carta carbone, in 100 buste sigillate dandole alla banca;

la banca apre 99 di queste buste, scelte a caso fra le 100, spedite da A e verifica che ci siano assegni da $1000 e che

ogni assegno abbia una stringa univoca;

la banca firma l’unica busta rimasta sigillata e grazie alla carta carbone tale firma si trasferisce sull’assegno.

la banca restituisce ad A tale busta detraendo $1000 dal suo conto;

A apre la busta e spende l’assegno da un commerciante B;

B controlla che la firma della banca apposta sull’assegno sia autentica;

B chiede ad A di scrivere una stringa casuale di identificazione sull’assegno;

A accetta;

B incassa l’assegno dalla banca;

la banca verifica la propria firma e controlla nel proprio database di nn aver già pagato un assegno con la stessa stringa identificativa; se tutto è corretto, accredita $ 1.000 sul conto del negoziante, quindi memorizza la stringa identificativa e quella di riconoscimento nel data-base;

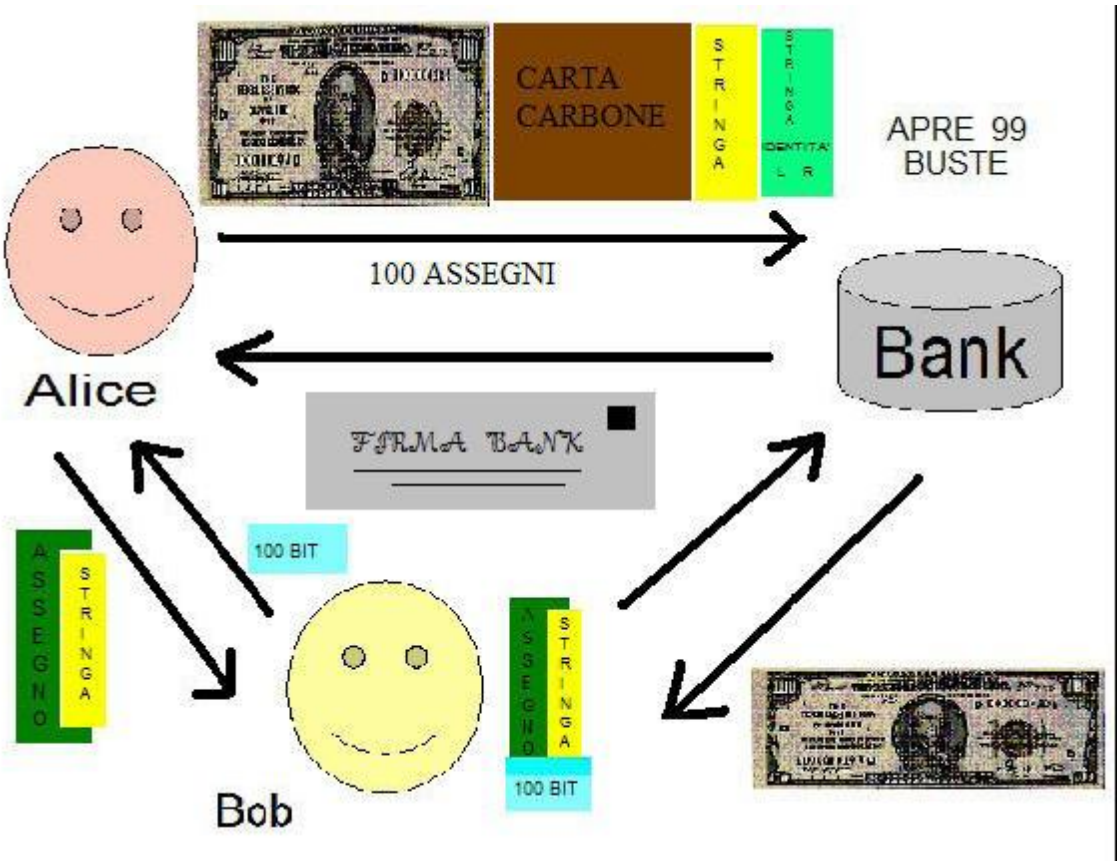
se l'assegno è stato già pagato, la banca non lo accetta. In tal caso, confronta la stringa di identificazione con quella memorizzata nel data-base. Se è la stessa, la banca sa che a copiarlo è stato B; se è diversa, sa che la copia è stata fatta da A che ha preso l'assegno.

Questo protocollo presuppone innanzitutto che B non possa variare la stringa scritta da A, a meno di un danneggiamento dell’assegno (per esempio l’assegno contiene dei quadrati da riempire con delle X) e poi che A sia presente al momento in cui B verifica l’assegno con la banca, altrimenti A può anche incastrare B spendendo lo stesso assegno una seconda vola e fornendo la stessa stringa d'identificazione. Se B non conserva un data-base degli assegni già ricevuti, viene imbrogliato ma non si riesce a risalire all'identità di A.

Protocollo digital cash 4

Con questo protocollo la banca può scoprire l'identità di A nel momento in cui questi tenta di imbrogliare, riesce quindi a fare ciò che il protocollo precedente non poteva. Per fare ciò è necessario utilizzare le nozioni di crittografia.

I passi sono:



A prepara 100 assegni anonimi di $1000 ciascuno e ognuno di essi contiene:

- Ammontare : $1000;

- Stringa di riconoscimento assegno: X, unica per ciascun assegno;

- 100 coppie di stringhe di bit d'identità: I1 (I1L , I1R),I2(I2L, I2R)..I100 (I100L , I100R). ognuna delle quali è generata come

segue:

A crea una stringa che contiene il suo nome,l'indirizzo, e altre informazioni richieste dalla banca;

A scompone il tutto in due pezzi con un protocollo segreto e invia tutto alla banca attraverso il protocollo *bit-commitment*.

Per esempio, I37 è composto da due stringhe I37L e I37R che, a richiesta della banca, possono essere aperte da A in ogni momento; quindi ogni assegno contiene l'importo, la stringa di riconoscimento e le 100 stringhe d'identità.

A nasconde i 100 assegni con il protocollo di firma alla cieca visto in precedenza e spedisce tutto alla banca;

la banca apre 99 di queste buste, scelte a caso fra le 100, spedite da A e verifica che siano assegni da $1000, verifica la stringa di riconoscimento e chiede ad A di rivelare le stringhe di identità;

nel caso tali controlli siano andati a buon fine la banca firma l'assegno non scoperto ad A e detrae $1000 dal suo conto;

A apre la busta e spende l'assegno da un commerciante B;

B controlla che la firma della banca apposta sull'assegno sia autentica;

B consegna ad A una stringa casuale a 100 bit, e chiede di aprire la parte destra o sinistra di ciascuna delle stringhe di identità poste sull'assegno, a seconda che l'i-esimo bit vale 0 o 1 alice rivelerà la parte sinistra di Ik o la parte destra.

B incassa l'assegno dalla banca;

la banca verifica nel suo data-base che un assegno con quella stringa non sia già stato depositato, in tal caso accredita l' importo sul conto di B e memorizza la stringa di riconoscimento e le altre informazioni

identificative nel data-base; se la stringa di riconoscimento dell'assegno è già presente nel data-base la banca lo rifiuta. Poi confronta la stringa identificativa sull'assegno con quella memorizzata nel data-base:

se sono uguali allora capisce che B è colpevole di aver copiato l'assegno; altrimenti la banca capisce che A ha copiato l'assegno e siccome A ha riutilizzato la moneta con un altro commerciante C, quest'ultimo avrà dato ad A una stringa casuale di 100 bit differenti da quella datagli da B, la banca trova una stringa di identificazione di cui un negoziante ha avuto la parte destra e l'altro la parte sinistra. Completando il protocollo segreto la banca rivela l'identità di A.

La banca ricerca proprio i bit in cui differiscono le due stringhe per rivelare l'identità di A. In particolare la probabilità che le due stringhe di 100 bit siano uguali è 1/2100 per cui la banca ricerca una coppia che, combinando i risultati delle due differenti aperture, abbia entrambe le metà svelate. Per esempio supponiamo che le due stringhe differiscano nell' *n-esimo* bit allora le parti da mettere insieme (in xor) sono InL InR rivelando così l'identità di A.

E' facile intuire che in questo protocollo viene mantenuto l'anonimato di A (a meno che non imbrogli) e soprattutto all'atto dell'acquisto (in particolare della verifica dell'assegno da parte di B) non è più necessaria la presenza di A per i motivi visti prima.

Crimine perfetto

Come abbiamo visto sino ad ora l'anonimato risulta essere fondamentale per le transazioni di moneta elettronica. Ma lo stesso anonimato può essere usato per scopi illeciti come mostrano i seguenti passi del protocollo noto come crimine perfetto:

A rapisce un bambino;

A prepara 10000 assegni anonimi di qualsiasi cifra;

A nasconde questi assegni con il protocollo di firma digitale e manda tutto alle autorità con le seguenti richieste che se non esaudite comporterebbero l'uccisione del bambino:

-una banca deve firmare tutti assegni;

-pubblicare i risultati su di un giornale;

-le autorità accettano;

A verifica l'avvenuta pubblicazione, scopre gli assegni e li spende;

A rilascia il bambino.

Lo schema del crimine perfetto rappresenta l'unico modo per poter attaccare con successo il protocollo 4.

Firma Digitale

I nuovi sistemi di pagamento elettronico si basano su un estensione delle firme digitali dette blind signature. Uno schema di firma digitale è un protocollo per ottenere una firma in modo che il firmatario non possa vedere ciò che firma. In un sistema di questo tipo viene garantito l'anonimato del mittente, ed è per questo motivo che tali schemi vengono utilizzati nei protocolli Digital Cash, ove assicurano l' anonimato di chi usa moneta digitale. Sfortunatamente questo anonimato può essere usato da un malintenzionato per ottenere una firma che altrimenti non potrebbe avere. Descriveremo tali protocolli con l'aiuto di due partecipanti immaginari A e B. Come accade usando del contante, A dovrà poter trasferire moneta digitale a B senza che una terza persona riesca a conoscere l'identità di A. Come conseguenza di ciò B potrà depositare la moneta elettronica sul suo conto bancario senza che la banca sappia chi sia A, quindi da dove o meglio da chi proviene il denaro. Ma se A

tentasse di dare a due persone lo stesso denaro elettronico con un programma che copia i bit, verrebbe individuata dalla banca, se B cercasse di depositare la stessa moneta digitale su due differenti conti, sarebbe individuato, ma A resterebbe anonima. La nozione di firma digitale fu inventata da D.Chaum che propose anche una prima implementazione. Lo schema usa l'algoritmo RSA.

6. Ecash

Ecash è un sistema di pagamenti sicuri per Internet elaborato da Digicash. Non è un sistema basato su transazioni con carta di credito: Ecash usa un vero e proprio denaro virtuale, rappresentato da monete fornite dalle banche associate. Tali banche sono responsabili della certificazione dell'autenticità delle monete virtuali di Ecash. Per poter utilizzare Ecash è necessario aprire un conto corrente con una delle banche partecipanti. Sarà poi possibile memorizzare sul proprio computer monete elettroniche; tali monete, al momento di un acquisto, verranno trasferite al venditore sfruttando tecniche di crittografia a chiave pubblica e di firma digitale.

Caratteristiche

Operare con Ecash è piuttosto semplice: è sufficiente procurarsi il software Ecash client e aprire un conto con una delle banche partecipanti. Il client Ecash, reperibile gratuitamente in rete, è in grado di operare pagamenti con qualsiasi altro utente Internet che stia anch'egli utilizzando tale client. Ad utenti e negozi non è richiesto alcun hardware particolare, le banche invece avranno uno speciale hardware di codifica che assicuri velocità e affidabilità delle operazioni. Ecash è un sistema basato su moneta, il che significa che si crea denaro digitale usando una

firma elettronica: tale firma rappresenta una quantità fissa di denaro chiamata moneta. Sfruttando il client, il compratore ritira ecash (una particolare forma di denaro digitale) da una banca e lo memorizza sul proprio computer. Il compratore è adesso in grado di spendere tale denaro presso qualsiasi negozio che accetti ecash, senza dover aprire un conto con tale negozio e senza dover trasmettere un numero di carta di credito. Il negozio è rappresentato da un documento html contenente una serie di indirizzi (URL) indicanti le merci in vendita. Ecash consente anche di effettuare pagamenti da persona a persona.

Ecash lavora con tutte le maggiori piattaforme (MS Windows, Macintosh, UNIX). Esiste sia una versione con interfaccia grafica che una versione solo testuale. Per la versione corrente di Ecash è necessaria una connessione, ma una versione che sfrutterà l'e-mail è annunciata per un prossimo futuro.

Sicurezza e riservatezza

Per garantire sicurezza e riservatezza dei dati e delle transazioni Ecash sfrutta tecniche di firma digitale a chiave pubblica . I prelievi di ecash dal conto di ogni utente sono inoltre protetti da una password nota esclusivamente all'utente stesso. Quando è utilizzato per la prima volta, il software Ecash genera automaticamente una coppia di chiavi per codifica RSA. Ogni persona che utilizza Ecash possiede un'unica coppia di chiavi. Con queste è possibile garantire la sicurezza di ogni transazione e messaggio. Ecash garantisce l'anonimato solo di chi paga. Durante un pagamento, colui che lo effettua può rendere nota la propria identità, ma solo se decide di farlo. Chi riceve il pagamento, invece, non gode di anonimato: durante la fase di compensazione il beneficiario di una transazione è identificato dalla banca. Quando il compratore ha necessità di effettuare un pagamento, deve avere ecash sul proprio computer. Il

prelievo dalla banca è in realtà, per ragioni di riservatezza, qualcosa di più complesso che un semplice trasferimento di ecash dalla banca al PC del compratore.Il PC dell'utente calcola quante monete sono necessarie per ottenere la somma richiesta. Successivamente è il computer dell'utente stesso che crea monete assegnando ad ognuna di loro un numero di serie casuale. Quindi spedisce alla banca queste monete, una ad una inserite in una speciale busta: la quale rappresenta il fattore "cecità".

La banca codifica i numeri "ciechi" con la propria chiave segreta (firma digitale), grazie alla proprietà della firma cieca che consente di applicare tale firma attraverso la busta; allo stesso tempo, la banca addebita sul conto dell'utente la stessa somma. Le monete autenticate sono restituite all'utente, che potrà togliere loro il fattore di "cecità" introdotto in precedenza, senza alterare la firma della banca. I numeri di serie con le loro firme rappresentano adesso moneta digitale; il valore delle monete è garantito dalla banca. Quando l'utente spenderà tali monete, la banca le accetterà in quanto da lei firmate. Tuttavia, poiché non sarà in grado di riconoscere le monete (che erano nascoste nella busta al momento di essere firmate), la banca non potrà dire chi ha effettuato il pagamento.

7. Micromint

Ideato da Rivest e Shamir. In MicroMint le monete, prodotte da un broker, vengono distribuite agli utenti che girano queste monete ai venditori come pagamento. I venditori restituiscono le monete al broker che ne rimborsa l'ammontare attraverso altri mezzi.

Una moneta è una stringa di bit la cui validità può essere facilmente constatata da ognuno, ma che è difficile da produrre.

In MicroMint generare più monete risulta più conveniente di generarne poche infatti è necessario un cospicuo investimento iniziale per coniare la prima moneta, ma le successive vengono prodotte con molta facilità e in minor tempo. Simile all'economia per una coniatura reale, per la quale si investe molto per acquisire macchinari costosi che consentono poi di produrre monete in modo economico.

Monete come collisioni di funzioni hash

Le monete di MicroMint sono rappresentate da collisioni di funzioni hash, tramite specifiche funzioni hash one-way h che trasformano stringhe x di m bit in stringhe y di n bit. Diciamo che x è un'anteprima di y se h(x)=y. Una coppia di stringhe distinte di m bit (x1,x2) è detta collisione a due vie se h(x1)=h(x2)=y per qualche y di n bit. Un modo per produrre una collisione a 2-vie in modo accettabile potrebbe essere quello di effettuare l'hash di $2^{n/2}$ valori di x.

Se effettuiamo l'hash per c volte su tanti valori di x, quanti sono necessari per produrre la prima collisione, si generano approssimativamente in c^2, con $1 <= c <= 2^{n/2}$, altrettante collisioni. Ciò vuol dire che una volta trovata la prima collisione produrre le altre non richiede eccessivi tempi di calcolo.

Monete come collisioni a k-vie

Per rendere facile il lavoro del broker nel generare collisioni a due vie si sceglie generalmente un valore n abbastanza piccolo. In questo modo però si rende altrettanto facile il lavoro per un intruso che vuole falsificare le monete. La sicurezza si raggiunge scegliendo le collisioni a k-vie. Una collisione a k-vie è un'insieme di k valori distinti x1,..,xk che hanno lo stesso valore hash y. In questo modo per trovare una collisione a k-vie,approssimativamente, si dovrebbero esaminare $2^{n(k-1)/k}$ valori di x.

Esaminando c volte questi valori, con $1 <= c <= 2^{n/k}$, ci si aspetta di vedere circa ck collisioni a k-vie. La scelta di k influenza il grado di sicurezza contro eventuali falsificazioni. Inoltre la validità di k monete può essere facilmente verificata dalla relazione h(x1)=h(x2)=…=h(xk)=y.

Coniatura delle monete

Il processo di calcolo h(x)=y equivale a lanciare una biglia in uno tra 2^n recipienti. Supponiamo che una moneta sia proprio una serie di k biglie che lanciate sono entrate nello stesso contenitore, per ottenere ciò se ne dovrebbero lanciare una quantità abbastanza grande.

Per coniare una moneta il broker crea 2^n recipienti e lancia circa $k*2^n$ biglie, il contenitore che contiene almeno k biglie diventerà una moneta. Quindi k biglie scelte a caso formeranno una moneta mentre le altre in genere non vengono utilizzate, questo costituisce un problema in quanto possono risultare utili nel momento in cui qualcuno intende effettuare delle falsificazioni. Inoltre in questo modo si semplifica il lavoro del broker che può tener traccia di ogni monete coniata utilizzando un semplice bit. 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 quelle che possono essere memorizzate su un hard - disk ed il numero di quelle di cui il broker ha realmente bisogno. Per trovare un giusto equilibrio possiamo pensare di rendere molte biglie inutili allo scopo di coniare monete. Ciò può essere fatto supponendo che una biglia sia "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. Allora se i 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 recipiente nel quale la biglia x è stata lanciata. Utilizzando questo processo il broker lancia $k*2^n$

e ne memorizza circa $k*2^u$ generando circa $1/2*2^u$ monete valide. Nel caso in cui il numero di bit in output della funzione hash scelta (ex.: DES, MD5) superi il valore si possono scegliere soltanto n bit, ad esempio gli n bit di peso minore.

Distribuzione delle monete

Il broker inizia la distribuzione delle monete ai suoi clienti generalmente verso la fine di ogni mese. Queste monete verranno poi utilizzate nel mese successivo e soltanto dopo che il broker rende pubblico il criterio per ritenerle valide. I clienti che comprano monete caricano questo acquisto sulla propria carta di credito. Il broker da parte sua tiene traccia delle monete distribuite ai singoli utenti. Quelle monete non utilizzate gli verranno restituite a favore di altre valide per il mese successivo.

Scenario in dettaglio

Vedremo un esempio specifico di come un broker fissa la scelta dei parametri per coniare monete valide per un dato mese. I calcoli sono approssimati alla più vicina potenza di 2 ma restano ugualmente molto significativi. Il broker investirà su un hardware molto solido, per avere dei vantaggi su eventuali contraffattori, che farà lavorare ininterrottamente in un mese per produrre monete da distribuire il mese successivo. È inoltre preferibile che l'hardware contenga dei chip specifici per calcolare i valori della funzione hash. Supponiamo che il broker abbia un profitto,mensile, di un milione di dollari (2^{27} cent), caricando la tariffa di mediazione pari al 10% per ogni moneta inviata pagherà al venditore 0.9 cent quando la stessa verrà riscattata, così facendo distribuirà un numero di monete (circa 2^{30}) tali da raccogliere il suo milione di dollari. Scegliamo come valore di k per la collisione 4 mentre il valore di u=31. In questo modo deve creare un array di 2^{31} recipienti, ognuno dei quali può contenere dai 4 valori di x in su. Con queste assunzioni il broker deve lanciare una

media di 4 biglie per ciascuno dei 2^{31} recipienti generando quindi $4*2^{31}= 2^{33}$ valori di x che producono i valori di y buoni. Usando questi parametri la probabilità che un recipiente contenga 4 biglie è pari ad ½ equivalente alla probabilità di ogni recipiente di generare una moneta. Ciò vuol dire che il numero di monete generate è $½ *2^{31}= 2^{30}$ che è il numero desiderato. La memorizzazione di una coppia (x, h(x)) richiede meno di 16 byte, in totale lo spazio richiesto è di circa 2^{37} cioè 128 Gbyte.

Pagamenti

Ogni volta che il cliente deve pagare un acquisto ad un venditore gli spedisce la serie x=x1, x2,…,xk che forma la moneta. Quest'ultima controlla che sia una moneta eseguendo l'hash su ogni valore della serie e verificando che è uguale per tutti(collisione a k-vie).

Riscatto delle monete

Il venditore, ogni giorno, restituisce al broker le monete che ha accumulato. Il broker controlla le varie monete tentando di individuare eventuali monete che sono state già riscattate. Per le monete valide paga al venditore la somma stabilita. Per quelle che gli sono state inviate più volte sceglie di pagare uno solo dei venditori, penalizzando inevitabilmente gli altri.

Sicurezza

Gli attacchi possibili ad uno schema Micromint sono a larga e piccola scala. Con questi termini si intendono rispettivamente attacchi che portano a consistenti guadagni o a piccoli guadagni per eventuali contraffattori. Gli attacchi a piccola scala non portano a veri profitti pertanto i meccanismi di sicurezza, per lo schema, sono costruiti per opporsi agli attacchi a larga scala. Di seguito verranno descritti tre tipi di attacchi che si possono effettuare.

I. Contraffazione

Una contraffazione a piccola scala è inapplicabile se si pensa che una normale workstation può effettuare solo 2^{14} operazioni hash al secondo mentre per coniare una moneta falsa ci vogliono 2^{45} operazioni hash. E' necessario quindi contrastare la contraffazione a larga scala. Può essere fatto nel seguente modo:

- tutte le monete false vengono invalidate automaticamente alla fine del mese;

- le monete false non possono essere generate sino a quando il broker non annuncia il nuovo criterio di validità delle monete per il prossimo mese;

- utilizzare dei predicati nascosti che forniscono un intervallo di tempo più adeguato per respingere monete false senza andare ad intaccare la validità delle monete legali che sono in circolazione;

- il broker può intercettare un falsario verificando tra le monete ricevute quelle che lui non ha mai generato. Ciò è possibile perché, come precedentemente detto, solo la metà dei recipienti utilizzati producevano monete;

- in qualsiasi momento del mese il broker può dichiarare la validità delle monete ritirando tutte le monete ed immettendone delle nuove comunicando il nuovo criterio di validità;

- il broker può simultaneamente generare monete per più mesi utilizzando una grande computazione; ciò rende il compito del falsario più difficile a meno che questi non abbia a disposizione gli stessi mezzi del broker.

II. *Furto di monete*

Il furto di monete è relativo alla fase di distribuzione delle monete agli utenti e alla fase di riscatto di monete dal venditore. Ciò è risolvibile eseguendo in queste fasi delle cifrature e quindi utilizzando una chiave di crittografia nelle relazioni broker-utente e broker-venditore. Per quanto riguarda l'altra relazione, utente-venditore, la protezione durante l'invio di monete può essere fatta utilizzando una chiave pubblica di crittografia oppure fornendo ad ogni utente delle monete personalizzate in modo tale che questi sia l'unico a poterle utilizzare.

Un'altra situazione da considerare è che due venditori possono riscattare le stesse monete. Ciò è evitabile creando monete specifiche anche per i venditori.

III. Riutilizzo di monete

Lo schema Micromint non garantisce l'anonimato e quindi il broker può individuare,per moneta riutilizzata, i venditori che gli hanno fornito le diverse copie. Conoscendo a chi egli ha fornito la monete, può individuare attraverso l'aiuto di venditori onesti, il cliente che ha speso diverse copie della stessa moneta. Provare però legalmente che un cliente sia reo di utilizzo di monete duplicate risulterà molto difficile poiché non si utilizzano schemi di firme digitali.