Contanti Elettronici

Marco Quartullo
Luca Proietto
Claudio Neri
Leonardo Puleggi

Introduzione

Marco Quartullo

Il Digital Cash

È un sistema di pagamento elettronico

I sistemi di pagamento tradizionale si dividono in:

- Sistemi basati su oggetti (es: denaro, carte prepagate)
- Sistemi basati su acconto (es: carte di credito)

Hanno caratteristiche differenti (praticità e anonimato vs sicurezza)

I sistemi digital cash vogliono ottenere i pregi ed evitare i difetti dei sistemi tradizionali.

Con il termine moneta digitale definiamo dunque un sistema di pagamento anonimo e basato su oggetti, che permette di avere sicurezza verso la perdita di denaro e contemporaneamente di proteggere la privacy degli utenti, mantenendo la facilità d'uso tipica del denaro tradizionale.

Caratteristiche del denaro digitale Sicurezza

- Contraffazione
  - Riproduzione della moneta

    Viene creato denaro valido senza un corrispondente prelievo bancario
  - Spesa multipla

    Lo stesso denaro viene utilizzato più volte
- Autenticità
  - Identificazione degli utenti di una transazione
  - Integrità dei messaggi
  - Non ripudiabilità

Caratteristiche del denaro digitale Privacy e Portabilità

- Privacy

  Come per il denaro tradizionale, anche quello digitale deve essere anonimo: non deve essere possibile risalire da esso agli utenti che lo hanno utilizzato

  L'anonimato tuttavia incrementa il rischio di attività illegali come il riciclaggio, la contraffazione e il ricatto. Maggiore privacy significa dunque minor sicurezza, e viceversa

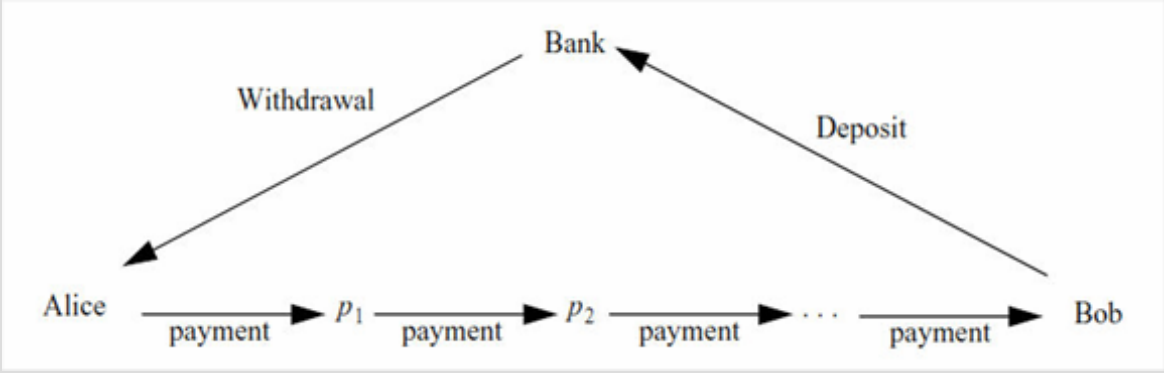
  I diversi sistemi di digital cash devono scegliere il livello di compromesso tra le due opposte esigenze.

- Portabilità

  Le caratteristiche di utilizzo e sicurezza del denaro digitale non devono dipendere da locazioni fisiche

Caratteristiche del denaro digitale Trasferibilità

La proprietà del denaro di essere trasferibile consente agli utenti di effettuare pagamenti senza dover contattare la banca:



Possibili problemi:

- Aumento di dimensioni delle monete
- Ritardi nell'identificazione di spese multiple
- Difficoltà nell'identificazione di generiche attività illecite

Caratteristiche del denaro digitale Divisibilità, Pagamenti fuori linea

- Divisibilità

  Proprietà di poter effettuare dei cambi per le monete; consente una maggior flessibilità e praticità nelle transazioni

- Pagamenti fuori linea

  I pagamenti possono essere implementati sia in linea che fuori linea:

  - In linea: ad ogni transazione la banca fa da intermediario; maggior sicurezza, maggiori costi, no anonimato, no trasferibilità
  - Fuori linea: il deposito di denaro (e il relativo controllo di autenticità) può avvenire dopo vario tempo dalla transazione; maggior praticità, minori costi, proprietà di trasferibilità e anonimato garantite, ma maggiore vulnerabilità.

Basi Matematiche e Algoritmiche

Marco Quartullo

Fondamenta Matematiche

Riprendiamo solamente alcuni dei concetti e delle definizioni utili, in quanto molti di essi sono stati già ampiamente trattati; in particolare, vedremo quelli meno visti finora.

Quadrato residuo

a in Z^*_n è un quadrato residuo modulo n se esiste una x in Z^*_n tale che risulti $x^2 \equiv a \bmod n$. L'insieme di tutti i residui quadratici modulo n è indicato con Q_n.

Radice quadrata

Sia a in Z^*_n. Se x in Z^*_n soddisfa $x^2 \equiv a \bmod n$, allora x è una radice quadrata di modulo n

Fondamenta Matematiche Residuità quadratica

Il predicato della residuità quadratica definisce una relazione di equivalenza in $\boldsymbol{Z^*_n}$

Esiste un algoritmo efficiente, basato sul teorema di Eulero, per decidere la residuità quadratica modulo interi con fattorizzazione conosciuta.

Nel caso in cui invece la fattorizzazione non sia conosciuta, qualche informazione circa la residuità quadratica può essere data dal simbolo di Jacobi.

Fondamenta Matematiche
Simbolo di Jacobi

Simbolo di Jacobi:

Il simbolo di Jacobi (*y* / *x*) è definito come

$$\left(\frac{y}{x}\right) = \prod_{i=1}^{k} \left(\frac{y}{p_i}\right)^{h_i}$$

dove i p_i sono primi distinti e gli h_i sono interi positivi.

Se (*y* / *x*) = *-1* allora sicuramente *y* è un quadrato non residuo modulo *x.*
Non si conosce alcun algoritmo efficiente per decidere la residuità quadratica se il simbolo di Jacobi vale +1: in questo caso la via più "veloce" consiste nel fattorizzare *n.*

Fondamenta Matematiche
Interi di Blum e Williams (1)

Strettamente collegati allo studio del simbolo di Jacobi sono gli

Interi di Blum:

- Se *p* e *q* sono primi differenti ed entrambi ≡ 3 mod 4, e *k(1)* e *k(2)* sono interi positivi dispari, allora *x* definito come

  $x = p^{k(1)} q^{k(2)}$

  è un intero di Blum

Interi di Williams:

Un intero di Williams è un tipo specifico di intero di Blum: ne condivide infatti tutte le proprietà, ma definisce p e q come ≡ 3 mod 8

Fondamenta Matematiche Interi di Blum e Williams (2)

Assunzione della residuità quadratica

- È la proprietà a noi d'interesse riguardo gli interi di Blum (e di Williams).
- La classe degli interi di Blum prodotto di due primi della stessa lunghezza costituisce il più difficile input per ogni conosciuto algoritmo efficiente di fattorizzazione; più precisamente, nessun algoritmo efficiente può dare il valore del predicato della residuità quadratica con probabilità maggiore della scelta casuale.

Importanti sistemi di digital cash sfruttano questa proprietà

Rappresentazione in gruppi del primo ordine

- Estende la difficoltà computazionale del problema del logaritmo discreto al calcolo di un insieme di numeri collegati a logaritmi discreti
- Il problema consiste nel trovare, a partire da un gruppo primo *G*, una sequenza di generatori (g_1, g_2, ..., g_k) con $k \geq 2$, g_i *in G* e un elemento fissato *h in G*, una rappresentazione di *h* tale che

$$h \bmod p = \prod_{i=1}^{k} g_i^{a_i}$$

(dove a_i è un intero).

- Trovare una rappresentazione di *h* è difficile, tranne nel caso in cui prima scegliamo (g_1, g_2, ..., g_k) e (a_1, a_2, ..., a_k) e poi calcoliamo *h*.
- È un concetto di base per il sistema digital cash di Brands.

Algoritmi e protocolli

Vedremo ora alcuni algoritmi e protocolli utili nei sistemi per la gestione di contanti elettronici. Alcuni di essi verranno solo brevemente richiamati in quanto già ampiamente trattati in altri lavori

RSA

- Sistema a chiave pubblica
- Utilizzabile sia per cifrare messaggi che per la firma digitale
- Sicurezza basata sulla difficoltà di fattorizzare grandi numeri primi

Firma digitale

- Equivalente della firma autografa
- Basata sulla crittografia a chiave pubblica
- Garantisce la non ripudiabilità nelle transazioni finanziarie

cut-and-choose secret sharing

Metodo cut-and-choose:

Rabin, 1978; assicura che un utente divida equamente un oggetto

- L'utente A taglia l'oggetto d'interesse in due parti
- L'utente B sceglie una delle due parti
- L'utente A prende la rimanente

Metodo secret sharing:

Utile quando si vuole dividere un messaggio segreto tra differenti persone. L'algoritmo più semplice per implementarlo prevede che si divida il messaggio tra n persone, e che esso sia ricostruibile in presenza di m delle sue parti (ovviamente risulta $m < n$)

Bit commitment

I protocolli di bit commitment impediscono di modificare risposte nella situazione in cui un utente A invii ad un utente B un messaggio contenente una risposta cifrata da conoscere solo dopo un certo tempo, dopo il quale A invierà a B la chiave per decifrare la risposta. A potrebbe però prevedere una chiave per ogni risposta, da inviare a B in base agli accadimenti trascorsi dopo il tempo prestabilito.

Vediamo una possibile implementazione, a chiave simmetrica:

- B genera una stringa casuale *R* e la invia ad A
- A invia il messaggio *b* cifrato con la chiave *K* a B, e contenente anche *R*: $E_k(R, b)$
- Al momento opportuno, A invia *K* a *B*
- *B* decifra il messaggio, e verifica che non è stato modificato controllando che la stringa *R'* decifrata sia proprio *R*

Schema di firma basato sul logaritmo discreto (1)

- Problema del logaritmo discreto: siano p e q primi, e tali che $q \mid (p - 1)$ (cioè q è un fattore primo di $(p - 1)$). Sia g un generatore. Allora, dati p, q, g e $y = g^a \bmod p$, il problema consiste nel trovare l'unico intero a, con $0 \leq a \leq q - 1$, tale che $g^a \equiv y \bmod p$
- Non esiste alcun algoritmo in grado di risolvere tale problema in tempo polinomiale.

Vediamo uno schema di firma *challenge-response* basato su questo problema.

Esso prevede come chiave privata il numero a, e come chiave pubblica $y = g^a \bmod p$. Si presume che g, p e q siano pubblici.

Analizziamo dunque il protocollo di invio di un messaggio *m* firmato da un utente A ad un utente B

Schema di firma basato sul logaritmo discreto (2)

- A firma il messaggio *m,* calcolando m^a mod *p.*
- B vuole verificare la firma; A crea un numero casuale *w* e calcola g^w mod *p* e m^w mod *p*, e li invia a B
- B genera a sua volta un numero casuale *c*, come sfida; lo invia ad A
- A invia, come risposta alla sfida, $r = ca + w$
- B calcola allora g^r mod *p*, e verifica che sia pari a g^{ca} x g^w mod *p*; fa lo stesso anche per m^r mod *p* e m^{ca} x m^w mod *p*
- La firma è dunque corretta, e non vi sono state manomissioni, se dunque

  $(g^a)^c \times g^w \equiv g^{ca+w} \equiv g^r;$

  $(m^a)^c \times m^w \equiv m^{ca+w} \equiv m^r$

Schema di firma di Schnorr

È un protocollo zero-knowledge, basato anch'esso sul problema del logaritmo discreto, e utilizzato nel sistema di Brands; analogamente al precedente schema la chiave privata è *a*, la pubblica $y = g^a \bmod p$

- A vuole firmare *M*; sceglie un numero casuale (il cosiddetto segreto) *w*, con $1 \le w \le q - 1$; calcola $x = a^w$ mod *p*
- A concatena poi *M* e *x*, e calcola una funzione hash del risultato: $e = H(M,x)$
- A calcola ancora $v = (w + ae)$ mod *q*, e invia la firma (*v, e*) a B, insieme ad *x* ed *M*
- L'utente B verifica allora che $g^v = x * y^e$ mod *p*, e che $e = H(M,x)$: se ciò accade la firma è valida
- La prova che la firma è corretta è dunque

$$g^v \equiv g^w g^{ae} \equiv xy^e$$

Crittografia a curva ellittica (1)

Ha come idea di base quella di utilizzare gruppi di punti su una curva ellittica nei sistemi esistenti basati sul problema del logaritmo discreto.

I Crittosistemi a Curva Ellittica offrono elevata efficienza ed un basso overhead per la cifratura, la firma digitale e la gestione delle chiavi, proponendosi come la scelta ideale per sistemi di contanti elettronici basati su smart card.

I vantaggi dei Crittosistemi a Curva Ellittica includono:

- La più alta forza crittografica per bit tra tutti i sistemi a chiave pubblica.
- Enorme risparmio di calcoli (più velocità), occupazione di banda e spazio su disco rispetto agli altri sistemi a chiave pubblica
- Ulteriore risparmio di banda per le ridotte dimensioni delle firme

Crittografia a curva ellittica (2)

Una curva ellittica è definita da un'equazione della forma

$$y^2 [+xy] = x^3 + ax^2 + b$$

dove x e y sono variabili, a e b sono costanti, e le parentesi quadrate indicano l'opzionalità del termine $+xy$.

La proprietà cruciale di una curva ellittica è che possiamo definire una regola per “aggiungere” due punti appartenenti alla curva per ottenere un terzo punto, anch'esso appartenente alla curva; tale regola soddisfa le normali proprietà dell'addizione.

Per le equazioni per la regola dell'addizione si rimanda alla relazione, dove sono descritte nel dettaglio

Crittografia a curva ellittica (3)

Affinché l'addizione sia ben definita per ogni coppia di punti, è necessario includere un punto zero “extra”, che non soddisfi l'equazione della curva ellittica. Questo punto zero è considerato un punto della curva a tutti gli effetti. L'ordine della curva è il numero di punti distinti della curva, incluso il punto zero.

Avendo definito l'addizione tra due punti, possiamo anche definire la moltiplicazione tra un punto *F* e un intero positivo *k* come la somma del punto con se stesso ripetuta *k* volte: *k*F*

La sicurezza del sistema a curva ellittica è basata sulla difficoltà di calcolare *k* dato *F* e *k*F.*

Crittografia a curva ellittica (4)

Utilizzo della crittografia a curva ellittica:

- Un insieme di utenti si accordano su una curva ellittica comune e su un punto segreto fissato *F*. Ogni utente sceglie poi un numero intero segreto, (ad es. l'utente A sceglierà un numero segreto A_k), e pubblicherà poi una chiave pubblica pari al prodotto tra la sua chiave segreta e il punto *F*, (A avrà dunque la chiave pubblica $A_p = A_k * F$).
- Ora, supponiamo che l'utente A voglia inviare un messaggio all'utente B. Un possibile procedimento consiste nel far sì che l'utente A calcoli semplicemente $A_k * B_p$ e utilizzi il risultato come chiave segreta per un convenzionale sistema di cifratura a chiave simmetrica a blocchi, ad esempio DES. L'utente B può poi calcolare lo stesso numero calcolando $B_k * A_p$, dal momento che risulta

$$B_k * A_p = B_k * (A_k * F) = (B_k * A_k) * F = A_k * (B_k * F) = A_k * B_p$$

Crittografia a curva ellittica (5)

Viene detto *punto fisso c* un punto sulla curva tale che $c*F = 0$, dove $c = b - a$ e $a*F = b*F$, con $b > a$; l'ultimo c per cui vale ciò è detto ordine del punto F.

Per maggior sicurezza, la curva e il punto fisso sono scelti in modo tale che l'ordine del punto fissato F sia un numero primo grande.

Con le specifiche suddette, se l'ordine del punto fissato F è un numero primo di n bit, allora per calcolare k da $k*F$ e F sono necessarie pressappoco $2^{n/2}$ operazioni.

Questo significa, ad esempio, che se l'ordine di F è un numero primo di 240 bit, allora un attacco avrebbe bisogno, per avere successo, di circa 2^{120} operazioni. Questo è il motivo per cui l'utilizzo delle curve ellittiche è interessante: permette di avere chiavi pubbliche e firme digitali molto più piccole di quelle previste ad esempio per RSA, fornendo il medesimo livello di sicurezza, nonché maggior velocità.

Università degli Studi di “Roma Tre”

Moneta Elettronica

Proietto Luca

Sistemi di pagamento con Moneta Elettronica

1. Approccio innovativo e di ultima generazione che permette la compravendita direttamente durante la navigazione in Internet.

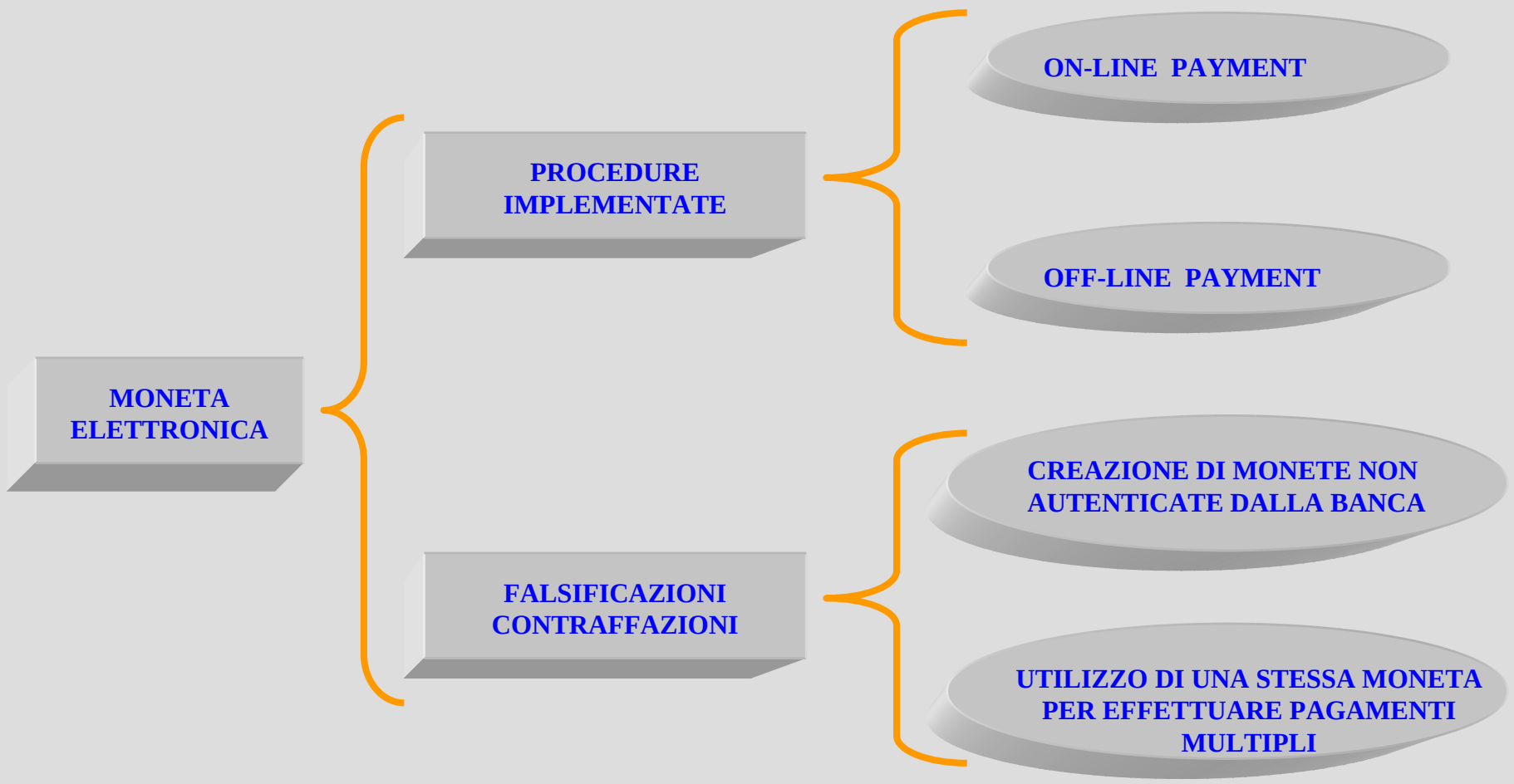
2. La Moneta Elettronica risulta essere a tutti gli effetti un titolo di credito digitale opportunamente firmato da un soggetto bancario.

3. La Moneta Elettronica è immateriale ed essendo una sequenza di bit può essere utilizzata per qualsiasi attività di compravendita tramite reti telematiche.

4. La memorizzazione di moneta elettronica avviene su supporto fisico (hard disk).

5. Esempi di Sistemi che supportano tale tecnologia sono E-Cash e DigiCash.

Sistemi di pagamento con Moneta Elettronica



PROCEDURE

ON – LINE PAYMENT → Questa modalità di pagamento prevede che il venditore verifichi la validità delle monete dell'acquirente prima di accettare il pagamento

OFF – LINE PAYMENT → Il venditore verifica la validità delle monete elettroniche dell'acquirente solo dopo che l'ultima transazione di pagamento sia stata completata

CONTRAFFAZIONI

MONETE NON AUTENTICATE → L'originalità di questa tecnologia sta nel fatto che ogni utente crea localmente la stringa (numero di serie delle future monete). Queste potranno essere definite come monete digitali solo quando avranno apposta la firma cieca della banca. Stringa di bit che non possiedono la firma cieca della banca NON sono considerate valide

PAGAMENTI MULTIPLI CON UNA STESSA MONETA → Per evitare che un utente utilizzi due volte la stessa moneta elettronica, la banca mantiene una base di dati con il numero di serie delle monete già spese

Perplessità sulla gestione della persistenza

CODICE_NUMERICO_MONETA	PARAMETRO_CASUALE_PAGAMENTO	CARTELLO_SFIDA_VENDITA	TIMESTAMP

- **Quanto tempo una tupla deve essere persistente nella tabella ? (validità moneta)**
- **Gestione critica di transazioni concorrenti che gestiscono una stessa tupla (pagamenti effettuati in sequenza)**

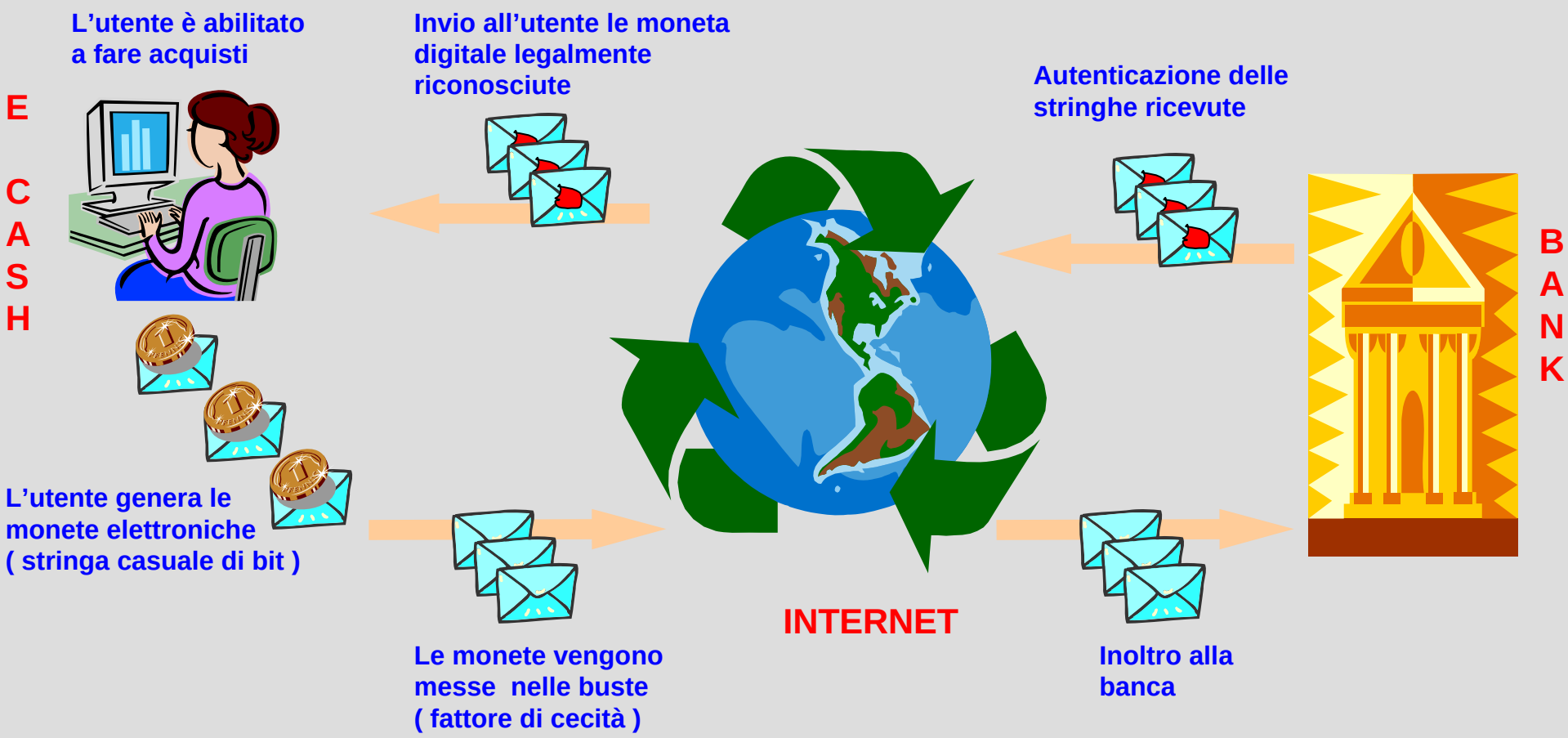
Firma cieca

1. Le tecniche di firma cieca (blind signature) permettono di far firmare un documento senza rivelare il contenuto
2. Metaforicamente l'effetto della firma cieca corrisponde a mettere un documento all'interno di una busta con un foglio di carta carbone. Se qualcuno firma la busta avrà contemporaneamente firma anche il documento all'interno della busta senza esserne necessariamente al corrente del contenuto.
3. Va sottolineato che la firma rimane sul documento anche quando quest'ultimo viene estratto dalla busta.
4. La tecnologia della firma cieca trova largo impiego oltre che nei sistemi di pagamento elettronico con monete virtuali anche nei sistemi che gestiscono votazioni elettorali elettroniche.

Tipico scenario di prelievo

1. Il PC dell'utente su cui è installato il sistema calcola quante monete sono necessarie per ottenere la somma richiesta.
2. Lo stesso utente tramite la sua applicazione genera un numero di monete pari alla quantità sopra calcolata, associando ad ognuna di esse un numero di serie casuale. Quindi spedisce queste monete una ad una inserite in una speciale busta. (rappresenta il fattore "cecità").
3. 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, contemporaneamente la banca addebita sul conto dell'utente la stessa somma.
4. 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 relative firme costituiscono la moneta digitale garantita dalla banca.
5. Quando l'utente spenderà tali monete, la banca le accetterà poiché da lei autenticate ma non potrà risalire all'identità dell'utente in quanto i numeri di serie delle monete erano nascosti (nella busta) al momento dell'autenticazione da parte della banca stessa.

Tipico scenario di prelievo



FERGUSON SCHEME

(come identificare ciascuna moneta elettronica)

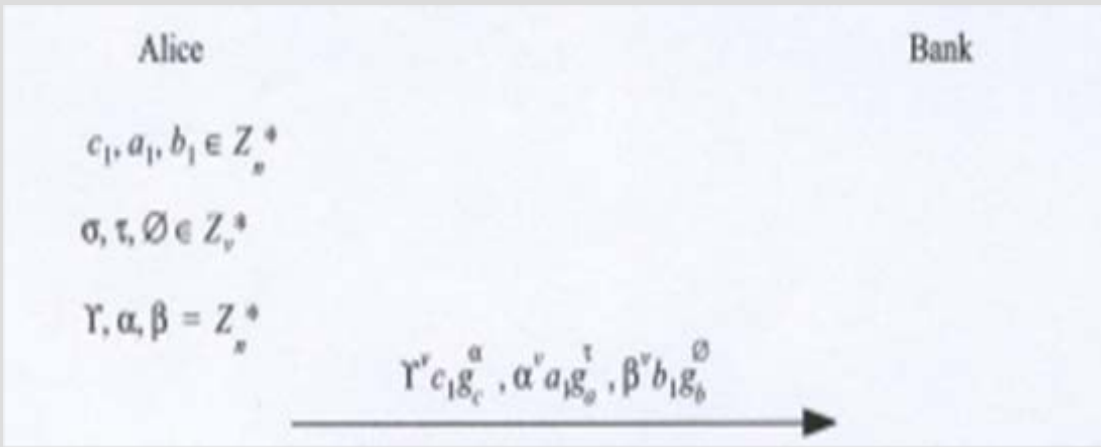
c, a, b (numeri base)

k (parametro casuale per il profilo di condivisione)

Sa , Sb (double blind signatures)

FERGUSON SCHEME

(protocollo di prelievo – prima fase)



L’utente (nel caso specifico Alice) sceglie i tre suoi contributi casuali che andranno a formare i tre numeri base relativi alla moneta. Inoltre i tre valori numerici vengono opportunamente nascosti da tre fattori di cecità scelti sempre dall’utente stesso.

FERGUSON SCHEME

(protocollo di prelievo – seconda fase)

$c_2, a_2, b_2 \in Z_n^*$

$\xleftarrow{\quad h_c^{c_2},\ c_2,\ h_b^{b_2} \quad}$

$k_1 \in Z_v^*$

$e_c = (f(h_c^{c_1 c_2}) - \sigma)$

$e_b = (f(h_b^{b_1 b_2}) - \varnothing)$

$a = (a_1 a_2 f_2(e_c, e_b))^{k_1}$

$e_a = (1/k_1) f(a) - \tau$

$\xrightarrow{\quad e_c, e_a, e_b \quad}$

$\bar{C} = \Upsilon^v c_1 g_c^{\varnothing} c_2 g_c^{e_c}$

$\bar{A} = \alpha^v a_1 g_a^{\tau} a_2 f_2(e_c, e_b) g_a^{e_a}$

$\bar{B} = \beta^v b_1 g_b^{\varnothing} b_2 g_b^{e_b}$

$k_2 \in Z_v^*$

La banca spedisce all'utente un suo contributo ad un solo numero base ed altri due numeri che sono così calcolati: la base della potenza è nota e i due esponenti sono gli altri due contributi ai numeri base. Quindi in questa fase solo un contributo viene inviato in modo esplicito dalla banca.

FERGUSON SCHEME

(protocollo di prelievo – terza fase)

$\xleftarrow{\quad c_2, b_2, k_2, (\bar{C}^{k_1}\bar{A})^{1/v}, (\bar{C}^{U}\bar{B})^{1/v} \quad}$

$c = c_1 c_2$

$b = b_1 b_2$

$k = k_1 k_2$

$C = c g_c^{f(k_c^{\tau})}$

$A = a g_a^{f(a)}$

$B = b g_b^{f(a_b^{\tau})}$

$S_a = ((\bar{C}^{k_2}\bar{A})^{1/v} / \Upsilon^{k_2}\alpha)^{k_1}$

$S_b = ((\bar{C}^{U}\bar{B})^{1/v} / (\Upsilon^{U}\beta))$

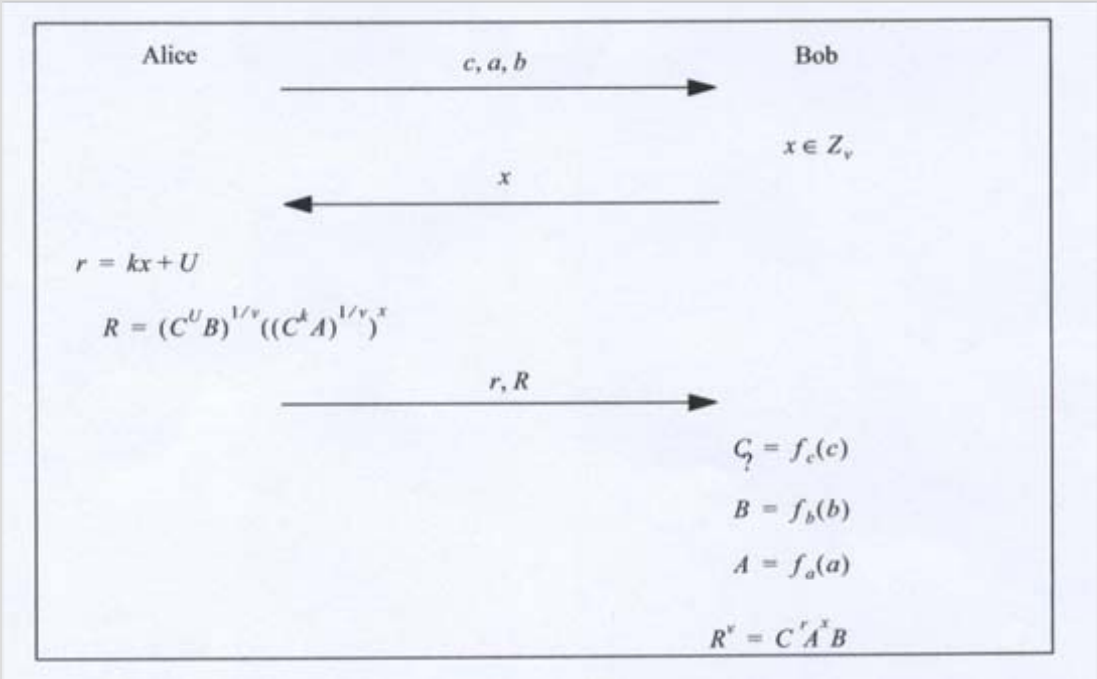
$S_b^v \stackrel{?}{=} C^k A$

$S_a^v \stackrel{?}{=} C^{\sigma} B$

In questa fase della comunicazione la banca invia all'utente gli altri due contributi per i numeri base della moneta ed una parte essenziale nel calcolo della firma che permetterà alla moneta di circolare in modo legalmente riconosciuto.

FERGUSON SCHEME

(protocollo di pagamento)



L'acquirente, come forma di pagamento, invia i tre numeri base relativi alla moneta, riceve un cartello di sfida dal venditore e con questo ultimo comunica una linea univoca di transazione associata alla moneta e al suo utilizzo.

FERGUSON SCHEME

(caratteristiche del protocollo)

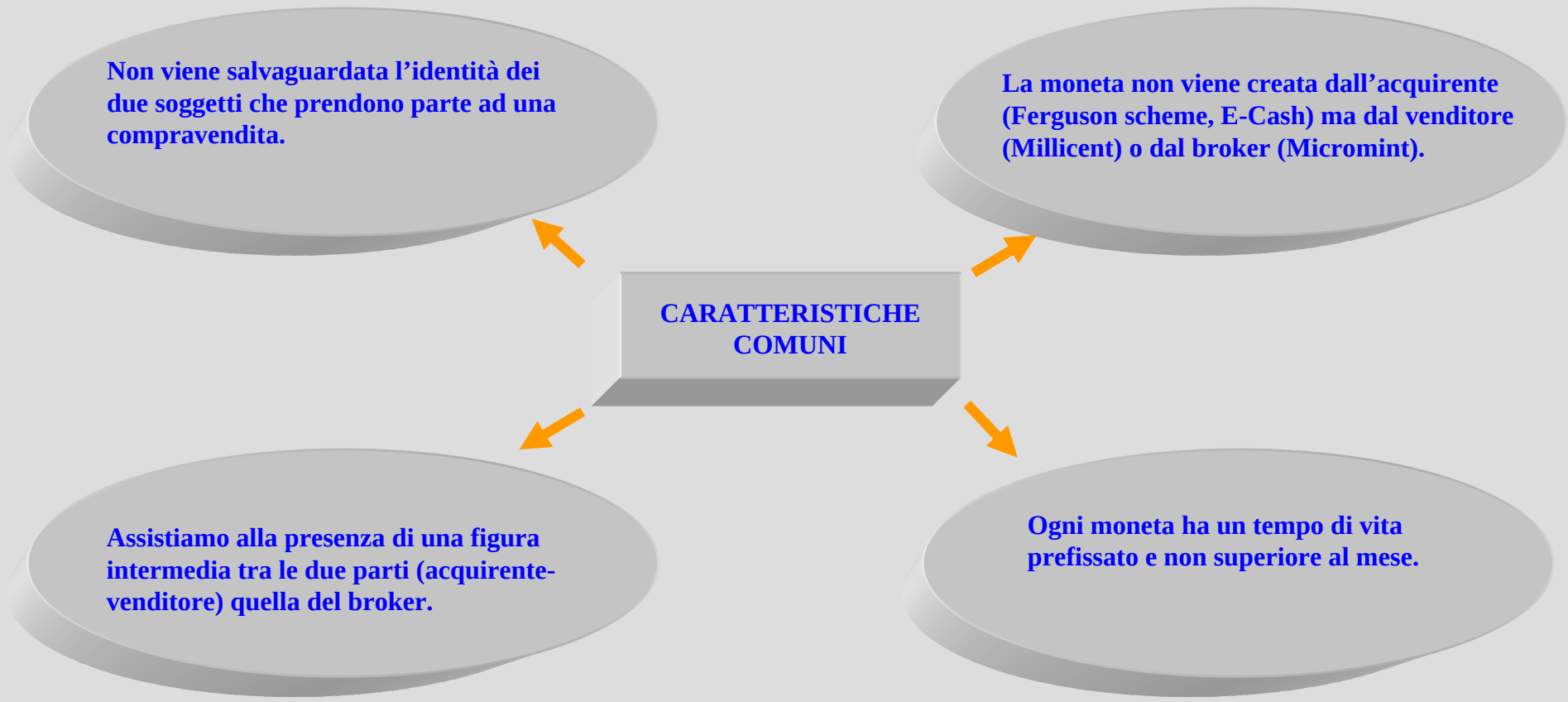
- Sicurezza : per prevenire l'utilizzo multiplo di una stessa moneta vengono adottate due soluzioni. Ogni moneta spesa viene memorizzata in un base di dati gestita dalla banca e in secondo luogo ogni volta che l'utente spende una moneta deve associargli un parametro di transizione opportunamente negoziato con il venditore.

- Privacy : La banca autentica delle monete, attraverso la firma digitale, di cui non conosce il numero di serie. In questo modo la banca è sempre in gradi di riconoscere le monete valide ma non può risalire all'identità dell'utente che le ha spese.

- Portabilità : Ogni moneta elettronica (stringa di bit opportunamente ottenuta) consiste di 3 numeri base, un elemento univoco di transizione e 2 firme digitali cieca. Queste informazioni possono essere immagazzinate all'incirca in 250 byte, cifra dalle dimensioni confortevoli ed estremamente portabile.

- Divisibilità : Questo schema permette di trattare prezzi interi e multipli dell'unità di valore stabilita per la singola moneta elettronica. E' estremamente oneroso inglobare nella stringa di bit relativa ad una moneta anche un suo eventuale valore di scambio.

PROTOCOLLI di PAGAMENTO
(Millicent, Micromint)



MILLICENT

Millicent è un protocollo progettato per l'implementazione della moneta elettronica sul server del commerciante senza :

- comunicazioni addizionali
- onerose operazioni di crittografia
- processing fuori linea

Il protocollo Millicent si basa su Scrips e su Brokers ……..

MILLICENT

- Quando un cliente effettua un acquisto con lo scrip, il costo dei beni è scalato dallo scrip stesso.
- Il cliente stabilisce un account con il broker, mentre quest'ultimo lo stabilisce con il commerciante.
- Un cliente ha un solo account con il broker, mentre ciascun venditore ha un ristretto numero di account verso più brokers.
- Il cliente ed il venditore possono controllare indipendentemente il loro bilancio.
- Il venditore è in grado di controllare la validità dello scrip senza dover necessariamente memorizzare tutti i movimenti effettuati dal cliente.
- Le identità di entrambi i soggetti che prendono parte ad una compravendita non sono tutelate dall'anonimato.
- Il Broker si pone tra il cliente ed il venditore e manipola tutte le transazioni di moneta reale.

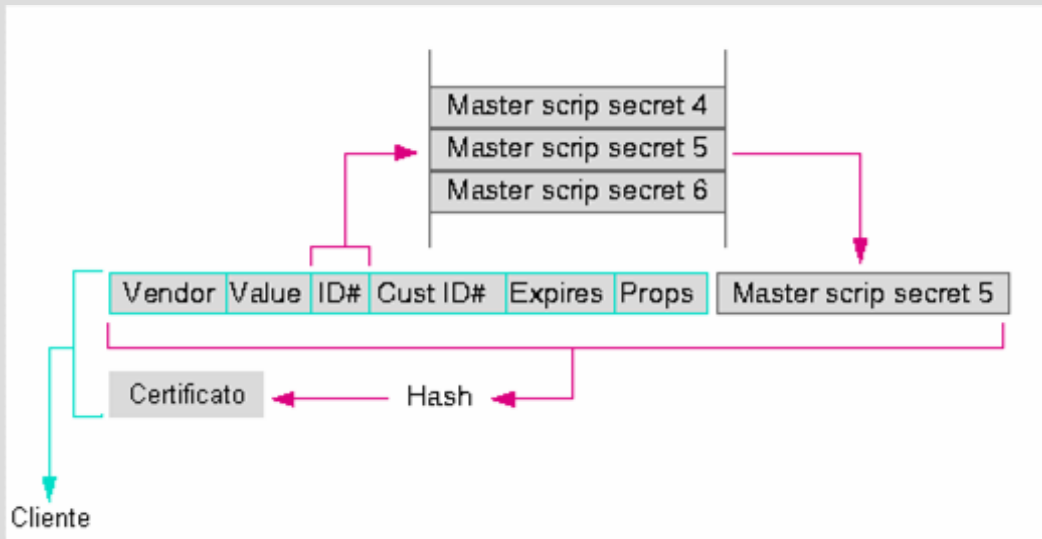
MILLICENT
(Campi dello Scrip)

Vendor	Value	ID#	Cust ID#	Expires	Props

Lo **Scrip** ha i seguenti campi:

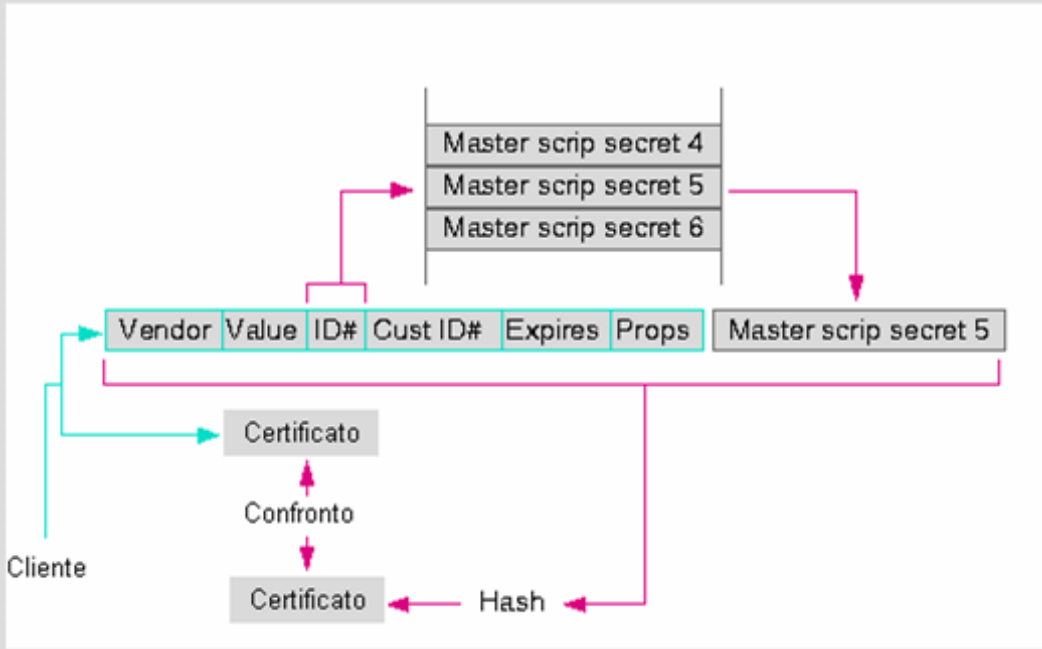
- Vendor: identifica il commerciante;
- Values: il valore dello scrip;
- ID#: identificatore dello scrip (parte di esso contiene il master_scrip_secret);
- Cust_ID: produce il customer_secret (da cui è possibile selezionare anche il customer_secret);
- Expires: indica la scadenza dello scrip;
- Props: descrive informazioni aggiuntive del cliente al commerciante;
- Certificate: è la firma dello scrip.

MILLICENT
(Certificazione dello Scrip)



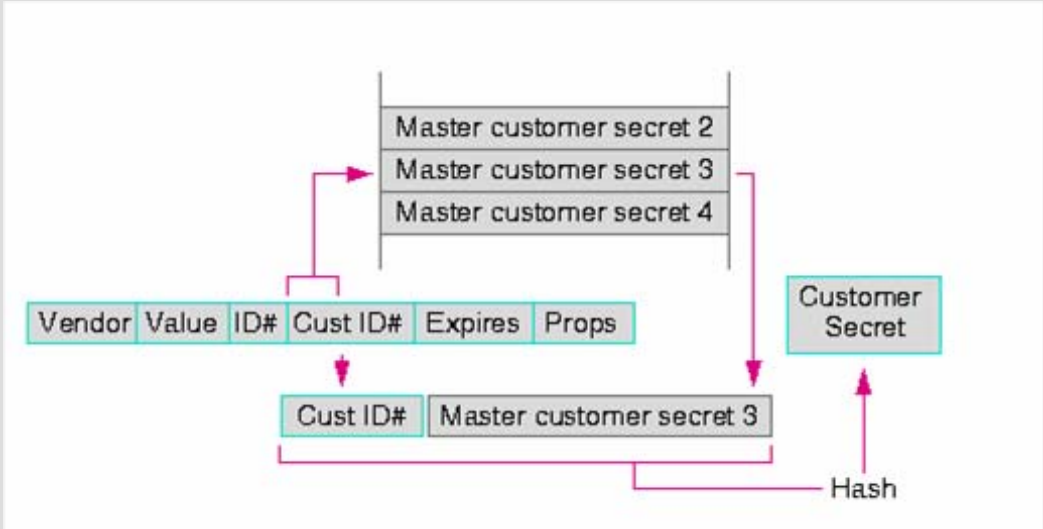
Il certificato relativo ad uno scrip viene generato dal venditore applicando una funzione hash al corpo dello scrip con l'aggiunta di una chiave segreta. Quest'ultima viene calcolata da un sottoinsieme del campo ID#. Al cliente verrà spedito lo scrip ed il certificato ad esso relativo.

MILLICENT
(Validazione e scadenza dello Scrip)



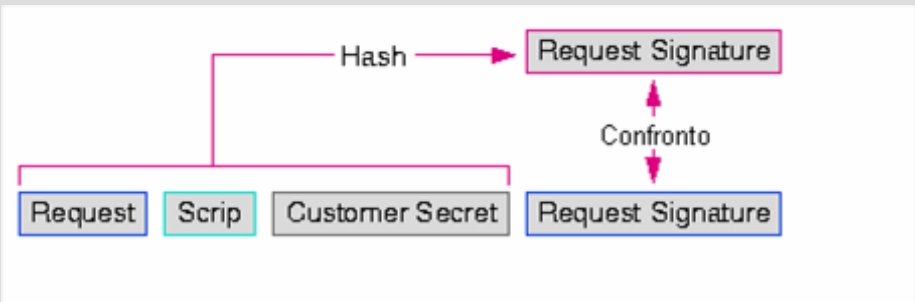
Lo scrip ricevuto è valido rigenerando il certificato e confrontandolo con quello ricevuto. Se sono identici lo scrip è valido. Il commerciante registra l'ID dello scrip in modo tale da evitare frodi. Il commerciante non memorizzerà perennemente l'ID dello scrip, in quanto esso scade. Alla scadenza lo scrip è sottoposto al commerciante il quale restituisce al cliente un nuovo scrip con una scadenza nuova e un nuovo numero seriale.

MILLICENT
(Costruzione del Customer Secret)



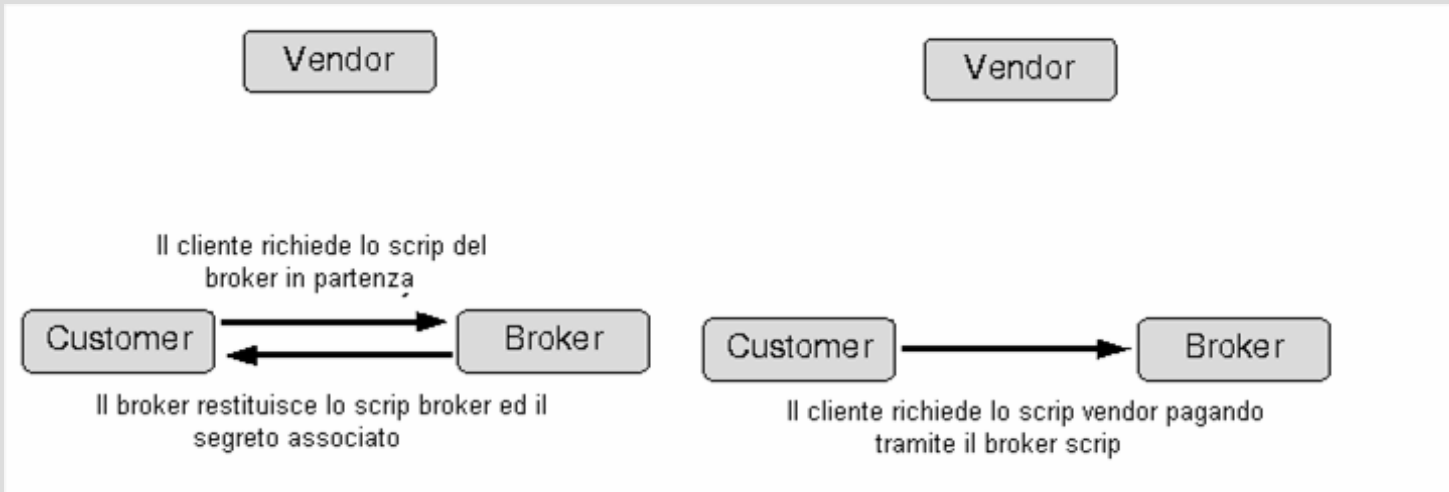
Il segreto del cliente è generato applicando una funzione hash all'ID del cliente con una chiave. Quest'ultima è ottenuta selezionando una porzione del Cust_ID.

MILLICENT
(Pagamento relativo ad una richiesta di acquisto)



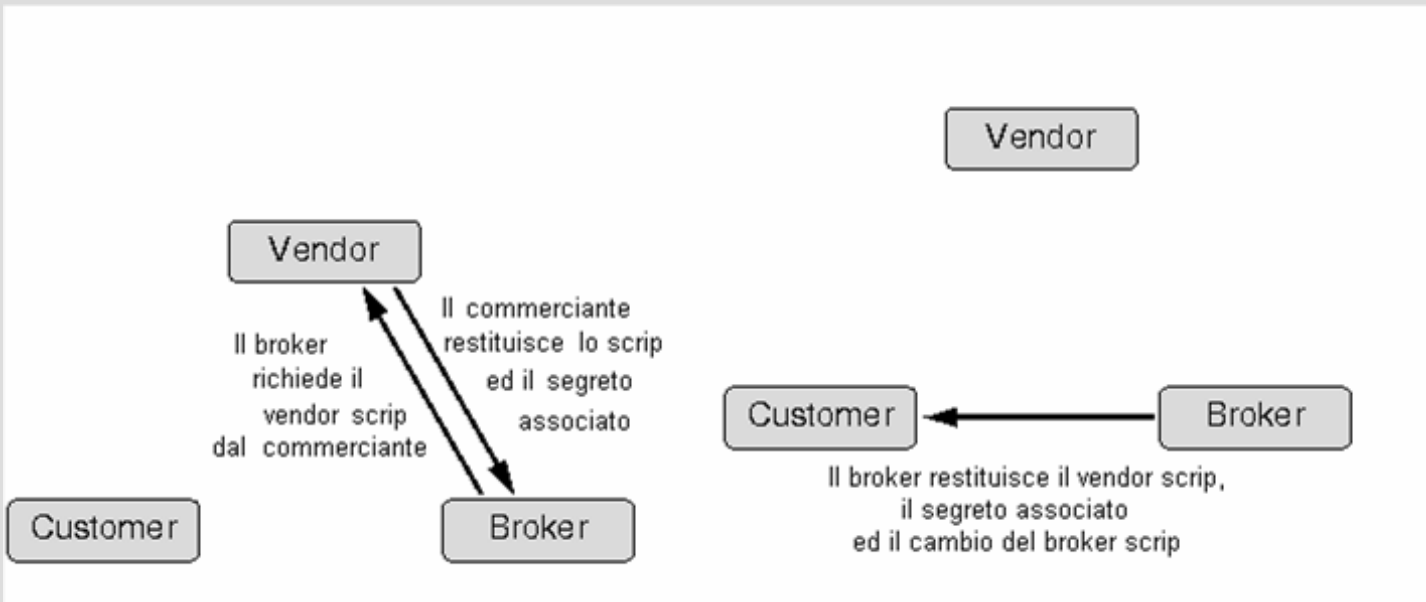
Per acquistare il cliente invia la richiesta (scrip) ed una firma al venditore. La firma è prodotta nello stesso modo in cui è prodotto il certificato dello scrip. Lo scrip e la richiesta sono concatenate al customer secret.
Il cliente esegue una funzione crittografica hash efficiente su tale stringa ed invia il risultato come firma. Quando il commerciante riceve la richiesta deriva il customer secret dallo scrip e rigenera la firma per la richiesta. In caso di alterazioni non c'è match tra le firme.

MILLICENT
(Interazione Customer - Broker)



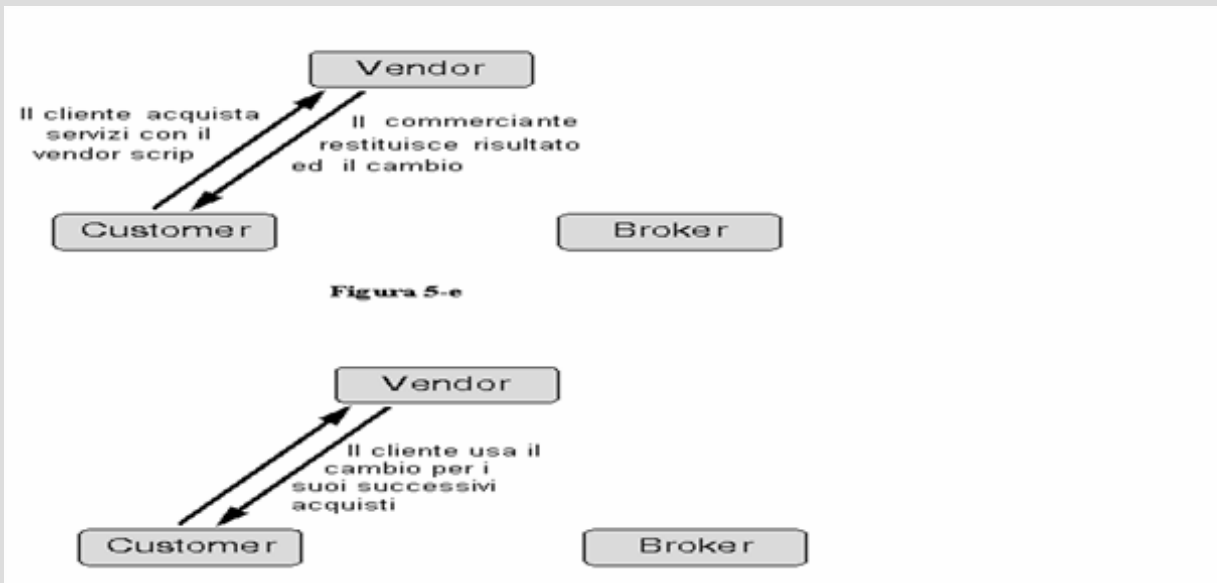
- Il primo passo viene eseguito una sola volta nel corso dell'intera sessione.
- Il secondo passo si rende necessario ogni volta che il cliente non ha memorizzato lo scrip relativo a quel determinato venditore.

MILLICENT
(Interazione Vendor - Broker)



- Il terzo passo si ha quando il broker deve acquistare uno scrip da un determinato commerciante.
- Il quarto passo si ha quando il broker restituisce il vendor scrip al cliente che lo ha richiesto precedentemente.

MILLICENT
(Interazione Vendor - Customer)



- Il quinto passo mostra il cliente che utilizza lo scrip per acquistare.
- L'ultimo passo mostra una tipica transazione Millecent: il cliente ha già lo scrip del commerciante e lo utilizza per effettuare acquisti.

MICROMINT
(caratteristiche generali del protocollo)

- Le monete sono prodotte da un broker che le distribuisce agli utenti quando ne fanno esplicita richiesta.
- Gli utenti 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 produrre.
- Ogni moneta ha una validità prefissata (all'incirca un mese).

MICROMINT
(formato della moneta elettronica)

moneta = $X_1, X_2, X_3, \ldots\ldots, X_n$

$h(X_1) = h(X_2) = h(X_3) = \ldots \quad h(X_n) = Y$
(condizione necessaria ma non sufficiente di validità)

h = funzione hash

MICROMINT
(la centralità e flessibilità della figura del broker)

- Il broker inizia la distribuzione delle monete verso la fine di ogni mese. Questa scadenza è soggetta a particolari cambiamenti per motivi di sicurezza. In questo modo il broker tende a non lasciare alcun punto di riferimento a probabili falsari.

- Le monete distribuite verranno poi utilizzate nel mese successivo quando sarà reso pubblico anche il criterio per ritenerle valide.

- I clienti che comprano queste monete scaricano il loro acquisto sulla loro carta di credito. In molti casi questa procedura può essere effettuata anche dal broker stesso.

- Il broker da parte sua tiene traccia delle monete distribuite agli utenti. Al contrario di quanto viene teorizzato nel Ferguson scheme e applicato concretamente nel sistema E-Cash, non viene palesemente garantito l'anonimato delle parti che prendono gioco ad una compravendita.

MICROMINT
(principi basilari di scambio)

- Ogni volta che un cliente deve pagare un prodotto ad un venditore gli spedisce la serie x = x1, x2, x3, ……., xk che forma la moneta.
- Il venditore controlla che sia una moneta valida calcolando la funzione hash su ognuno dei singoli valori e verificando che risulti un numero per tutti uguale (collisione hash a k-vie).
- Il venditore ogni giorno restituisce al broker le monete che ha accumulato.
- Il broker controlla le varie monete tentando di individuare quelle che sono state già riscattate poiché mantiene traccia completa delle monete distribuite e di quelle gia spese.
- Se giungono al broker monete valide, quest'ultimo provvederà a pagare al venditore la somma stabilita. Se al contrario giungono al venditore monete uguali da differenti venditori, allora il broker deciderà di pagare casualmente uno dei contendenti a discapito di tutti gli altri.

MICROMINT
(contraffazioni)

- Contraffazioni a piccola scala : con un protocollo così strutturato sono del tutto inapplicabili. Questo è dimostrato dal fatto che una normale workstation può effettuare solamente 2^14 operazioni hash al secondo contro le 2^45 operazioni hash minime richieste per creare una moneta. La normale workstation impiegherebbe circa 80 anni !!
- Contraffazioni a larga scala : per contrastare falsificatori che possiedono potente hardware dedicato come quello del broker possono essere messe in atto diverse strategie: dichiarare dinamicamente e casualmente la validità delle monete, generare monete nuove sempre più frequentemente e memorizzare in modo sempre più accurato e puntuale identità delle parti che utilizzano le monete elettroniche.

Simple Anonymous Cash (Chaum-Fiat-Naor)

Contanti anonimi semplici

Neri Claudio

Pagamenti elettronici anonimi: Chaum-Fiat-Naor

Presentiamo prima i soggetti coinvolti in una transazione economica:

1. Alice, titolare del conto, la persona generica che vuole effettuare l'acquisto;
2. Bob, la persona generica che vende il bene o servizio;
3. La banca, l'ente finanziario cui Alice si appoggia

Protocollo di Prelevamento

- La banca pubblica una chiave RSA (e,n) e regola un parametro di sicurezza k.
- La banca pubblica due funzioni, *f* e *g*, dove *f* produce un valore casuale mentre g è una funzione di valore univoco.
- Alice ha un numero di conto bancario *u* di cui la banca mantiene un contatore *v*.

Protocollo di Prelevamento

- Alice crea *k* unità campioni, U_i.
- Ogni unità è data da numeri di serie casuali, a_i, c_i, d_i, $1 \leq i \leq k$, scelti da un' insieme abbastanza grande per accertarsi che nessun' altra unità ottenga lo stesso valore.

$$U_i = f(x_i, y_i) \quad 1 \leq i \leq k$$

Dove $x_i = g(a_i, c_i)$, $y_i = g(a_i \oplus (u \wedge (v + i)), d_i)$

Indicando con $\oplus$ l'or esclusivo e ^ la concatenazione

Protocollo di Prelevamento

- Alice maschera queste k unità con fattori casuali $\{r_1 \ldots r_k\}$ e li trasmette alla banca.

$$B_i = r_i^{b_i} U_i \mod n$$

- Alla banca viene quindi nascosto il contenuto.

Protocollo di Prelevamento

- La banca sceglie casualmente *k/2* unità per eseguire il test.
- Alice fornisce alla banca
  - r_i, a_i, c_i, d_i, per $1 \leq i \leq k/2$ (supponendo che la banca abbia scelto la i tali che $1 \leq i \leq k/2$)

Protocollo di Prelevamento

- La banca verifica *k/2* unità e controlla che Alice non abbia provato a truffare.
- Se non ci sono errori, la banca firma le rimanenti k/2 unità con la relativa chiave riservata e le fornisce ad Alice

$$\prod_{k/2 \le j \le k} (B_i)^d \bmod n$$

ed addebita il suo cliente.

Protocollo di Prelevamento

- Alice decripta le unità firmate moltiplicando per l'inverso r_j^{-1}.
- Alice ha una moneta elettronica ed incrementa la copia del suo conto personale *v* con la moneta digitale ottenuta.

$$C = \prod_{k/2 \le j \le k} f(x_i, y_i)^d \bmod n$$

Protocollo di Pagamento

- Alice spende la moneta digitale ottenuta dalla banca, la C, per pagare Bob.
- Bob sceglie una stringa binaria casuale $z_1, z_2, \ldots, z_{k/2}$ e la trasmette ad Alice.
- Alice risponde come segue, per ogni $1 \leq i \leq k/2$
  - a- se $z_i = 1$ allora Alice invia a Bob a_i, c_i e y_i
  - b- se $z_i = 0$ allora Alice invia a Bob $x_i,\ a_i \oplus (u \Lambda (v+i))\ d_i$
- Bob verifica che la C sia valida prima di accettare il pagamento da parte di Alice.

Protocollo di Deposito

- Bob trasmette i pagamenti alla banca.
- La banca verifica che sulla moneta sia apposta la propria firma digitale e che la moneta non sia già stata spesa.
- La banca registra la moneta nella base di dati della moneta spesa ed inoltre immagazzina la stringa binaria e la risposta corrispondente di Alice.
- La banca accredita l'importo sul conto personale di Bob.

Caratteristiche del protocollo

- ***Sicurezza:*** Uno degli atti fraudolenti più semplici qui è double-spending. Per minimizzare questo atto la banca registrare il numero della moneta in una base di dati, quindi se lo stesso numero compare due volte, la banca sa che è stato commesso un atto fraudolento.
- ***Portabilità:*** Essendo queste monete costituite da numeri digitale e casuali sono facili da trasportare.

Caratteristiche del protocollo

- ***Trasferibilità:*** Questo schema è non trasferibile proprio perché queste monete non possono essere spese parecchie volte. Devono essere restituite alla banca dopo ogni transazione.
- ***Divisibilità:*** Questo sistema non dà alcuna alternativa ad una moneta divisibile.

Attacchi

- Essendo basato su RSA, questo schema è vulnerabile a tutti gli attacchi del RSA.
- Attacco di cooperazione fra Alice ed una eventuale terza persona.
  - Se Alice dopo una transazione di pagamento con Bob, trasmette la sua moneta spesa ad una terza persona con la stringa binaria scelta da Bob e la risposta a questa stringa, questa terza persona avrà una storia di pagamento esatta. La banca non potrà determinare quale di loro sta truffando.

Divisible Electronic Cash

Puleggi Leonardo

Problemi

- I sistemi di pagamento digitale descritti finora non hanno la possibilità di passare il denaro a più persone né di dividerlo in parti più piccole

Panoramica

Year	Name	Cut-and Choose?	Underlying Problem	Unlinkable?
91	Universal Electronic Cash	Yes	Factoring	No
94	Single-term divisible electronic coins	No	Brands's scheme	No
95	An Efficient Divisible Electronic Cash Scheme	No	Factoring	No
98	Easy come-easy go divisible cash	No	Factoring	No
00	Unlinkable Divisible Electronic Cash	No	Discrete Log	Yes

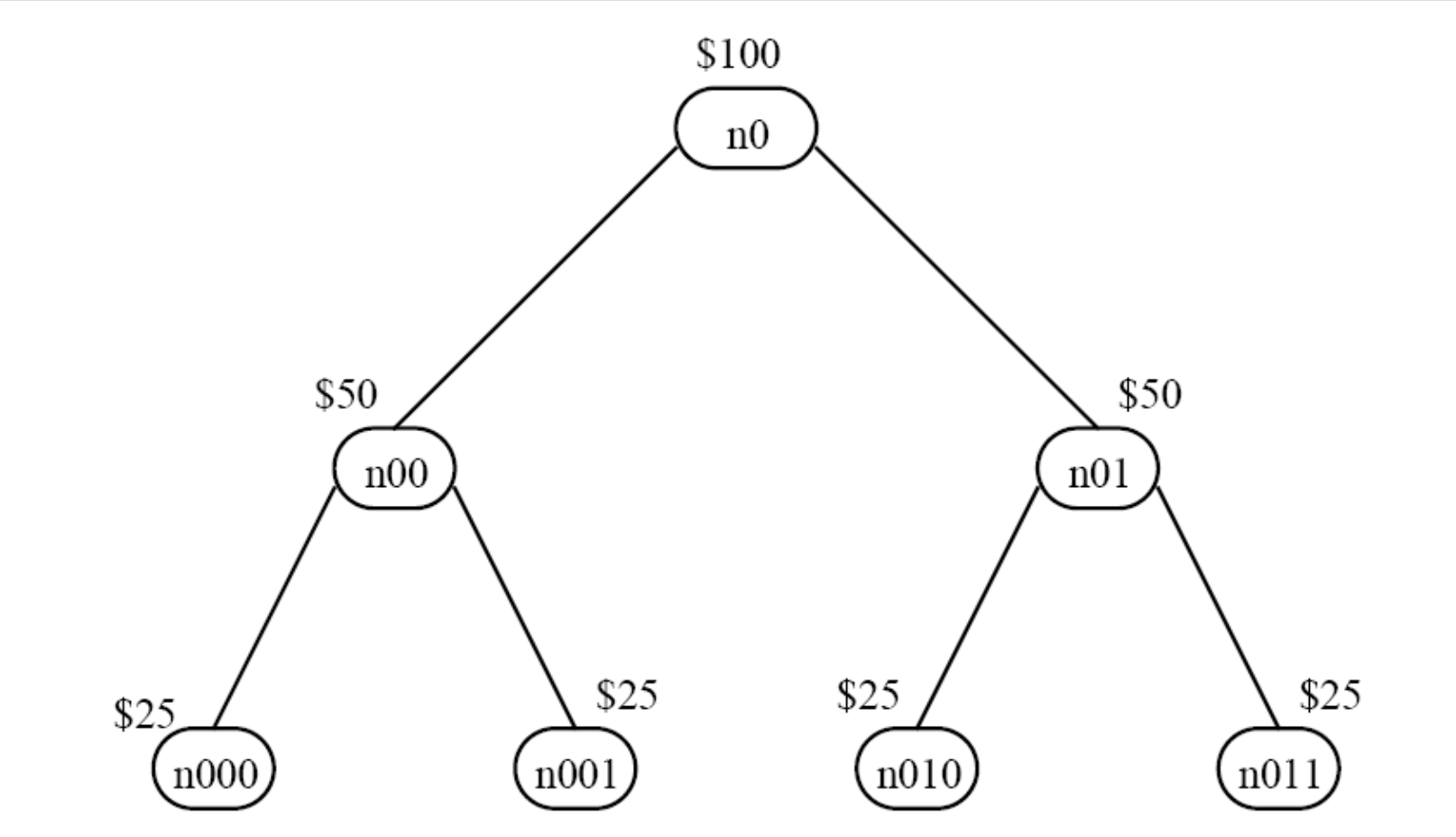
Una soluzione

- Questo sistema è stato proposto da **Okamoto** ed è basato sullo schema di bit commitment già visto, sulla radice quadrata modulo N e una rappresentazione dell'albero binario
- La sicurezza di questo sistema si basa sulla fattorizzazione (poichè utilizza l'RSA) e su funzioni hash

Rappresentazione dell'albero binario

- La rappresentazione dell'albero binario gioca un ruolo importante poiché permette di **dividere** la moneta elettronica in molti pezzi, in modo che poi questi possano essere usati separatamente

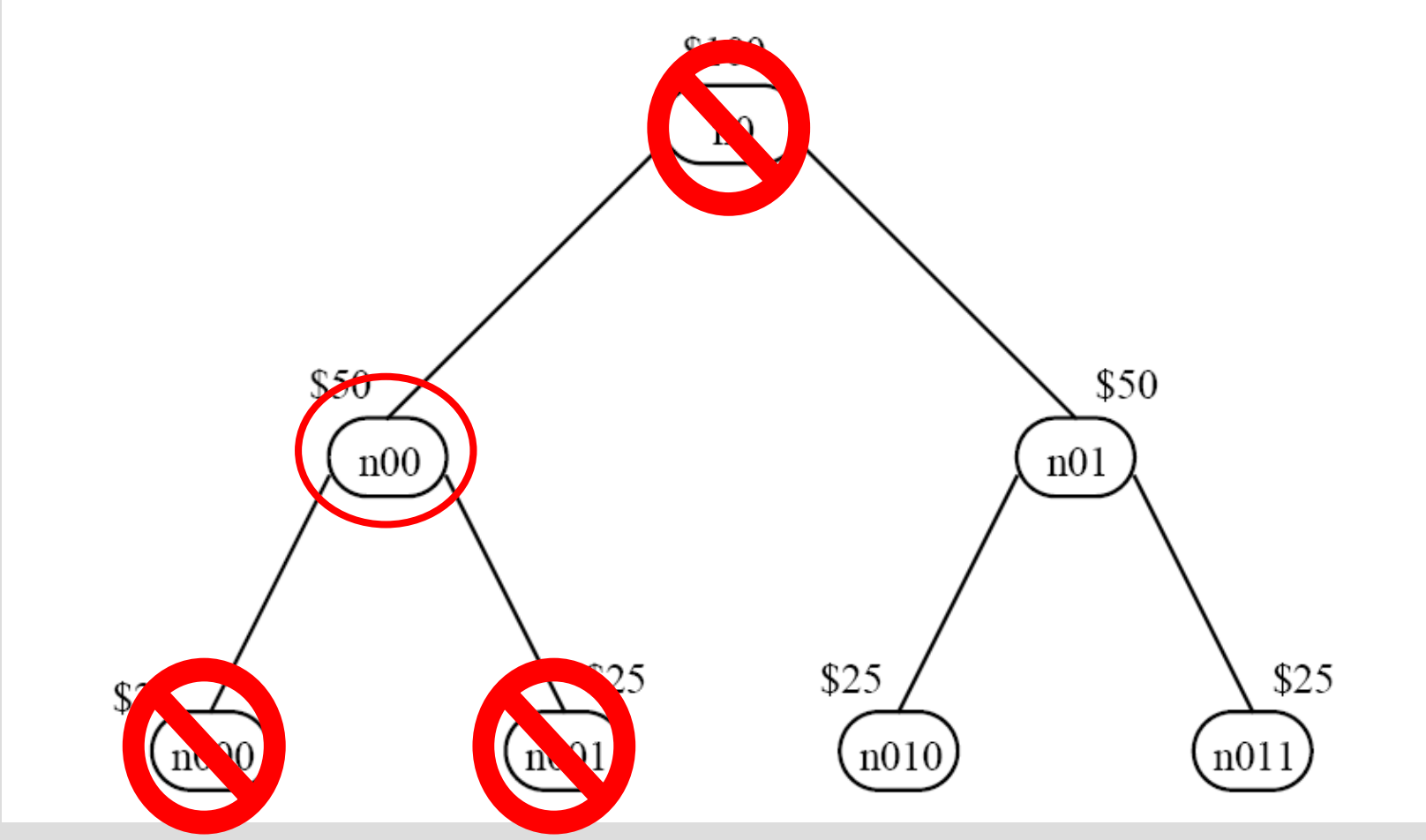
Rappresentazione dell'albero binario



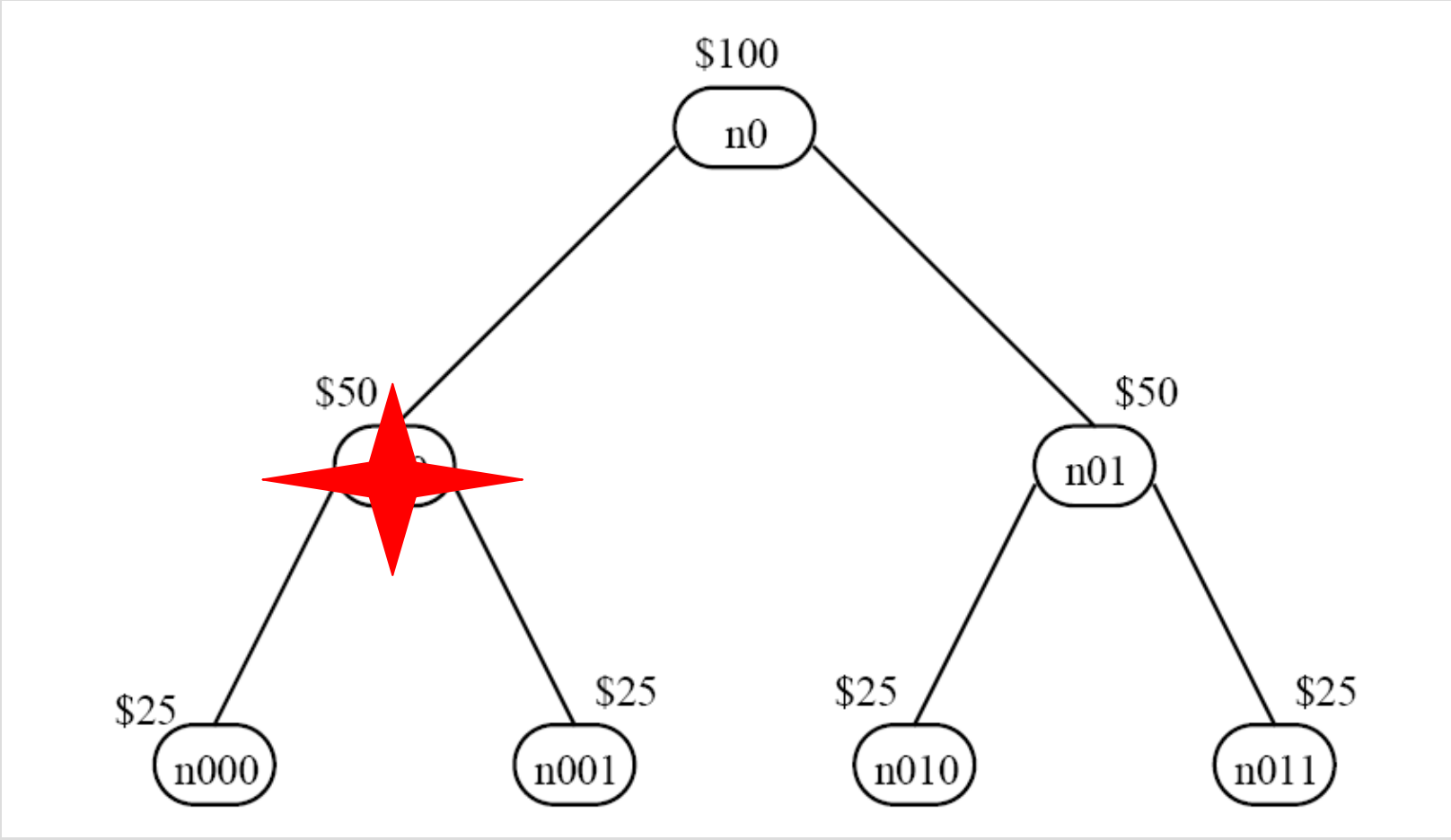
Regole

- Il metodo proposto da Okamoto si basa sull'uso di 2 regole
  1. **Nodi intermedi** - quando un nodo è usato non è possibile usare nessuno dei sui figli né un suo genitore
  2. **Nodo uguale** – un nodo non può essere usato più di una volta

Regole : esempi



Regole : esempi



Osservazioni

- **Oss** – Le regole garantiscono che un utente non possa usare una moneta di un valore maggiore di quello richiesto alla banca
- **Oss** – La profondità dell'albero dipende dal valore iniziale della moneta e dalla precisione desiderata (ad esempio una moneta da $1000 da cui si vogliano generare i cent richiederà un albero di $\log_2(1000*100)$ livelli

Protocollo di apertura conto

- Questo protocollo serve per fornire al cliente una licenza elettronica, che successivamente può essere usata per manipolare le monete della banca

Protocollo di apertura conto(1)

1. La banca pubblica le sue chiavi pubbliche (n_1,k), (n_2,k) e a_1,a_2(sia **RSA** che delle firma cieca) e un generatore g
2. Il cliente controlla che siano valide e genera due numeri primi **p** e **q**

$$p = 3 \bmod 8 \ , \ |p| < k \ , \ p > (1/4)\, n_1^{\frac{1}{2}}$$

$$q = 7 \bmod 8 \ , \ |q| < k \ , \ q > (1/2)\, n_1^{\frac{1}{2}}$$

Protocollo di apertura conto(2)

3. Invia poi **x** e **y** alla banca(questi rappresenteranno l'identità del cliente)

$$x = g^{p} \bmod p$$

$$y = g^{q} \bmod p$$

Fine della parte relativa allo schema di bit commitment

Protocollo di apertura conto(3)

4. La banca, a questo punto, invia al cliente **N** e le sue chiavi pubbliche **a1** e **a2** firmate
5. Il cliente calcola allora

$$L_1 = (N + a_1)^{1/k} \bmod n_1$$

$$L_2 = (N + a_2)^{1/k} \bmod n_2$$

6. La licenza elettronica del cliente è formata da (L1,L2,N)

Protocollo di prelievo

- La transizione di prelievo è molto semplice perché consiste in una firma sul numero **N** concatenato ad un numero casuale **b**(la moneta è quindi unica)
- Assumiamo che il cliente voglia ritirare un moneta da **w=2^l** (la banca avrà diverse chiavi pubbliche per i vari tagli) a cui corrisponde la chiave (e_w,n_w)

Protocollo di prelievo(1)

1. Il cliente sceglie un numero casuale b e lo concatena con N
2. Questo valore viene passato per una funzione hash unidirezionale per ottenere una moneta C

$$z = r^{e_w} \ \mathrm{H(N,b)} \mod n_w$$

Protocollo di prelievo(2)

3. La banca firma la moneta che è stato inviato e toglie il valore corrispondente a w dal conto del cliente

$$\mathrm{sign(z)} = Z^{\frac{1}{e_w}} \mod n_w$$

4. Il cliente decifra la moneta e lo usa

$$C = (H(N\,,b))^{1/e_w} \mod n_w$$

Protocollo di pagamento

- Questo protocollo si divide in due parti: **autenticazione della moneta** e **rivelazione del taglio**

Pagamento: autenticazione della moneta

Durante questa fase il venditore verifica che la moneta sia legittima

- Il cliente fornisce la sua licenza,cioè (L1,L2,N), la moneta e il numero random per generarla (C,b) e infine il taglio del moneta w
- Il venditore deve semplicemente verificare la firma della banca sulla moneta e che

$$L_i \equiv N + a_i \mod n_i \,, (i = 1 \,, 2)$$

Pagamento: rivelazione del taglio

Il cliente in questa fase rivela delle informazioni sui nodi dell'albero che sono stati già usati

- Il cliente deve selezionare un nodo (il cui valore associato sia almeno pari a w) tra quelli che non violino le due regole
- Se lui non le rispetta? Tramite le informazioni che il cliente deve inviare, la banca è in grado di verificare immeditamente se la transizione è corretta ed eventualmente bloccarla

Protocollo di deposito

- Il commerciante manda una copia della transizione di acquisto alla banca
- La banca deve ovviamente verificare che la stesso moneta non sia già stata usata

Protocollo di deposito(1)

- Per verificare cerca nel suo database una moneta con gli stessi valori **C** e **b** e con la licenza elettronica (**L1**,**L2**,**N**)
- Verifica poi per quella moneta i nodi usati (grazie all'albero)
- Se una moneta è stata usata due volte può isolare N da cui risalire al cliente

Caratteristiche e-cash : sicurezza

- La caratteristica comune ai vari protocolli è la K che viene presentata all'inizio e rappresenta il numero di bit richiesto per la chiave
- Poiché tutto il sistema si basa su RSA possiamo consideraci al sicuro fintanto che non sia possibile realizzare un attacco polinomiale

Caratteristiche e-cash : privacy

- Ogni cliente riceve una licenza elettronica da mandare al negozio durante la fase del pagamento, legando quindi ad ogni pagamento la licenza e una particolare moneta
- Questo rende facile seguire i vari pagamenti

Non è garantita la segretezza

Caratteristiche e-cash : portabilità

- Ogni volta che il cliente vuole usare la moneta deve calcolare la rappresentazione di un albero binario
- Se prendiamo k=257 e una moneta da $1000 (con precisione fino ai centesimi)
- moneta + licenza=> 264 byte
- Per tutti i protocolli i dati totali occupano 3.2 KB

Caratteristiche e-cash : cedibilità

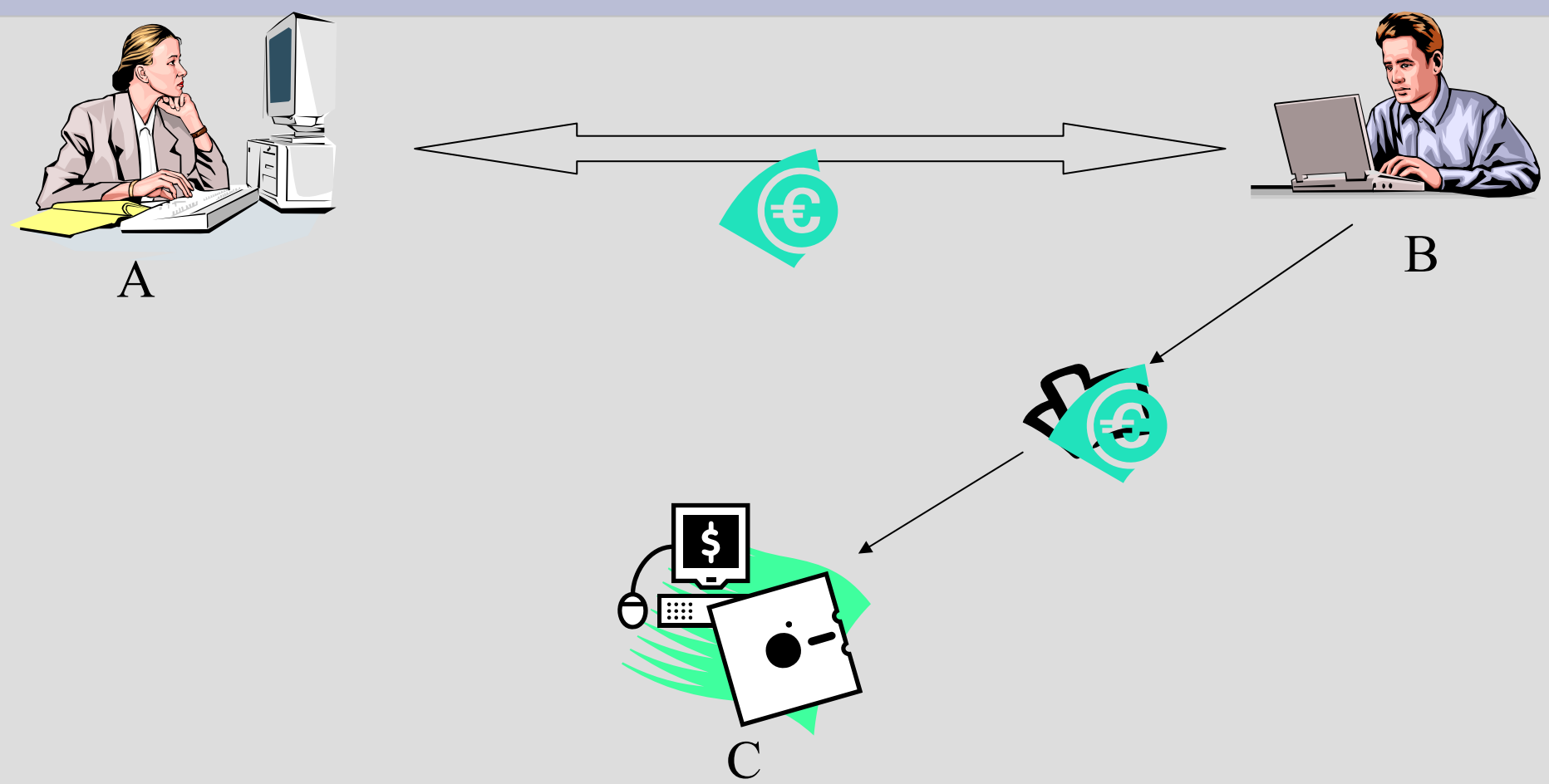
- Supponiamo che il cliente A voglia pagare a B ad esempio $25 da una moneta di $100 e che B voglia poi passare i $25 ad un'altra persona C

1. Il protocollo di pagamento procede come specificato precedentemente
2. Al passo di rilevazione del taglio A invia una certificazione che denota il trasferimento della moneta a B, che poi potrà usare la certificazione con C

Caratteristiche e-cash : cedibilità (2)

3. A e B continuano come da protocollo
4. B manda succesivamente tutta la trascrizione del pagamento tra lui e A a C
5. C controlla la validità della trascrizione inviatagli da B e poi seguono il protocollo

Caratteristiche e-cash : cedibilità (3)



Caratteristiche e-cash : divisibilità

- Ovviamente questa è una delle caratteristiche più importanti che hanno guidato lo sviluppo; l'idea stessa di albero binario per rappresentare la moneta nasce dall'esigenza di dividire la stessa in formati diversi

Attacchi

- Tutti gli attacchi derivati da quelli contro l'RSA risultano validi per questo sistema, anche se Okamoto usa un funzione hash unidirezionale per minimizzare il rischio di attacchi basati su messaggi creati ad hoc

Svantaggi

- Le monete create dal sistema sono concatenabili rendendo possibile risalire all'identità dell'utente
- La divisibilità della moneta è basata sull'uso di complesse funzioni matematiche che porteranno a delle difficoltà nell'implementazione pratica, minando alla base questo sistema

Contanti Elettronici

- Confronto dei vari schemi
- Panoramica sui prodotti

Neri Claudio

Confronto schemi

- *Riguardo alla sicurezza:*
  - Chaum-Fiat-Naor, Ferguson e Okamoto sono sistemi basati su RSA
    - quindi vulnerabili ai possibili attacchi del RSA. La loro sicurezza è basata sulla difficoltà di scomporre i grandi numeri in fattori, mezzi di elaborazione veloci potrebbero violare la sicurezza, anche se ci vuole del tempo quindi dobbiamo aumentare la lunghezza della chiave di questi sistemi.
  - La sicurezza dello schema Brands è basata sulla difficoltà di computare i logaritmi discreti.

    Lo schema Brands sta offrendo una sicurezza migliore.

Confronto schemi

- *Riguardo ai costi di comunicazione:*
  - Lo schema di Chaum-Fiat-Naor consiste di 8 movimenti (4 per il ritiro, 3 per il pagamento e 1 per il deposito).
  - Lo schema di Ferguson 8 movimenti (4 per il ritiro, 3 per il pagamento e 1 per il deposito).
  - Lo schema Brands 10 movimenti (6 per il ritiro, 3 per il pagamento e 1 per il deposito).
  - Lo schema di Okamoto 6 movimenti (2 per il ritiro, 3 per il pagamento e 1 per il deposito).

  Ogni movimento ha chiaramente un costo di tempo per la trasmissioe.

  Lo schema di Okamoto è quello più efficiente in termini di numero di movimenti.

Confronto schemi

- *Riguardo a complessità di esecuzione:*
  - Lo schema di Chaum-Fiat-Naor è il più facile in termini di esecuzione.
  - Lo schema di Okamoto è più complesso poiché usa una combinazione dell'albero binario, il numero intero di William e la radice quadrata che provoca una procedura complessa da capire ed effettuare.

- *Riguardo al problema di double-spending:*

  Tutti gli schemi presentati hanno il problema del *double-spending*.

Confronto schemi

Il risultato del confronto di questi quattro schemi mostra che lo schema Brands risulta essere il migliore poiché offre un elevato livello di sicurezza mantenendo comunque l'anonimato degli utenti.

L'unico svantaggio dello schema Brands è l'alto costo di comunicazione che può tuttavia essere abbassato chiedendo all'utente di ritirare parecchie monete nello stesso momento. Tuttavia, persino il sistema di contanti Brands non è né divisibile né trasferibile anche se offre un' alternativa di divisibilità al costo dell' anonimato.

Confronto schemi

Quindi possiamo dire che neanche questo sistema soddisfa tutte le condizioni di un sistema di contanti digitali ideale. Prima che questi sistemi possano essere realmente utilizzatati da parte dell'utente devono essere risolti vari problemi:

- se una moneta viene persa o rubata chi ci rimette?
- che genere di valuta deve essere attribuita alle monete? I dollari degli STATI UNITI, i contrassegni tedeschi o dovremo avere un nuovo valore di contanti del cyber?
- Come dovremo cambiare i soldi?
- Come dovrebbero essere calcolate le tasse? Su una transazione o sulla creazione di una moneta?

Sommario

Successivamente verranno presentati tre sistemi principale:

- Il DigiCash
- Il CyberCash
- Il First Virtual

Il DigiCash

- Il DigiCash (denaro digitale) è stato ideato da David Chaum nel 1990 ad Amsterdam. Principalmente si occupa di sviluppare i prodotti relativi al pagamento elettronico e ai sistemi di autorizzazione.
- Uno dei prodotti dell'azienda è l'Ecash,
  - prototipo di contante digitali il cui progetto è iniziato nel 1995.
  - progettato per i pagamenti sicuri da effettuare tramite Internet o posta elettronica.
  - realizzato quasi completamente sul sistema digital cash di Chaum, l'unica differenza è che è un sistema in linea

DigiCash: Ecash

- Utilizza un proprio denaro virtuale, rappresentato da monete fornite da banche associate (responsabili della certificazione e dell'autenticazione delle monete virtuali di ecash);
- Per poter utilizzare ecash è necessario aprire un conto con una delle banche partecipante. Sarà possibile memorizzare sul proprio computer monete elettroniche;
- E' necessario procurarsi il software *Ecash client* (reperibile gratuitamente in rete) è in grado di operare (o ricevere) pagamenti con qualsiasi altro utente Internet che stia anch'egli utilizzando tale client.
- Il trasferimento delle monete per mezzo di reti telematiche avviene sfruttando tecniche di crittografia a chiave pubblica e di firma cieca.

DigiCash: Ecash

- Per garantire sicurezza e riservatezza 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.
- 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.

DigiCash: Ecash

- Per quanto riguarda la **riservatezza** delle transazioni, Ecash sfrutta la tecnica della **firma cieca**.
    - Quando il compratore ha necessità di effettuare un pagamento, deve avere ecash sul proprio computer.
    - Il pc dell'utente calcola quante monete sono necessarie per ottenere la somma richiesta e crea le monete assegnando ad ognuna di loro un numero di serie casuale.
    - Spedisce alla banca queste monete, una ad una inserite in una speciale *busta* la quale rappresenta il fattore "cecità ".

Il Cybercash

- Azienda CyberCash è stata fondata nel 1994 da William Melton e Daniel Lynch; (nota come una fra i leader mondiali nella fornitura di supporti al CE)
- Offre un prodotto denominato Wallet ai relativi clienti.
- Sistema basato su firme digitali e sull' algoritmo di crittografia di tipo RSA con una chiave a 1024 bit,
- Il consumatore, cliente dell'azienda, che volesse utilizzare deve scaricare sul proprio sistema il software necessario per il collegamento con la Cybercash.
- Il venditore dovrebbe in primo luogo aprire un conto presso una banca che supporti i pagamenti elettronici con questo prodotto e poi modificare il software del proprio server in modo da poter utilizzare Cybercash.

Il Cybercash

- Nel momento in cui il consumatore effettua il suo acquisto presso il venditore inviandogli l'ordine, questi gli risponderà con il riepilogo degli articoli scelti indicando per ognuno il singolo prezzo, aggiungendo il numero della transazione ed altre informazioni. A questo punto il cliente potrà scegliere l'opzione "PAY" lanciando il *Cybercash wallet* in cui dovrà inserire il numero della sua carta di credito.

- Queste informazioni verranno trasferite in forma cifrata al commerciante, il quale provvederà ad estrarvi l'ordine e ad inoltrare il resto del messaggio alla Cybercash, avendolo prima cifrato con la propria chiave privata. Si noti che il venditore non è in grado di decifrare la parte del messaggio contenente il numero della carta di credito. Ciò garantisce un'ulteriore protezione. La Cybercash provvederà a decifrare e a trasferire alla banca del merchant attraverso linee dedicate non collegate ad Internet. La banca del venditore inoltrerà la richiesta di autorizzazione alla banca del cliente.

- L'esito dell'operazione di verifica della validità degli estremi comunicati sarà notificato alla Cybercash che ne informerà di conseguenza il venditore che poi provvederà a girarla all'acquirente.

- Il sistema di cyberCash offre una più alta sicurezza rispetto alle carte di credito. In questo sistema però né la segretezza dell'acquirente né quella del consumatore sono protette.

Il First Virtual

L'azienda First Virtual è una delle prime aziende in grado di offrire un sistema di trasferimento digitale dei soldi generato per Internet. Il First Virtual è un sistema di pagamento elettronico basato sulla sicuerezza senza crittografia.

Come funziona il protocollo di pagamento?

1) Registrazione del potenziale acquirente

   Prima di tutto il cliente deve registrarsi presso la First Virtual.propri dati personali. In seguito gli viene consegnato dalla First Virtual un numero telefonico che gli permetterà di comunicare anche il proprio numero di conto corrente. In questo caso l'acquirente evita di inviare il numero di conto direttamente su Internet.

   Conclusa la fase di registrazione, viene attribuito al richiedente un *codice identificativo personale* (*PIN*) che servirà per identificarsi al momento del pagamento della merce acquistata sulla Rete.

2) Registrazione del commerciante

   Anche il commerciante deve registrarsi presso la First Virtual, fornendo gli estremi del proprio conto bancario.

Il First Virtual

3) Procedura di acquisto.

L'acquirente accede al sito di commercio elettronico del venditore tramite una password inviatagli in chiaro dalla First Virtual. Seleziona i prodotti desiderati, per i quali gli viene comunicato il relativo importo.

Il cliente invierà poi al venditore il PIN rilasciatogli dalla First Virtual, in modo che questi abbia modo di verificare la validità presso l'istituto stesso. Il messaggio in questione può essere spedito secondo diverse modalità: una delle più semplici è l'e-mail.

4) First Virtual trasmette una richiesta al cliente per una conforma.

5) Il cliente può rispondergli con due risposte differenti:

Si Il cliente autorizza

No Il cliente sta rifiutando di pagare.

Il First Virtual

Se le risposte sono affermative condurranno alla detrazione dell'ammontare corrispondente all'acquisto dal conto corrente, altrimenti nessuna transazione finanziaria andrà in porto.

Qualora invece l'utente comunichi alla First Virtual di non aver mai autorizzato la transazione, questa provvederà ad eseguire una serie di controlli per intercettare l'autore della frode.

Quindi la First Virtual partendo dal presupposto che in una struttura come Internet è difficile evitare in assoluto le frodi, ha cercato di realizzare un sistema che fosse in ogni caso in grado di ridurre il più possibile gli effetti di un furto. Nel caso che fosse rubato il PIN l'intruso non avrebbe modo di concludere la transazione in quanto non riceverebbe lui la richiesta di conferma dell'ordine, ma il reale proprietario del codice. La maggiore attrazione di First Virtual è la sua estrema semplicità ed il fatto che, diversamente da altri approcci utilizzati per il commercio elettronico, non si basa su protocolli crittografici, ma sull'invio di messaggi di conferma al cliente tramite tradizionali messaggi di posta elettronica.

Grazie per l’attenzione