

Progetto #3

Tempo e crittografia

- Apporre una marca temporale su di un documento **D** per rispondere alla domanda: “Quando è stato creato **D** ?”
 - Digital timestamping
- Inviare informazioni nel futuro
 - Timed-release crypto
 - Capsula del tempo digitale
- Il progetto verterà su **Digital Timestamping**

Marca Temporale

- La *marca temporale* di un documento è qualcosa che viene aggiunta/associata al documento
 - Per un documento digitale è una stringa di bit ...
- La marca temporale prova che il documento *esiste* nel momento in cui è stata apposta la marca

Digital Timestamping

- Servizio che permette di associare data e ora certe e legalmente valide ad un documento informatico
- Consente di associare al documento una validazione temporale opponibile a terzi
 - Art. 20, comma 3 Codice dell'Amministrazione Digitale Dlgs 82/2005
- Le marche temporali emesse devono essere conservate in appositi archivi per un periodo non inferiore a 20 anni
 - Art. 49 del Dpcm del 30/03/2009

Servizi a pagamento

- Ad esempio, con Aruba
 - Pacchetti da 50, 100, 250, 500 marche temporali
 - Pacchetto da 500: 0,18€ + iva per marca
 - Pacchetto da 50: 0,25€ + iva per marca

Marcatura temporale

- La *marca temporale* viene apposta da un notaio depositando il documento presso il notaio stesso
- Inviare il documento a se stesso tramite un servizio postale (e.g., raccomandata, corriere), ma non aprire la busta
- Nel caso di una marcatura temporale di un'invenzione si può depositare un brevetto
- Si potrebbe pubblicare il documento su di un giornale
- Uso di un registro di protocollo

Due situazioni differenti


- Si appone una *marca temporale* su
 - un documento che è stato appena prodotto con il tempo e data attuale
 - un documento che è stato prodotto nel passato con il tempo e la data in cui è stato prodotto

Scenario di lavoro

- Consideriamo solo la prima situazione
 - *Marcatura con data attuale*
- È facile provare che un documento è stato prodotto dopo di una data fissata
- È difficile provare che un documento è stato prodotto prima di una data fissata

Standard RFC 3161

<https://datatracker.ietf.org/doc/rfc3161/>

- Una marca temporale fidata è un timestamp emesso da un terza parte fidata (TTP) che agisce in qualità di Autorità di TimeStamp (TSA) 
- È usata per provare l'esistenza di un determinato dato prima di un determinato punto nel tempo senza la possibilità per il possessore di retrodatare la marca temporale
- Può essere essere usato un insieme di TSA per incrementare l'affidabilità e ridurre la vulnerabilità
- ANSI ASC X9.95-2016 evoluzione RFC 3161

ANSI ASC X9.95-2016

- Evoluzione RFC 3161
- RFC 3161
 - Basata solo su PKI (firma digitale)
- ANSI ASC X9.95
 - I timestamp generati sono collegati ad altri timestamp
 - Si usa una chiave di firma differente per ogni timeframe, alla scadenza del timeframe la chiave segreta è cancellata

Possibili Soluzioni

TTP: Trusted Third Party

- Due famiglie di protocolli
 - Protocolli distribuiti (senza TSA)
 - Protocolli centralizzati con *collegamenti* (con TSA)
- In entrambi i casi si marca il valore hash del documento in esame per preservare la *confidenzialità* senza perdere in sicurezza
 - La confidenzialità dovrebbe essere garantita in maniera differente

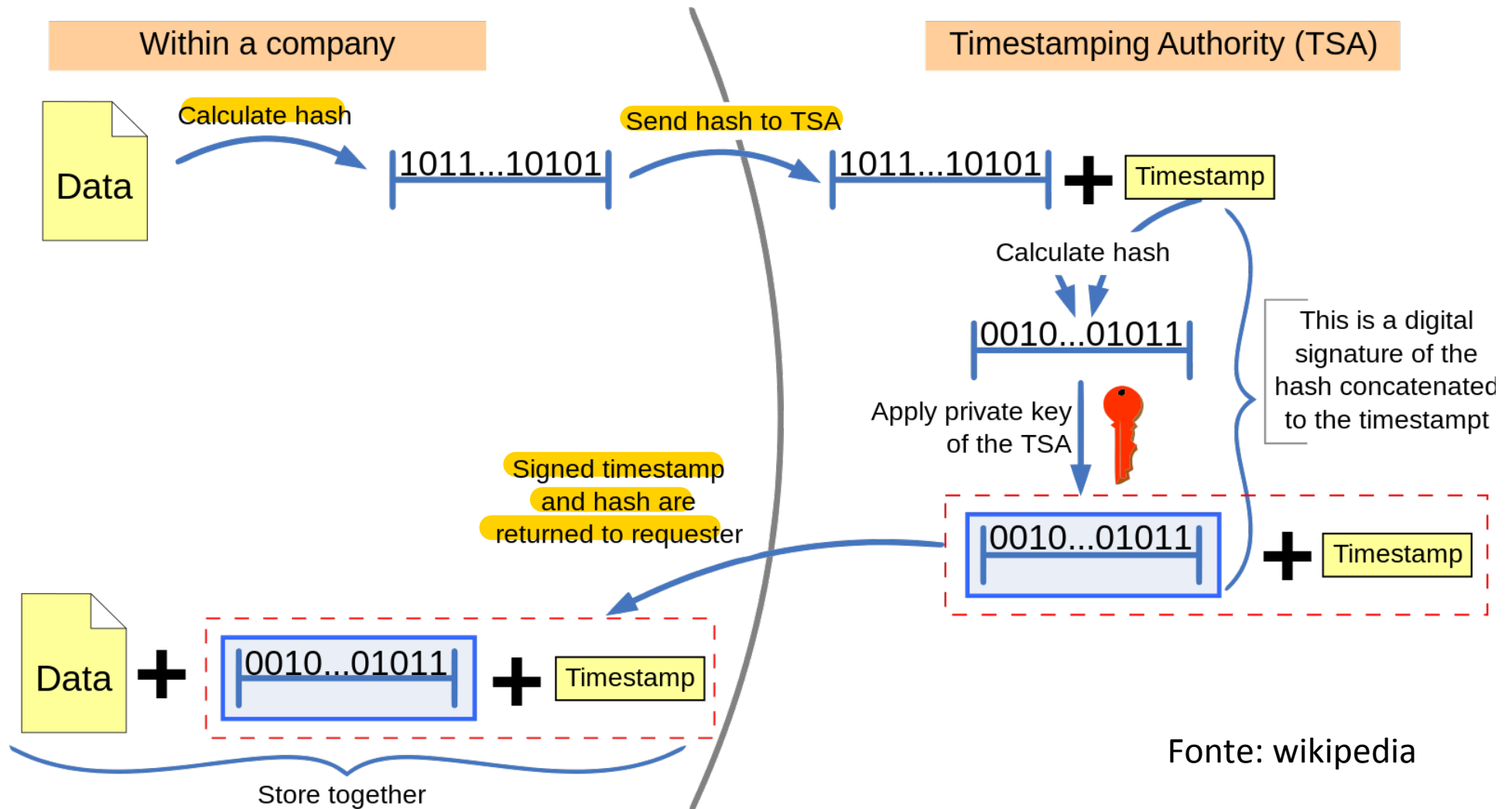
Soluzione *ingenua*

H: funzione hash

- Si invia il valore hash ad un documento D ad un'autorità fidata (TTP)
 - È chiamata TSA (TimeStamping Authority)
- L'autorità aggiunge un timestamp T a H(D)
 - Ad esempio: T = YYYY-MM-DD HH:MM:SS
- L'autorità firma T || D ed invia il messaggio firmato al richiedente

Grande fiducia nella TSA

Trusted timestamping



Protocollo Distribuito

- Generalizzazione della soluzione precedente
- Vogliamo datare un documento D
- Si calcola $y = H(D)$ e si usa y come seme di un PRNG generando k valori V_1, V_2, \dots, V_k
- Si considera V_i come *l'identità* di una persona (TSA) a cui inviare y
 - Se abbiamo 2^b TSA, ogni blocco di b bit di $\text{PRNG}(y)$ identifica un TSA
- Ogni V_i aggiunge tempo e data ad y , firma il tutto e restituisce il risultato
- $H(D)$ e le k firme ricevute sono conservate come marcatura temporale del documento D

Chiarimenti sul protocollo

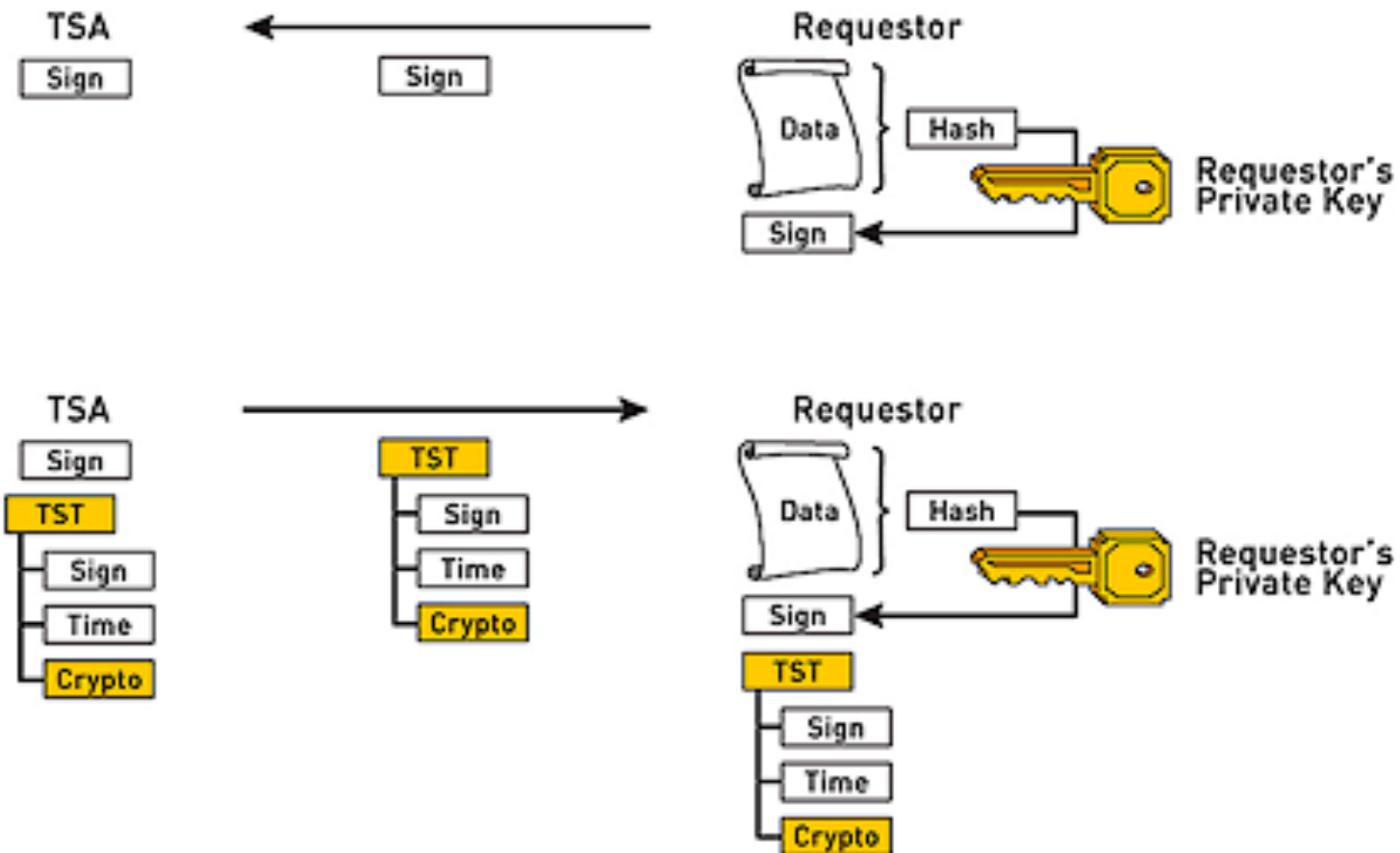
- Il valore k deve essere grande a sufficienza in modo che risulti difficile corrompere tante TSA
- La scelta delle TSA da contattare deve essere effettuata a caso per ogni documento, ecco perché si usa un PRNG avente come seme il valore hash del documento

Problemi del Protocollo Distribuito

- Ci vogliono molte TSA in grado di rispondere immediatamente alla richiesta di timestamp
- Durata (vita) delle firme digitali:
 - Una firma potrebbe non essere più valida al momento della verifica della marca temporale:
 - La chiave privata è stata compromessa
 - Lo schema di firme è stato rotto
 - Il certificato associato alla chiave di firma è scaduto

Estendere validità firme digitali

- Una marca temporale può essere associata anche a un documento su cui è stata una firma digitale
- Marcando temporalmente la firma del documento garantiamo che essa sia sempre valida anche nel caso in cui il relativo certificato risulti scaduto, sospeso o revocato
 - La marca deve essere apposta precedentemente alla scadenza, revoca o sospensione del certificato di firma



Fonte: wikipedia

Una singola TSA

- Problemi
 - Dobbiamo fidarci della TSA
 - In un qualsiasi momento, una TSA corrotta potrebbe apporre una marca temporale relativa ad una qualsiasi data precedente a quella attuale
- Soluzione
 - *Collegare* in qualche modo tutti i documenti marcati dalla TSA

Catena di Marche Temporal

- Soluzione proposta nel 1991 da Haber e Stornetta

S. Haber e W.S. Stornetta

[How to time-stamp a digital document](#)

Journal of Cryptology, Vol. 3 (2), 99–111, 1991

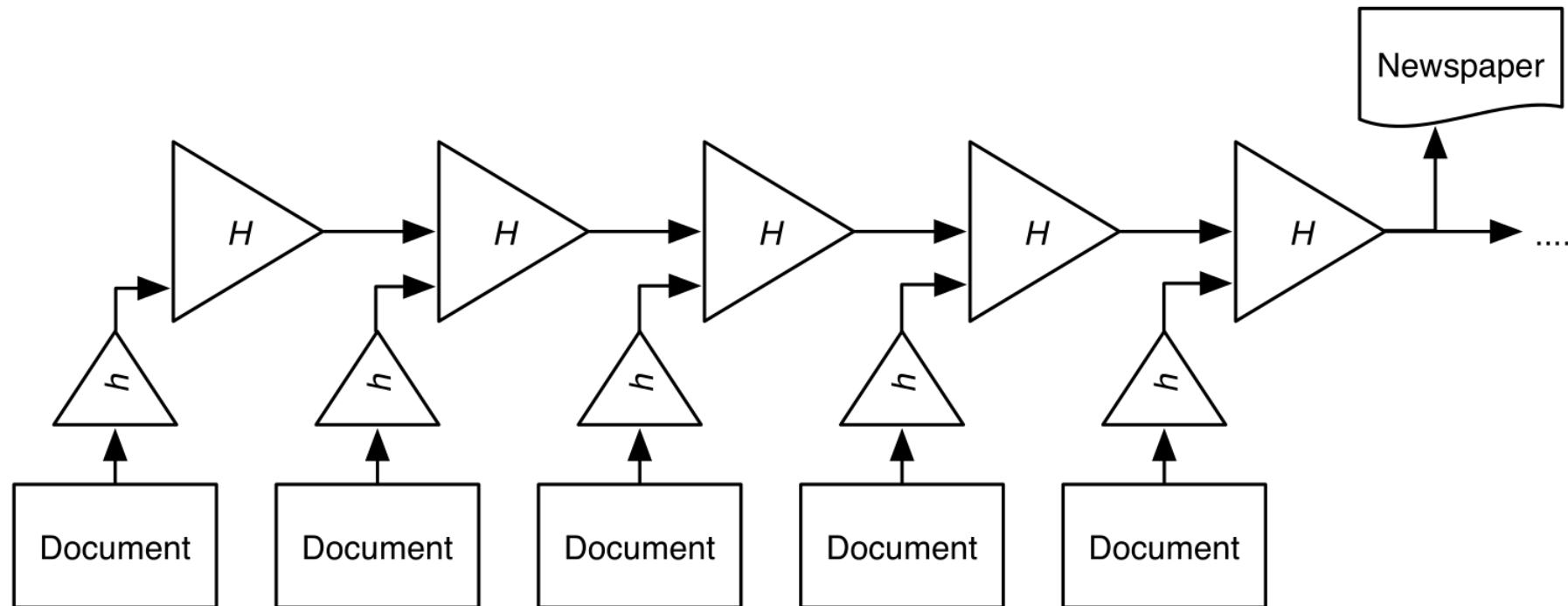
- **Notazione:**

- Sig funzione di firma della TSA
- h, H funzioni hash
- D , documento da marcare
- $y_n = h(D)$, n -esimo documento che la TSA deve marcare
- ID_n identità del richiedente

Il protocollo graficamente



17 luglio 2009



Protocollo

- Si invia $y_n = h(D)$ e ID_n alla TSA
- La TSA risponde con $s = \text{Sig}(n, t_n, ID_n, y_n; L_n)$
 - t_n è il timestamp
 - $L_n = (t_{n-1}, ID_{n-1}, y_{n-1}; H(L_{n-1}))$ *linking information*
- Quando arriverà una nuova richiesta di marcatura da ID_{n+1} , allora ID_{n+1} sarà inviato a ID_n e (s, ID_{n+1}) è la marca temporale di y_n

È necessario il valore iniziale L_0 , potremmo settare $L_0 = (0, 0, 0; 0)$

Marca Temporale n-esima

$\langle \text{Sig}(n, t_n, \text{ID}_n, y_n; (t_{n-1}, \text{ID}_{n-1}, y_{n-1}; H(L_{n-1})), \text{ID}_{n+1}) \rangle$

- Perché è sufficiente inserire solo l'hash di L_{n-1} ?

È sufficiente notare che L_{n-1} include L_{n-2}
che a sua volta deve includere $L_{n-3} \dots$

Verifica Marca Temporale

- ID_n verifica la firma $Sig(n, t_n, ID_n, y_n; L_n)$
- Chiede a ID_{n+1} la sua marca temporale
- Verifica che tutto coincide
- Chiede a ID_{n-1} la sua marca temporale
- Verifica che tutto coincide
- Può continuare il procedimento con ID_{n+2} , ID_{n-2} e così via

Sicurezza del sistema

- Non è possibile inserire una nuova marca nella catena perché i messaggi sono numerati
- Per cambiare un messaggio marcato un utente
 - Deve colludere con la TSA, con l'utente che lo precede e con quello che lo segue nella catena
 - Deve anche trovare una collisione in H

$$\langle \text{Sig}(n, t_n, \text{ID}_n, y_n; (t_{n-1}, \text{ID}_{n-1}, y_{n-1}; H(L_{n-1}))), \text{ID}_{n+1} \rangle$$

Sicurezza del Sistema

- Vogliamo sostituire $(n, t_n, \text{ID}_n, y_n; L_n)$ con $(n, t_n, \text{ZZ}_n, z_n; L_n)$

$\text{ZZ}_n \neq \text{ID}_n$
 $z_n \neq y_n$
- Bisogna colludere con ID_{n-1} per sostituire (s, ID_n) con (s, ZZ_n)
- Bisogna colludere con ID_{n+1} per sostituire $L_{n+1} = (t_n, \text{ID}_n, y_n; H(L_n))$ con $L'_{n+1} = (t_n, \text{ZZ}_n, z_n; H(L'_n))$ e deve risultare $H(L_{n+1}) = H(L'_{n+1})$

Sicurezza del Sistema

- Si potrebbe rompere lo schema colludendo solo con la TSA e creando una falsa catena lunga *a sufficienza*
- Si risolve il problema pubblicando $H(L_m)$ ad intervalli regolari.
 - Una volta al giorno su Internet o su di un giornale

Migliorare la sicurezza

- Collegare ogni richiesta alle precedenti k ed alle successive k

$$L_n = ((t_{n-k}, ID_{n-k}, y_{n-k}; H(L_{n-1}), \dots, (t_{n-k}, ID_{n-1}, y_{n-1}; H(L_{n-1})))$$

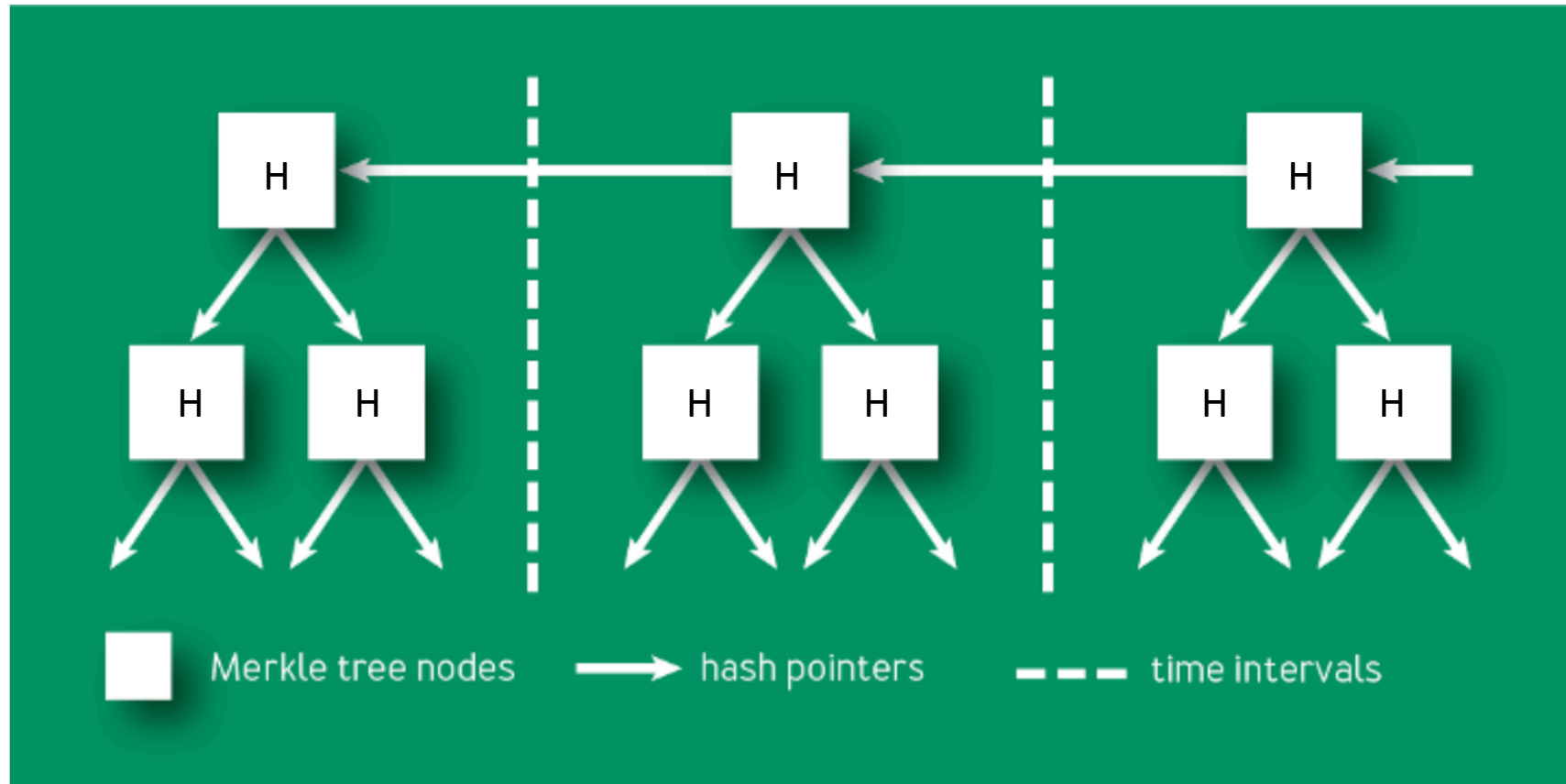
- Una volta che le successive k richieste sono state elaborate la TSA invia a ID_{n+1} le identità $ID_{n+1}, \dots, ID_{n+k}$

Albero di Marche Temporali

- La struttura utilizzata è un albero binario (Merkle Tree) che sostituisce la lista doppiamente concatenata
- Il TSS produce una marca temporale dopo che, in un'unità di tempo (*timeframe*), ha esaminato un numero fissato di richieste

Merkle Tree

Albero di valori hash

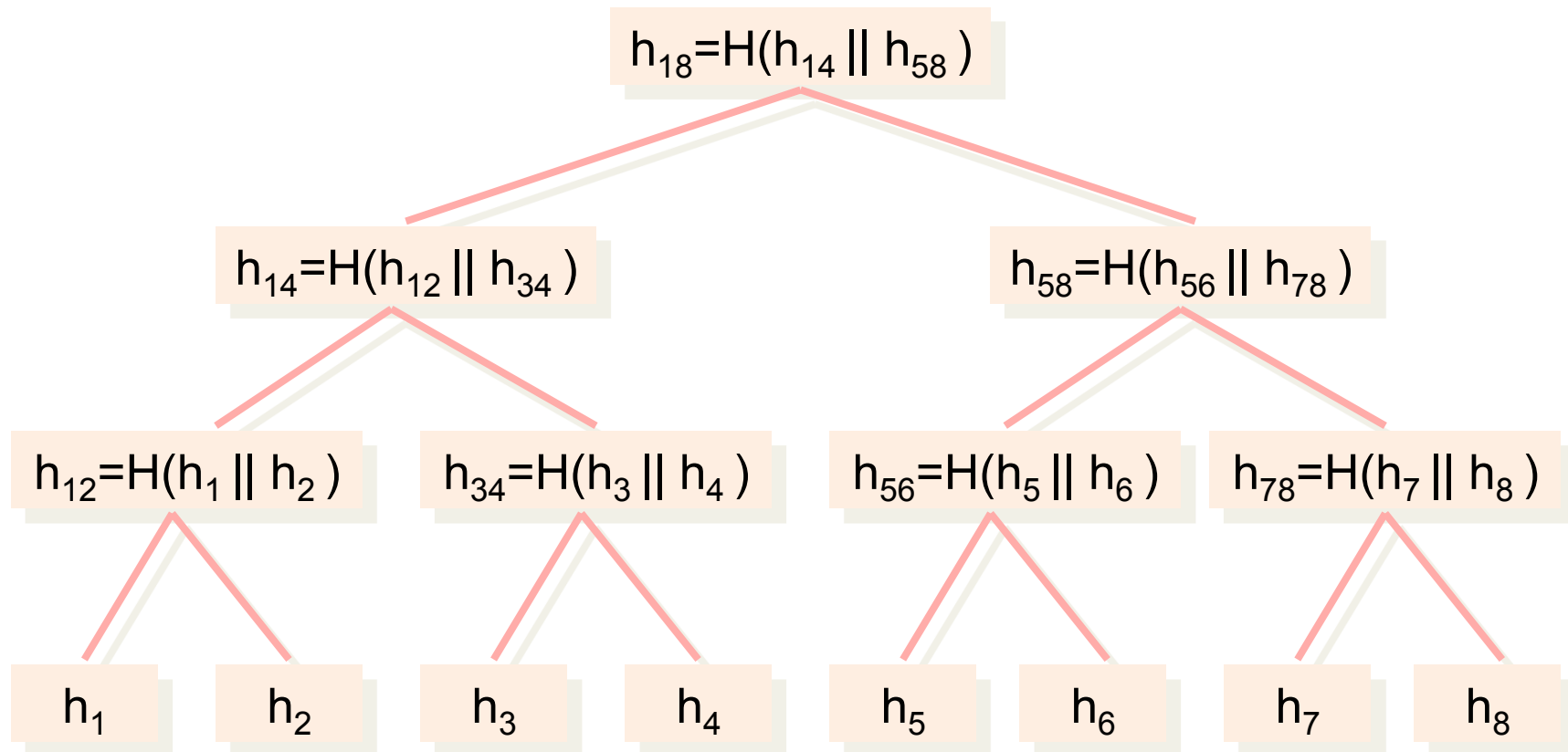


Utilizzato anche in Bitcoin

Struttura dell'albero

$$HV_i = h_{18}$$

\parallel indica la concatenazione



$$h_i = h(D_i)$$

Il protocollo

- La TSA riceve n richieste nello stesso intervallo di tempo (*timeframe*) t_i
- La TSA calcola il valore HV_i (*root hash*) e lo rende pubblico
- La marca temporale di un utente (e.g., ID_4) contiene informazioni per poter ricostruire il valore HV_i , ad esempio
 - $h_4, (h_3, sx), (h_{12}, sx), (h_{58}, dx)$
 - sx/dx indicano se il valore è nel nodo sinistro o destro

Collegamento tra timeframe

- Si calcola e pubblica un Super Hash Value
 - $SHV_i = H(SHV_{i-1} || HV_i)$, è necessario un SHV_0
 - SHV_{i-1} e SHV_i possono essere inserite nel timestamp
- Verifica
 - Non sono necessari i timestamp di altri utenti, tutto è codificato nel timestamp ricevuto
 - Si può controllare l'HashValue (HV) pubblicato corrispondente al proprio timestamp
 - Si può controllare la correttezza della catena dei Super Hash Value

Sicurezza del Sistema

- Fissato il valore hash della radice, non è possibile
 - Inserire/Cambiare anche un valore nell'albero Merkle
 - Per fare ciò dovremmo essere in grado di calcolare collisioni di funzioni hash
 - Due valori x_1 ed x_2 tali che $H(x_1) = H(x_2)$

Sicurezza del Sistema

- Si potrebbe rompere lo schema colludendo solo con la TSA e creando un insieme *sufficientemente grande* di alberi collegati
- Tale attacco è limitato notevolmente pubblicizzando ad intervalli regolari il Super Hash Value

Digital Notary <http://www.surety.com>

- L'utente usa un applicativo venduto dalla Surety
- La funzione hash produce un digest di 416 bit
 - Prodotto dalla concatenazione di SHA-256 e RIPEMD-160
- Il sistema usa una struttura ad albero
- L'unità di tempo corrisponde ad un secondo
- Un numero seriale è inserito nel documento.
- Il SHV è pubblicato in posti accessibili via rete, su un CD-ROM, ed ogni settimana sul Sunday New-York Times

Progetto #3

- Implementare un servizio di timestamping
 - Non deve essere un'applicazione client/server
 - Il server esaminerà un lotto di richieste e genererà le corrispondenti marche temporali
 - Le richieste degli utenti sono cifrate con la chiave pubblica della TSA
- L'altezza dell'albero è 3
 - Si possono apporre 8 marche temporali in un timeframe
 - Se ci sono meno di 8 documenti, aggiungere nodi fittizi

Possibili campi in una Marca Temporale

- L'identificativo del mittente
- Il numero di serie della marca temporale
- Il tipo di algoritmo di firma della Marca Temporale
- L'identificativo del certificato della chiave pubblica della TSA con cui ha firmato la Marca
- Data ed ora in cui la Marca è stata generata
- Il digest calcolato dalla TSA partendo da quello fornito dal richiedente
- La firma digitale della marca apposta dalla TSA

KeyRing

- Deve conservare
 - Chiavi pubbliche e private di firma e di cifratura
 - Chiavi di cifrari simmetrici
 - Password di accesso a siti web
- Deve essere conservato in un file cifrato
 - Libera scelta per l'organizzazione delle informazioni nel KeyRing
 - Nella documentazione indicare le modalità di accesso al KeyRing e il recupero delle informazioni associate alle chiavi

Può essere consegnato con l'ultimo progetto