

introduzione a Bitcoin e alle criptovalute digitali

P.Bertoni

4 luglio 2014

Tabella Contenuti

- 1 Introduzione
 - Definizione di una criptovaluta
 - Elementi teorici
- 2 Protocollo
 - Strutture dati
 - Primitive crittografiche
 - Modello formale di sicurezza
- 3 Attacchi tipici
 - Forgiatura
 - Anonimità
 - Double spending

contenuto

- 1 **Introduzione**
 - Definizione di una criptovaluta
 - Elementi teorici
- 2 **Protocollo**
 - Strutture dati
 - Primitive crittografiche
 - Modello formale di sicurezza
- 3 **Attacchi tipici**
 - Forgiatura
 - Anonimità
 - Double spending

Criptovaluta

specifiche sul protocollo

problema

implementare una **valuta** economica

- affidabile
- informatica
- decentralizzata
- distribuita

Criptovaluta

specifiche sulle transazioni

problema

trasmettere **transazioni** di valuta tra enti

- pubbliche
- anonime \Rightarrow tra **indirizzi**, non utenti
- autenticate
- non ripudiabili
- irreversibili

registrate in una sorta di **storico** globale

Criptovaluta

idea astratta di transazione \mathfrak{T}

- atto tra N mittenti e M destinatari
- utenti incoraggiati a usare un indirizzo unico $\forall \mathfrak{T}$
- \sim assegno
“in data t , $\{x\}_1^N$ ha versato tot a $\{y\}_1^M$ che ora ne è proprietario”

gestione del resto

- $y' \in \{y\}$ destinatari, controllato da chi emette \mathfrak{T}
- deframmentare: nuova \mathfrak{T} con molteplici mittenti

Criptovaluta

specifiche sull'affidabilità: **proof of work**

problema

progettare algoritmo di **mining** per convalida 

- *trattabile* da decidere
- *intrattabile* da risolvere
- dipendente da lista transazioni in attesa

motivazione a **partecipare**

- onestà ricompensata
- complessità lavoro onesto \equiv complessità disonesto
- $\Pr[\text{successo disonestà}] \rightarrow 0$

Differenze con valute tradizionali



ontologia

esplicita: unità fisiche

implicita: in funzione di \mathfrak{T}

fiducia nell'accettazione di moneta

difficoltà di contraffazione

possibilità di furto o smarrimento

gettone fisico

chiave privata di firma digitale

fiducia nel protocollo di supporto

ente nazionale o sovranazionale

modello formale di sicurezza

i primordi: eCash [Chaum]

sistema di firma digitale a *conoscenza zero*

Scenario (e.g. voto digitale)

- sia m plaintext, A autore e F firmatario, $A \neq F$
- firma *classica*: $S(m, K_{PR}^A)$
- firma *cieca*: $S^B(f(m), K_{PR}^F)$
- F non può calcolare m data $f(m)$

Algoritmo

- 1 A estrae *nonce* x
- 2 A invia $\bar{m} = f(m, x)$ messaggio *cieco* a F
- 3 F restituisce $s^B = \bar{s} = E(\bar{m}, K_{PR}^F)$
- 4 chiunque conosca K_{PB}^F calcola $\bar{m}^S = E(\bar{s}, K_{PB}^F)$
- 5 “ “ pure x ” $m^S = f^{-1}(\bar{m}^S, x)$

contenuto

- 1 **Introduzione**
 - Definizione di una criptovaluta
 - **Elementi teorici**
- 2 **Protocollo**
 - Strutture dati
 - Primitive crittografiche
 - Modello formale di sicurezza
- 3 **Attacchi tipici**
 - Forgiatura
 - Anonimità
 - Double spending

Curve Ellittiche

- curve definite su un certo \mathbb{F}_q da

$$y^2 = x^3 + ax + b$$

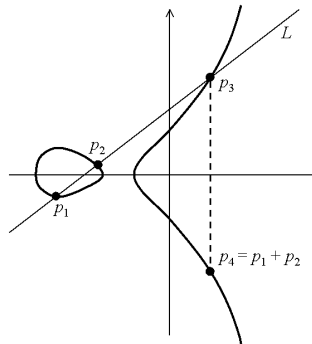
- non singolari, i.e. $4a^3 + 27b^2 \neq 0$

Theorem (Hasse)

sia \mathbb{F}_q il campo di Galois di ordine q
sia $\mathcal{E}_q = \mathcal{E}_{(a,b)}(\mathbb{F}_q)$ una sua curva ellittica

$$|o(\mathcal{E}_q) - (q + 1)| \leq 2\sqrt{q}$$

\Rightarrow **ordine** GF governa *difficoltà*



Curve Ellittiche

legge di gruppo: definizione

$(\mathcal{E}_{(a,b)}(\mathbb{F}_q), +)$ definisce un **gruppo abeliano**

$$R = P + Q \triangleq (x_R, -y_R)$$

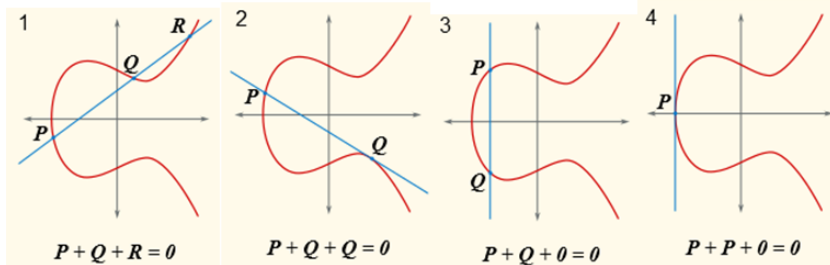
$$x_P \neq x_Q : \begin{cases} y_R \triangleq y_P + s(x_R - y_P) \\ x_R \triangleq s^2 - x_P - x_Q \end{cases} \quad s = \frac{y_P - y_Q}{x_P - x_Q}$$

$$x_P = x_Q : \begin{cases} y_P = -y_Q : & R = O \\ y_P = y_Q \neq 0 : & \begin{cases} y_R \triangleq y_P + s(x_R - y_P) \\ x_R \triangleq s^2 - 2x_P \end{cases} \end{cases} \quad s = \frac{3x_P^2 + a}{2y_P}$$

$$R = P \times n \triangleq P + P + \dots + P \quad n \in \mathbb{Z} \text{ volte}$$

Curve Ellittiche

legge di gruppo: casistica



Curve Ellittiche

problema matematico

trovare un segreto $d \in [1, n-1]$, dati

- $\mathcal{E} = \mathcal{E}_{(a,b)}(\mathbb{F}_q)$
- $G \in \mathcal{E} : \quad \langle G \rangle = \mathcal{E}$
- $n = o(G) : G \times n = O = P_\infty, \quad n \text{ primo}$
- $P \in \mathcal{E}$
- $Q = P \times d$

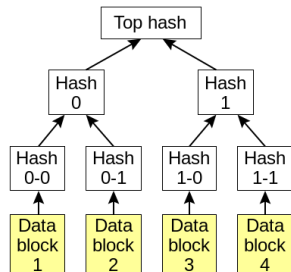
Funzioni di Hash One Way

- $h: \mathbb{Z} \rightarrow \mathbb{Z}_n$ non iniettiva
- resistenza a
 - preimmagine \rightarrow ricerca *bruta* è $O(2^n)$
 - collisioni deboli \rightarrow " " "
 - collisioni forti \rightarrow *birthday*: ricerca *bruta* è $O(2^{n/2}) \ll O(2^n)$
- usate soprattutto per
 - autenticazione
 - integrità
- spesso firmato il **digest** $h(m)$ anzichè m

Alberi di Merkle

usati in protezione **integrità transazioni**

- foglia $\leftrightarrow \mathcal{T}$
- \neg foglia \leftrightarrow hash dei due figli
- decidere se una foglia \in
 - lista: $O(N)$
 - albero: $O(\log_2 N) \ll O(N)$
- Bitcoin: $\text{hash}(n) = \text{SHA}_{256}(\text{SHA}_{256}(\mathcal{L}_n | \mathcal{R}_n))$



contenuto

- 1 Introduzione
 - Definizione di una criptovaluta
 - Elementi teorici
- 2 Protocollo
 - Strutture dati
 - Primitive crittografiche
 - Modello formale di sicurezza
- 3 Attacchi tipici
 - Forgiatura
 - Anonimità
 - Double spending

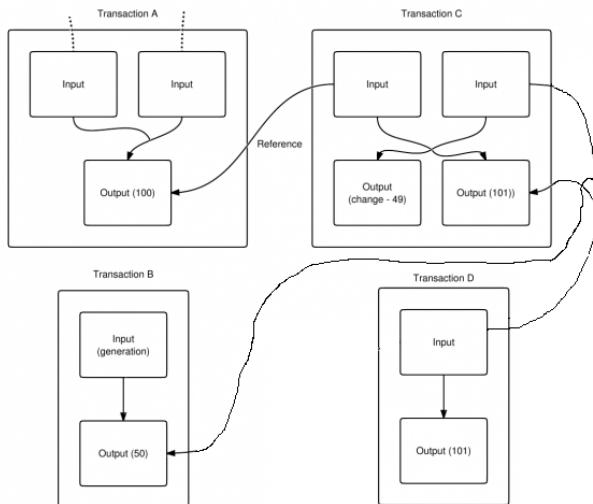
Transazione \mathcal{T}

struttura dati

- $\mathcal{T}^{ID} = \text{hash}(\mathcal{T})$
 - timestamp t
- $\forall i$ indirizzo di input \mathcal{I}_i^{in}
 - ~~somma trasferita \mathcal{B}_i^{in}~~
 - chiave $K_{i,in}^{PB}$
 - indice $p : \mathcal{I}_i^{in} = [\mathcal{I}_p^{out}]_{\mathcal{T}_{-1}}$
 - $\mathcal{T}_{-1}^{ID} = \text{hash}(\mathcal{T}_{-1})$
 - firma $\mathcal{T}^{Si} = E(\tilde{\mathcal{T}}_i, K_{i,in}^{PR})$
 - $\forall j$ indirizzo di output \mathcal{I}_j^{out}
 - somma ricevuta \mathcal{B}_j^{out}
 - $\text{hash}(K_{j,out}^{PB})$

Transazione Σ

collegamento tra transazioni



Blocco \mathfrak{B}

struttura dati

Header

- hash di $\mathfrak{B}_{i-1}^{\mathfrak{H}}$
- MerkleTree di $\{\mathfrak{T}\}_{\mathfrak{B}}$
- timestamp t
- target z
- nonce x
- titolare della *coinbase* \mathfrak{C}

Payload

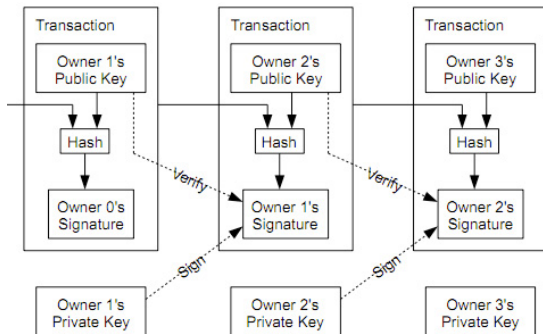
- lista transazioni $\{\mathfrak{T}\}_{\mathfrak{B}}$

- durante *mining* di \mathfrak{B} , campi continuamente modificati
- \mathfrak{B} descrive la propria *proof of work*

Transazioni → Blocchi

generazione nuova transazione

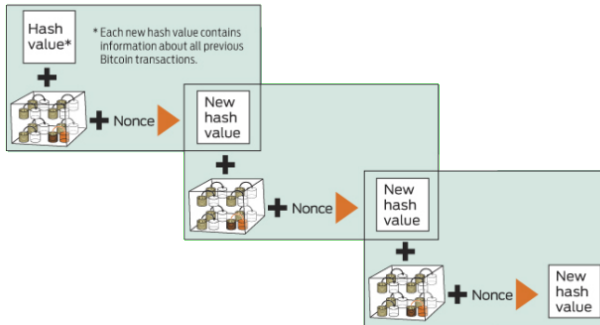
- 1 broadcastata tramite protocollo *flooding*
- 2 ogni miner **può** includerla nel suo *pool*
- 3 inizialmente inserita in un pool come *invalida*
- 4 dopo risoluzione del \mathfrak{B} corrente è rimossa da ogni pool



Blocchi → Blockchain

generazione catena di blocchi

- hash dei blocchi precedenti \sim puntatori di una lista
- a ritroso si giunge al \mathfrak{B} di *genesis*



Motivazione all'utilizzo

fees \mathfrak{F} sulle transazioni

\forall transazione \mathfrak{T}

- $\mathfrak{F}\mathfrak{T} = \sum_i^N \mathfrak{B}_i^{in} - \sum_i^M \mathfrak{B}_i^{out} \geq 0$
- spetta a miners che risolvono $\mathfrak{B} \ni \mathfrak{T}$
- $\left\{ \begin{array}{l} \text{mai obbligatoria, ma...} \\ \text{miners } \textbf{mai} \text{ obbligati ad aggiungere } \mathfrak{T} \text{ al proprio pool} \end{array} \right.$

incentivi per

- velocizzare validazione \mathfrak{T}
- *mining* costante nonostante decrescita *coinbase rewards*

Motivazione all'utilizzo

transazioni Coinbase \mathcal{C}

- $\forall \mathcal{B}, \exists! \mathcal{C}$
- inputs: \emptyset
- outputs: ricompensa a **miners risolutori** di \mathcal{B}
 - *coinbase*
 - 50 $\text{\$}$ iniziali
 - dimezzata ogni 210K blocchi risolti
 - nulla dopo 6.93M blocchi
 - $\{\mathcal{F}\} \in \mathcal{B}$
- inflazione
 - dettata solo da mining
 - limitata

$$\sum_{i=0}^{6.93M-1} \frac{50}{2^{\lfloor i/210K \rfloor}} = 21M \text{\$}$$

contenuto

- 1 Introduzione
 - Definizione di una criptovaluta
 - Elementi teorici
- 2 Protocollo
 - Strutture dati
 - **Primitive crittografiche**
 - Modello formale di sicurezza
- 3 Attacchi tipici
 - Forgiatura
 - Anonimità
 - Double spending

ECDSA

inizializzazione

Alice: scelta dei parametri **pubblici**

- 1 $q = 2^m$
- 2 $(a, b) : 4a^3 + 27b^2 \neq 0$
- 3 $G = (x_G, y_G) \in \mathcal{E}_{(a,b)}(\mathbb{F}_q)$
- 4 $n = o(G)$

Alice: generazione coppia chiavi

- 1 $K^{PR} \triangleq d_A \leftarrow \text{rand} \in [1, n-1]$
- 2 $K^{PB} \triangleq Q_A \leftarrow n \times d_A$

Bob: verifica validità di Q_A ricevuta

- 1 $Q_A \neq O$
- 2 $Q_A \in \mathcal{E}$
- 3 $Q_A \times n = O$

ECDSA

firma digitale

Alice: firma del messaggio m

- 1 $e \leftarrow \text{hash}(m)$
- 2 $k \leftarrow \text{rand} \in [1, n] \subset \mathbb{N}$
- 3 $(x_1, y_1) \leftarrow k \times G$
- 4 $r \leftarrow x_1 \bmod n$
- 5 **if** $r = 0$ **goto** 2
- 6 $s \leftarrow k^{-1}(e + rd) \bmod n$
- 7 **if** $s = 0$ **goto** 2
- 8 **return** (r, s)

ECDSA

verifica firma

Bob: verifica firma (r, s) di m

- 1 $(r, s) \in [1, n-1] \times [1, n-1]$
- 2 $e \leftarrow \text{hash}(m)$
- 3 $w = s^{-1} \pmod n$
- 4 $(u_1, u_2) \leftarrow (ew \pmod n, rw \pmod n)$
- 5 $(x_1, y_1) \leftarrow u_1 \times G + u_2 \times Q$
- 6 **ok** $\Leftrightarrow r \equiv x_1 \pmod n$

SHA-256

Secure Hash Algorithm 256 bit

- $2^{256}/2 = 128$ bit di sicurezza collisioni non ancora trovate
- $\text{len}(m) < 2^{64}$, $\text{len}(n) = 256$

1 padding

- a) $\text{len}(m|0\dots) \equiv 448 \pmod{512}$
- b) 64 bit di $\text{len}(m)$

2 N blocchi da 512 bit: $B^{(1)}, \dots B^{(N)}$

3

$$H^{(i)} \triangleq H^{(i-1)} + C_{B^{(i)}}(H^{(i-1)})$$

4 ritorna $d = H^{(N)}$

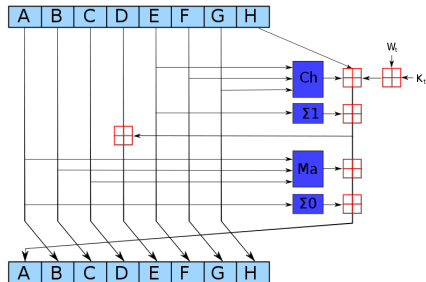


Figura: funzione di compressione C

contenuto

- 1 Introduzione
 - Definizione di una criptovaluta
 - Elementi teorici
- 2 Protocollo
 - Strutture dati
 - Primitive crittografiche
 - **Modello formale di sicurezza**
- 3 Attacchi tipici
 - Forgiatura
 - Anonimità
 - Double spending

Firma digitale

- $|K_{ECC}| = 256 \text{ bit} \simeq |K_{RSA}| = 3072 \text{ bit}$
- $q \simeq 10^{77}$, $n \simeq 10^{69}$, $\mathcal{E} : y^2 = x^3 + 0x + 7$
- **meet in the middle** [Shank]: $\Omega(\sqrt{q})$
- nonce k è confidenziale: $d = r^{-1}(ks - e)$
- **replay attack**: nonce deve tale
 - a) $r_1 = r_2 = r$
 - b) $s_1 \equiv k^{-1}(e_1 + dr) \pmod{n}$, $s_2 \equiv k^{-1}(e_2 + dr) \pmod{n}$
 - c) $k(s_1 - s_2) \equiv (e_1 - e_2) \pmod{n}$
 - d) $m_1 \neq m_2 \Rightarrow (s_1 - s_2) \neq 0 \Rightarrow k \equiv (s_1 - s_2)^{-1}(e_1 - e_2) \pmod{n}$

Proof of Work

metodo Hashcash

- *facile* verificare che il messaggio è soluzione di problema *difficile*
- **brute force** unica tecnica risolutiva
- Problema: dati $h : \mathbb{Z} \rightarrow \mathbb{Z}_n$, m , $z \leq n$, trovare nonce x :

$$d = \text{hash}^{\text{PoW}}(m|x) < T_z = 2^{n-z+1}$$

i.e. **digest ha z zeri non significativi** (parametro **target**)

$$\Pr[d < T_z | Z = z] = \frac{1}{2^z} \Rightarrow O(2^z)$$

- problema risolto \Leftrightarrow blocco \mathfrak{B} risolto $\Leftrightarrow \{\mathfrak{T}\}_{\mathfrak{B}}$ convalidate
- Bitcoin: $\text{hash}^{\text{PoW}}(\mathfrak{B}) = \text{SHA}_{256}(\text{SHA}_{256}([\text{Header}]_{\mathfrak{B}}))$

Proof of Work

esempli gratia: $z = 15$

hash("hello world"|001) = 9002381300129484192947128

⋮

hash("hello world"|034) = 0000834716283947104512438

⋮

hash("hello world"|415) = 000000000000000000000083201

n.b. *gambler's fallacy*

$$\forall t_1, t_2 \quad \Pr(Z = z, T = t_1) = \Pr(Z = z, T = t_2)$$

Proof of Work

adattamento target

target z_i ricalcolato ogni 2016 blocchi risolti ~ 2 settimane

1 $\Delta_i^t \leftarrow t_i - t_{i-1}$

2 $\Delta_i^t \leftarrow \text{clip}(\Delta_i^t, 0.5, 8)$

3 $z_{i+1} \leftarrow z_i \frac{\Delta_i^t}{2}$

- $z \propto \Delta^t \Rightarrow$ soluzioni veloci abbassano target
i.e. generazione problemi più difficili, vv.
- blocco risolto mediamente ogni 10 minuti

Proof of Work

sicurezza di SHA256

in teoria. . .

- **preimage** attack: $O(2^{256})$
- **birthday** attack: $O(2^{256/2})$

. . . in pratica

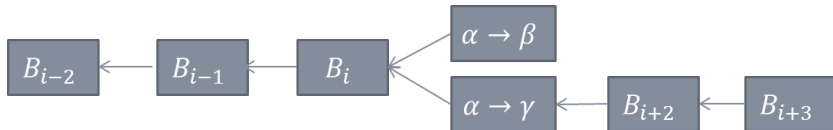
| <i>Metodo</i> | <i>Attacco</i> | <i>Iterazioni</i> | <i>Complessità</i> |
|--------------------|-------------------|-------------------|--------------------|
| deterministico | collisione | 24 | $2^{28.5}$ |
| meet in the middle | preimmagine | 42 | $2^{248.4}$ |
| differenziale | pseudo collisione | 46 | 2^{178} |
| biclique | preimmagine | 45 | $2^{255.5}$ |

Proof of Work

prevenzione double spending

spendere due volte la stessa \mathcal{T}

- 1 modifica $\mathcal{T}^{out} \Rightarrow \mathcal{T}$ stessa $\Rightarrow \mathcal{B}$ di appartenenza
ricalcolo nonce \times
- 2 modifica $\mathcal{B} \Rightarrow$ modifica N blocchi successivi
ricalcolo N nonces \Rightarrow risolvere N problemi esponenziali



forking

- *policy*: aggiungere sempre a ramo più lungo
- in una *web of trust* sopravviverà il ramo corretto

Web of trust

dato un pool di miners \mathfrak{M} con capacità di calcolo $\mathcal{C}_{\mathfrak{M}}$ [GH/s]

$$\Pr[\mathfrak{M} \text{ risolve blocco}] \propto \frac{\mathcal{C}_{\mathfrak{M}}}{\mathcal{C}_{\Omega}}$$

\Rightarrow condizione necessaria Bitcoin: $\mathcal{C}_{\text{fair}} \geq 50\% \mathcal{C}_{\Omega}$

se $\exists \mathfrak{M}_{\text{unfair}}$ pool disonesto t.c. $\mathcal{C}_{\mathfrak{M}_{\text{unfair}}} \geq 50\% \mathcal{C}_{\Omega}$

\Rightarrow catena più lunga comanda \Rightarrow crollo fiducia \Rightarrow crollo valore

- no motivazione diretta di lucro
- ma problema irrisolvibile perchè sistematico

contenuto

- 1 Introduzione
 - Definizione di una criptovaluta
 - Elementi teorici
- 2 Protocollo
 - Strutture dati
 - Primitive crittografiche
 - Modello formale di sicurezza
- 3 Attacchi tipici
 - **Forgiatura**
 - Anonimità
 - Double spending

forgiatura

unica possibilità: rubarli a qualcuno

- problema di forgiare dal nulla **non ha senso**
- conoscere K_{PR} \Rightarrow rompere ECDSA

se **quantum computers** implementati

- rottura ECDSA *può diventare facile*
- collisione SHA_{256} resta *difficile*
 - indirizzi $\mathfrak{J} = \text{SHA}_{256}(\text{SHA}_{256}(K_{PB}))$
 - ottenere K_{PB} dal solo \mathfrak{J} è *difficile*
 - ma se è nota sistema rotto

caso di studio: MtGoX [2014]

prima del crollo

web service di [exchange](#)

- scambio valute *fiat* con ₿
- 2013: Blockchain fork in rami con regole diverse
⇒ *MtGox* sospende transazioni

malleabilità

- $\mathcal{T}^{ID} \triangleq h(\mathcal{T})$, $\mathcal{T}^S \in \mathcal{T} \Rightarrow \mathcal{T}^{ID}$ dipende dalla sua firma
- codice *MtGox* accetta firme malformate
- problema \in implementazione, \notin protocollo ₿
- \exists altri exchanges più rigorosi \Rightarrow immunità

caso di studio: MtGoX [2014]

il crollo: attacco delle transazioni *mutanti*

frode di Eve ai danni di MtGox

- 1 M invia \mathcal{T} a E come prelievo
- 2 E ritocca \mathcal{T}^S prima della sua conferma
- 3 ora $\exists \tilde{\mathcal{T}} \equiv \mathcal{T}$, ma $h(\tilde{\mathcal{T}}) \neq h(\mathcal{T}) \Rightarrow \tilde{\mathcal{T}}^{ID} \neq \mathcal{T}^{ID}$
- 4 $\tilde{\mathcal{T}}$ diffusa da E , confermata prima di \mathcal{T}
- 5 \mathcal{T} non confermata perchè $\tilde{\mathcal{T}}$ invalido
- 6 E fa complain a M per \mathcal{T} non ricevuta
- 7 M controlla il suo storico: \mathcal{T} non è stata accettata
- 8 M costretto a inviare rimborso \mathcal{T}'

DDoS

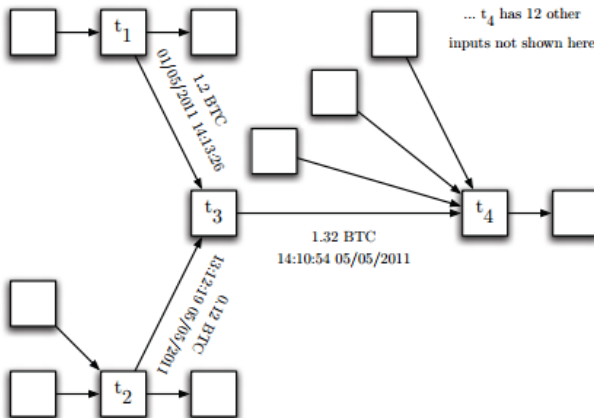
- grande sovraccarico, persino se c'è immunità
- latenza nelle risposte \Rightarrow incertezza \Rightarrow speculazione

contenuto

- 1 Introduzione
 - Definizione di una criptovaluta
 - Elementi teorici
- 2 Protocollo
 - Strutture dati
 - Primitive crittografiche
 - Modello formale di sicurezza
- 3 Attacchi tipici
 - Forgiatura
 - **Anonimità**
 - Double spending

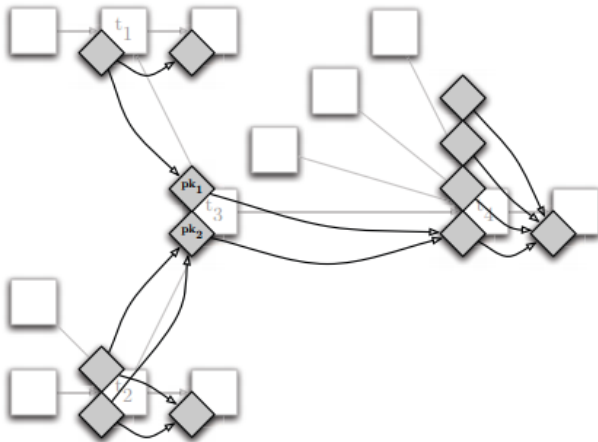
caso di studio: Reid [2011]

rete transazioni \mathcal{T}



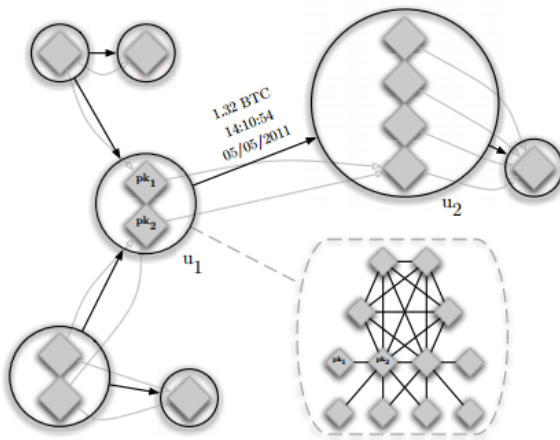
caso di studio: Reid [2011]

rete utenti imperfetta $\tilde{\mathcal{U}}$



caso di studio: Reid [2011]

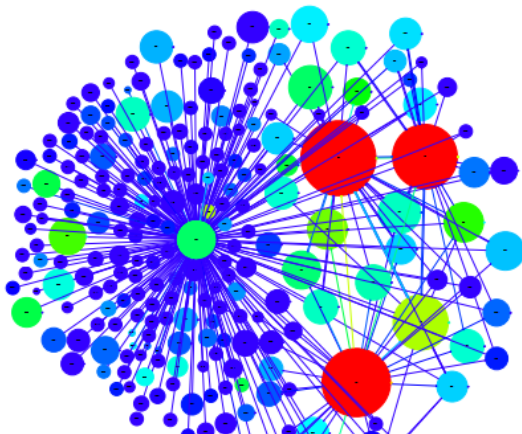
rete utenti \mathcal{U} , rete ancilla \mathcal{A}



caso di studio: Reid [2011]

integrazione con informazioni esterne

- dimensione $\propto |\{K_{PB}\}|$ utente = # transazioni
- colore $\propto \text{฿ scambiati}$



contenuto

- 1 Introduzione
 - Definizione di una criptovaluta
 - Elementi teorici
- 2 Protocollo
 - Strutture dati
 - Primitive crittografiche
 - Modello formale di sicurezza
- 3 Attacchi tipici
 - Forgiatura
 - Anonimità
 - Double spending

caso di studio: Karame [2012]

tipologia transazione

- lenta, *e.g.* acquisto ticket eventi
sicurezza offerta dal mining
- **veloce**, *e.g.* pagamento in negozio
 - ∃ possibilità di *double spending*
 - tempi scambio [s] \ll tempi validazione [*min*]
 - Bitcoin segue tecnica *struzzo*
 - problema non grave ma aperto

caso di studio: Karame [2012]

ipotesi

hosts

- A peer disonesto
- H complici di A
- V vendor onesto

transazioni

- \mathcal{T}_V : acquisto regolare
- \mathcal{T}_A : recupero fraudolento

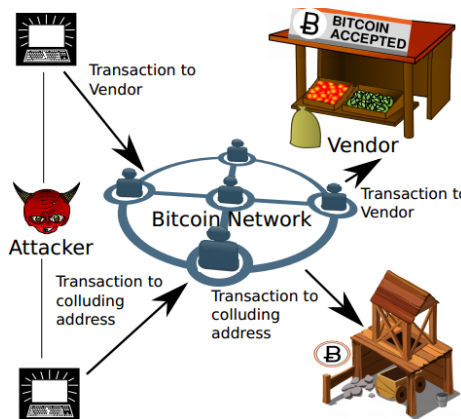
ipotesi

- A conosce indirizzo IP di V
- \mathcal{C}_A trascurabile
- $\mathcal{T}_V^{in} = \mathcal{T}_A^{in} \in A$
- $V \ni \mathcal{T}_V^{out} \neq \mathcal{T}_A^{out} \in A$
- implementazioni *plain vanilla*

caso di studio: Karame [2012]

idea di massima

- $\mathcal{T}_V, \mathcal{T}_A$ inviate contemporaneamente
⇒ incluse nello stesso pool
- se $\mathcal{T}_A^{in} = \mathcal{T}_V^{in}$
⇒ non ammesse nello stesso pool
- inclusa solo la prima \mathcal{T} ad arrivare
⇒
 - \mathcal{T}_A da validare rapidamente
 - \mathcal{T}_V sarà smentita dalla rete



caso di studio: Karame [2012]

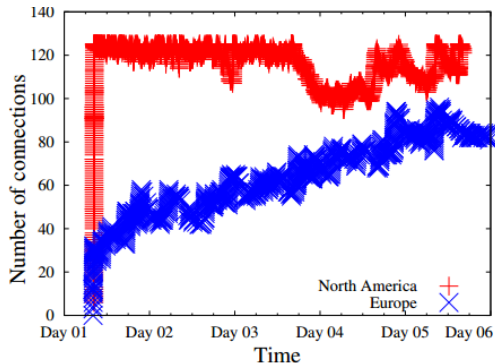
1^a condizione: connessione diretta tra A e V

V riceve prima \mathcal{T}_V di \mathcal{T}_A

oppure V includerebbe prima \mathcal{T}_A nel suo pool

- client accetta sempre nuove connessioni < 125 max
- A comunica con H
 - senza latenza
 - privatamente
- H non comunica con V
- A invia
 - 1 \mathcal{T}_V a V
 - 2 \mathcal{T}_A a H

$$\Rightarrow t_V^V < t_V^A$$



caso di studio: Karame [2012]

2^a condizione: diffusione manipolata

\mathcal{T}_A confermata in *blockchain* prima di \mathcal{T}_V

oppure \mathcal{T}_A non più validabile

- ogni peer include $\mathcal{T}_A \dot{\vee} \mathcal{T}_V$ in proprio pool
 - $\mathcal{T}_A, \mathcal{T}_V$ broadcastate in due partizioni
 - termine quando $\mathcal{T}_A \dot{\vee} \mathcal{T}_V$ confermata
- $\Pr[\tau_A < \tau_V] \propto \eta_A / \eta_V$ migliora se
 - invio di \mathcal{T}_A precede invio di \mathcal{T}_V
 - H aiutano A diffondendo \mathcal{T}_A e filtrando \mathcal{T}_V

caso di studio: Karame [2012]

probabilità di successo

$\Pr[\text{successo in tempo } \delta t] \sim \text{Bernoulli}(\eta_A, p)$

- $\eta_A = \#$ peers coinvolti
- $p = \Pr[\text{peer generi } \mathcal{B} \text{ in } \delta t]$

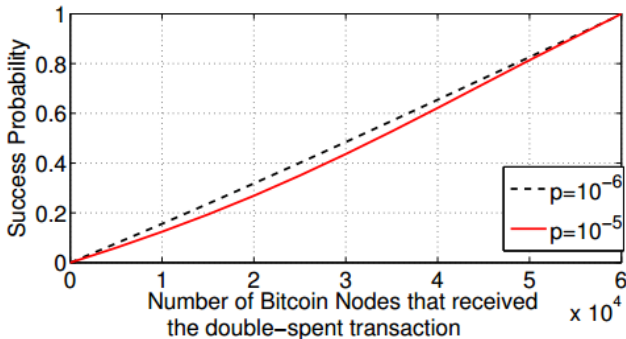


Figura: $\Pr[\text{successo} \mid \delta t = 10s, \eta = 6 \cdot 10^4]$