Hashing, teoria dei codici, numeri primi e sequenze generali di Cassini

Fabio Caldarola

Dipartimento di Matematica e Informatica

Università della Calabria, Rende (CS)

6th **Crati Valley Workshop on Blockchain**

INTERNET OF THINGS AND BLOCKCHAIN: ANALISI INTERDISCIPLINARE

CENTRO CONGRESSI AULA MAGNA - UNIVERSITÀ DELLA CALABRIA 

28 marzo 2018

Blockchain Governance

BLOCKCHAINLAB

CUP J48C17000230006 - Progetto di Ricerca & Sviluppo **IoT&B - Internet Of Things (IoT) e Blockchain: ricerca, design e sviluppo di soluzioni per il mercato**

UNIONE EUROPEA

REGIONE CALABRIA

POR Calabria 2014-2020

POR CALABRIA FESR-FSE 2014-2020

ASSE I – PROMOZIONE DELLA RICERCA E DELL'INNOVAZIONE

Obiettivo specifico 1.2 "Rafforzamento del sistema innovativo regionale e nazionale"

Azione 1.2.2 "Supporto alla realizzazione di progetti complessi di attività di ricerca e sviluppo su poche aree tematiche di rilievo e all'applicazione di soluzioni tecnologiche funzionali alla realizzazione delle strategie di S3"

Sommario

- INTRODUZIONE. Bitcoin e blockchain.

1. PARTE 1: Crittografia, numeri primi, teoria dei numeri.
2. PARTE 2: Funzioni hash e crittografia.
3. PARTE 3: Un ultimo esempio: il Primecoin.

INTRODUZIONE - Bitcoin e blockchain

- Prima dell'introduzione del sistema Bitcoin (BTC), c'erano stati diversi tentativi di creare una valuta digitale sicura, che tutelasse l'anonimato dei pagamenti e fosse in grado di operare senza un'autorità centrale (e.g. "b-money", "bit-gold", e tanti altri).
- Il bitcoin nasce a fine 2008 con il seguente paper pubblicato online

  SATOSHI NAKAMOTO, *Bitcoin: A Peer-to-peer Electronic Cash System*, www.bitcoin.org

  che illustra funzionamento e caratteristiche principali della **criptovaluta**.
- Il software per la creazione dei bitcoin viene introdotto l'anno successivo (gennaio 2009), ma per i primi due anni i bitcoin rimangon8 poco più che una curiosità tecnologica.
- Il primo pagamento viene effettuato il 22 maggio 2010: 10.000 BTC per due pizze (il valore stimato di un BTC era inferiore ad un centesimo di dollaro, mentre oraq 7.900 $ circa). Oggi, sul web, viene festeggiato (con ironia) il "Bitcoin Pizza Day"!

INTRODUZIONE - Bitcoin e blockchain

- La crazione di nuovi bitcoin avviene attraverso la procedura di *mining*: chi vuole può mettere a disposizione la potenza computazionale dei propri dispositivi per regolare le transazioni in bitcoin e divenire un *miner*, un nodo del sistema BTC.
- I computer impiegati dai miners risolvono i cosiddetti *proof-of-work*, “puzzle crittografici” molto complicati che garantiscono la sicurezza delle transazioni e creano nuovi bitcoin per ricompensare chi ha impiegato i propri mezzi.
- Maggiore è il numero di utenti coinvolti nel mining, maggiore è la complessita dei puzzle crittografici da risolvere.
- In seguito al successo di Bitcoin, sono state create un migliaio di criptovalute alternative (ad esempio, tra le più conosciute ci sono Litecoin, GeistGeld, SolidCoin, BBQcoin, Feathercoin, etc.)

INTRODUZIONE - Bitcoin e blockchain

- Molte criptomonete sono versione implementate o perfezionate, altre semplici copie dei bitcoin e del sistema **blockchain** originali.
- Altre criptovalute come Peercoin e Novacoin propongono invece un approccio diverso, sostituendo il sistema *proof-to-stake* come alternativa al *proof-of-work*.
- Nel *proof-to-stake*, quando una nuova unità della criptovaluta viene creata con successo, anzicchè premiare chi ha risolto il puzzle viene distribuito una sorta di dividendo a tutte le unità in circolazione, premiando chi ne ha di più.
- In stretta relazione a Peercoin c'è un'altra criptomoneta (per noi) interssante: Primecoin. Essa sfrutta sequenze di numeri primi.

INTRODUZIONE - Bitcoin e blockchain

- "*Blockchain*" vuol dire letteralmente "catena di blocchi" ed è un elenco in continua espansione di record (o strutture) chiamati *blocks*, che sono collegati tra loro e resi sicuri mediante l'uso della **crittografia**.
- Una blockchain funge sostanzialnente da registro aperto e distribuito (i.e. un enorme libro mastro) che può registrare le transazioni tra due parti in modo sicuro, verificabile e permanente. Per questo utilizzo, questo database sfrutta una rete peer-to-peer che si collega ad un protocollo per la convalida dei nuovi *blocks* he si vanno ad aggiungere continuamente alla *chain*.
- Una volta registrati, i dati in un blocco non possono essere retroattivamente alterati senza che vengano modificati tutti i blocchi successivi ad esso, e questo implica la collusione della maggioranza della rete.
- I file elettronici sono infinitamente riproducibili, ma con un tale libro mastro si esclude il problema del *double-spending*: nessuno può inviare bitcoin che non possiede.

PARTE 1 - Crittografia, numeri primi, teoria dei numeri

La *crittografia* (*kryptós graphía* = “scrittura nascosta”), la necessità di nascondere messaggi strategici da occhi nemici ha origini antichissime. Ci sono tracce di antichi cifrari presso gli Ebrei, gli Spartani e molti altri popoli.

- Un esempio famoso è il *Cifrario di Cesare*, che consisteva nel sostituire ad ogni lettera la terza lettera che la segue nell’alfabeto.
- Per metodi così arcaici sono state presto trovate soluzioni e metodi di decifratura generali (e.g. l’*analisi delle frequenze* è stato uno dei primi che hanno portato alla nascita della **crittoanalisi**).
- Crittografia + crittoanalisi = *crittologia*.
- Nell’era di internet la crittografia svolge un ruolo fondamentale. Le moderne metodologie sono a volte molto complesse e sofisticate e spesso si basano su una branca della matematica chiamata ***teoria dei numeri***.

Crittografia simmetrica e asimmetrica

La **crittografia simmetrica**, fino a pochi anni fa, era l'unico metodo crittografico esistente.

- Mittente e destinatario condividono la stessa chiave.
- A volte il metodo è molto sicuro, ma il problema è condividere la chiave di cifratura con il destinatario senza che questa venisse scoperta.

La **crittografia asimmetrica** è la vera novità del secolo scorso (W. Diffie e M.E. Hellman, Stanford, 1976).

- Ci sono due chiavi: una *chiave pubblica* per crittografare e una *chiave privata* per decifrare.
- Il mittente M invia il messaggio al destinatario D, cifrandolo con la chiave pubblica di D. Tutti vedono passare il messaggio ma solo con la chiave privata di D lo si può decriptare.
- Non c'è nessun bisogno di condividere chiavi segrete.
- Tutti invece possono conoscere le chiavi pubbliche di ciascuno.

L'algoritmo RSA

La sigla RSA indica un algoritmo di crittografia asimmetrica, inventato nel 1977 da Ronald Rivest, Adi Shamir e Leonard Adleman al MIT.

- Le due chiavi sono denominate "diretta" e "inversa".
- Le chiavi dirette sono rese pubbliche, venendo a creare una sorta di "elenco telefonico" a disposizione di tutti gli utenti.
- L'RSA è alla base dei sistemi di firma digitale. Se Alice usa la sua chiave privata per criptare un messaggio e Bob effettua la verifica decriptandolo con la chiave pubblica di Alice, allora Alice è sicuramente colei che ha crittografato il messaggio e non può ripudiure l'atto.
- Con tali metodi di cifratura è anche possibile garantire la provenienza di un messaggio:
  – Alice prima cifra il messaggio con la propria chiave inversa, poi con quella pubblica di Bob;
  – Bob lo decifra con la sua chiave privata e ottiene ancora un messaggio crittografato che necessiterà della chiave pubblica di Alice per essere decifrato, garantendo in questo modo che il messaggio è stato spedito soltanto da Alice.

Esempio: come funziona l'algoritmo RSA

RSA è uno degli esempi più semplici, ma importanti, di applicazione della matematica, e in particolare della teoria dei numeri, alla criptografia. Esso si basa sull'elevata complessità computazionale della fattorizzazione in numeri primi. Ecco il suo funzionamento base:

- si scelgono a caso due numeri primi, p e q abbastanza grandi (e.g. il più grande numero RSA, RSA-2048, utilizza due numeri primi lunghi più di 300 cifre);
- si calcolano il loro prodotto $n = pq$, chiamato *modulo* (poiché l'aritmetica che segue è modulo n), e il prodotto $\varphi(n) = (p-1)(q-1)$;
- mantenendo la fattorizzazione di n segreta (cioè come prodotto di p e q), si sceglie un numero $e < \varphi(n)$ (detto *esponente pubblico*) coprimo con $\varphi(n)$;
- si calcola il numero d (chiamato *esponente privato*) tale che

$$e \cdot d \equiv 1 \pmod{\varphi(n)};$$

- la chiave pubblica è (n, e) mentre la chiave privata è (n, d).

Esempio: come funziona l'algoritmo RSA

Funzionamento:

- Un messaggio m viene cifrato attraverso l'operazione $m^e \pmod{n}$ trasformandolo nel messaggio cifrato c.
- Una volta trasmesso, c viene decifrato con operazione $c^d \pmod{n} = m$, riottenendo il messaggio in chiaro m.

Note.

(i) Il procedimento funziona solo se la chiave e utilizzata per cifrare e la chiave d utilizzata per decifrare sono legate tra loro dalla relazione

$$e \cdot d \equiv 1 \pmod{\varphi(n)},$$

quindi un messaggio cifrato con una delle due chiavi può essere decifrato solo utilizzando l'altra.

(ii) La forza dell'algoritmo sta nel fatto che per calcolare d da e (o viceversa) non basta la conoscenza di n ma serve il numero $\varphi(n) = (p-1)(q-1)$, e il suo calcolo richiede tempi elevatissimi risultando impraticabile.

Esempio: come funziona l'algoritmo RSA

(iii) RSA, insieme a DSA (*Digital Signature Algorithm*), è oggi uno degli algoritmi più usati per la cifratura di firme digitali.

(iv) RSA è molto impegnativo da un punto di vista computazionale, quindi, per un uso efficiente, spesso si sfrutta RSA per codificare un unico messaggio contenente una chiave segreta;

tale chiave verrà poi utilizzata per scambiarsi messaggi tramite un algoritmo a chiave simmetrica, come AES (*Advanced Encryption Standard*, meglio conosciuto come *Rijndael*, dal nome dei due crittografi belgi, Joan Daemen e Vincent Rijmen, che lo hanno sviluppato).

PARTE 2 - Funzioni hash e crittografia

L'*hash* è una funzione che mappa stringhe in strighe.
Più precisamente, è una funzione non invertibile che mappa una stringa di lunghezza arbitraria (i.e. variabile da stringa a stringa) in una stringa di lunghezza minore (fissata).

Esistono numerosi algoritmi che realizzano funzioni hash, con diverse proprietà dipendenti dall'applicazione.

Nelle applicazioni crittografiche solitamente si chiede che la funzione hash abbia le seguenti proprietà:

- Resistenza alla preimmagine, ovvero sia computazionalmente non praticabile la ricerca di una stringa in input che dia un hash uguale a un dato hash;

Funzioni hash

- Resistenza alla seconda preimmagine, ovvero sia computazionalmente intrattabile la ricerca di una stringa in input che dia un hash uguale a quello di una data stringa;
- Resistenza alle collisioni, ovvero sia computazionalmente non praticabile la ricerca di una coppia di stringhe in input che diano lo stesso hash.

Quindi, un algoritmo di hash restituisce una stringa di numeri e lettere a partire da un qualsiasi flusso di bit di qualsiasi dimensione. L'output è detto *digest*.

Per quanto riguarda la lunghezza dei valori di hash, essa varia a seconda degli algoritmi. I valori più usati sono 128 e 256 bit.

Funzioni hash e crittografia

- Le funzioni hash hanno un ruolo molto importante nella crittografia perchè sono utili per verificare l'integrità di un messaggio.
- Infatti, dal momento che l'esecuzione dell'algoritmo di hash su un testo anche minimamente modificato fornisce un message digest completamente diverso rispetto a quello calcolato sul testo originale, allora ogni tentativo di modifica viene subito palesato.
- Le funzioni di hash possono essere anche utilizzate per la creazione di firme digitali, in quanto permettono la rapida creazione della firma anche per file di grosse dimensioni.
  E' infatti conveniente eseguire con rapiditè un hashing del testo da firmare, e poi autenticare solo quello, invece del testo originale.
- In questo modo si evita l'esecuzione dei complessi algoritmi di crittografia asimmetrica su file molto grandi.

Hash, protezione dagli errori, blockchain

- Un altro uso naturale delle funzioni hash è quello indirizzato al rilevamento degli errori nelle trasmissioni.
- La funzione hash viene calcolata dal mittente a partire dai dati e il suo valore è inviato insieme ai dati stessi.
  Il ricevente ricalcola di nuovo la funzione hash, e se i valori hash non corrispondono, significa che è avvenuto un errore durante la trasmissione.
- Questo metodo consente un controllo dell'integrità dei dati di gran lunga migliore della tradizionale *checksum* (e.g. la somma dei bit del messaggio è uno dei tipi più semplici di checksum, "somma di controllo").
- Nella blockchain, ad esempio, ogni blocco include l'hash del blocco precedente creando così il collegamento tra due blocchi. L'iterazione di questa procedura forma la catena della blockchain e garantisce l'integrità del blocco precedente, questo fino al blocco di genesi.

PARTE 3 - Un ultimo esempio di criptovaluta & matematica: il Primecoin.

- *Primecoin* (sign: Ψ; code: XPM) è una criptomoneta strettamente correlata a *Peercoin*.
- La particolarità di Primecoin, interessante per noi, è che il suo sistema di proof-of-work ricerca catene di numeri primi e, in particolare le cosidette *catene di Cunningham* del *primo* e del *secondo tipo*.

Definition

(i) *Una* catena di Cunningham del primo tipo *di lunghezza n è una sequenza di primi* $(p_1, \ldots, p_n)$ *tali che per ogni* $1 \leq i < n$ *si ha* $p_{i+1} = 2p_i + 1$.

(ii) *Similmente, una* catena di Cunningham del secondo tipo *di lunghezza n è una sequenza di primi* $(p_1, \ldots, p_n)$ *tali che per ogni* $1 \leq i < n$ *si ha* $p_i + 1 = 2p_i - 1$.

THANKS FOR YOUR ATTENTION!