e-privacy 2009
Firenze, 22-23 maggio 2009

Prevenire il computer quantistico: lo schema di firma digitale Lamport.

Tommaso Gagliardoni
gaglia@anche.no

Licenza

Sommario

- Parte 1: il computer Quantistico
- Parte 2: QC, crittografia e privacy
- Parte 3: lo schema di firma digitale Lamport
- Parte 4: varianti dello schema Lamport
- Conclusioni

Il computer quantistico

Timeline essenziale del Computer Quantistico *(QC)*:

- **1981**: primo modello teorico di QC (R. Feynman)
- **1984**: invenzione del primo protocollo di Crittografia Quantistica
- **1994**: algoritmo di Shor (fattorizzazione di interi, logaritmo discreto)
- **1996**: algoritmo di Grover (ricerca in database, inversione di funzioni)
- **1998**: realizzazione di un QC a 3 qubit
- **2000**: realizzazione di un QC a 7 qubit
- **2001**: l'algoritmo di Shor viene usato su un QC per fattorizzare il numero 15
- **2006**: realizzazione di un QC a 12 qubit, primo teletrasporto di informazione
- **2008**: D-Wave Systems afferma di aver realizzato un QC a 128 qubit *(?)*

Il computer quantistico

Come mai tutto questo interesse per il QC?

Motivi teorici

- studio efficiente di sistemi quantistici
- teoria dell'informazione
- teoria della complessità

Motivi economici e militari

- applicazioni pratiche (nanotecnologia, medicina, chimica)
- tecnologie correlate (crittografia quantistica, teletrasporto d'informazione, etc.)
- ***violazione di quasi tutti gli schemi crittografici classici oggi noti***

Il computer quantistico

Cosa il QC può fare:

Risolvere *alcuni* problemi in maniera più efficiente di un computer classico

Ricercare un elemento in database non ordinati e invertire funzioni (es: funzioni hash) in maniera *abbastanza* veloce (speedup quadratico da O(N) a $O(N^{1/2})$)

Fattorizzare interi e risolvere logaritmi discreti (in N bit) in maniera *molto* veloce (speedup da subesponenziale a polinomiale $O(N^3)$)

Cosa il QC *non* può fare:

Risolvere problemi che non siano risolubili *anche* con un computer classico

Brute-forcing di un algoritmo a chiave simmetrica in tempo polinomiale

Risolvere SAT e altri problemi NP-completi in tempo polinomiale

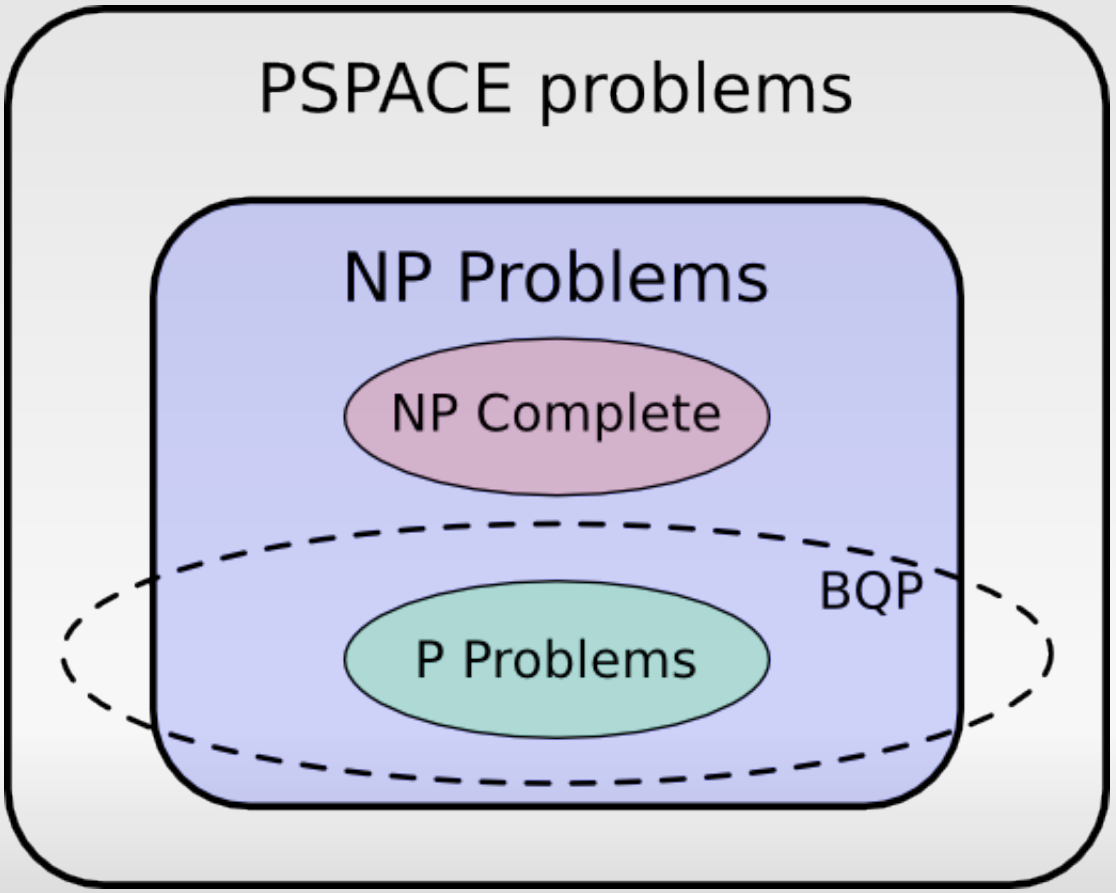
Invertire funzioni hash in tempo polinomiale

Il computer quantistico

Cosa *non si sa* del QC:

La classe BQP (Bounded, Quantum, Polynomial time)

è la classe di ”tutti i problemi che sono facilmente risolubili con un computer quantistico”



Sommario

- Parte 1: il computer Quantistico
- Parte 2: QC, crittografia e privacy
- Parte 3: lo schema di firma digitale Lamport
- Parte 4: varianti dello schema Lamport
- Conclusioni

QC, crittografia e privacy

Ipotizziamo che il QC diventi una realtà praticamente applicabile. Quali conseguenze?

- totale forzatura dei cifrari RSA, DSA, ElGamal e derivati
- violabilità di tutti i protocolli di sicurezza telematica che si basino su tali cifrari
- forzatura di tutti gli strumenti per la difesa della privacy delle comunicazioni (Tor, Freenet, PGP/GnuPG, Anonymous remailers, etc.)
- controllo totale di e-commerce, segreti militari, telecomunicazioni
- necessario un immediato raddoppio della lunghezza delle chiavi simmetriche

Secondo voi cosa staranno facendo all'NSA in questo momento?

Sommario

- Parte 1: il computer Quantistico
- Parte 2: QC, crittografia e privacy
- Parte 3: lo schema di firma digitale Lamport
- Parte 4: varianti dello schema Lamport
- Conclusioni

Lo schema di firma digitale Lamport

Ingredienti:

- una funzione hash crittograficamente sicura, **H**
- un generatore di numeri pseudocasuali crittograficamente sicuro, **R**

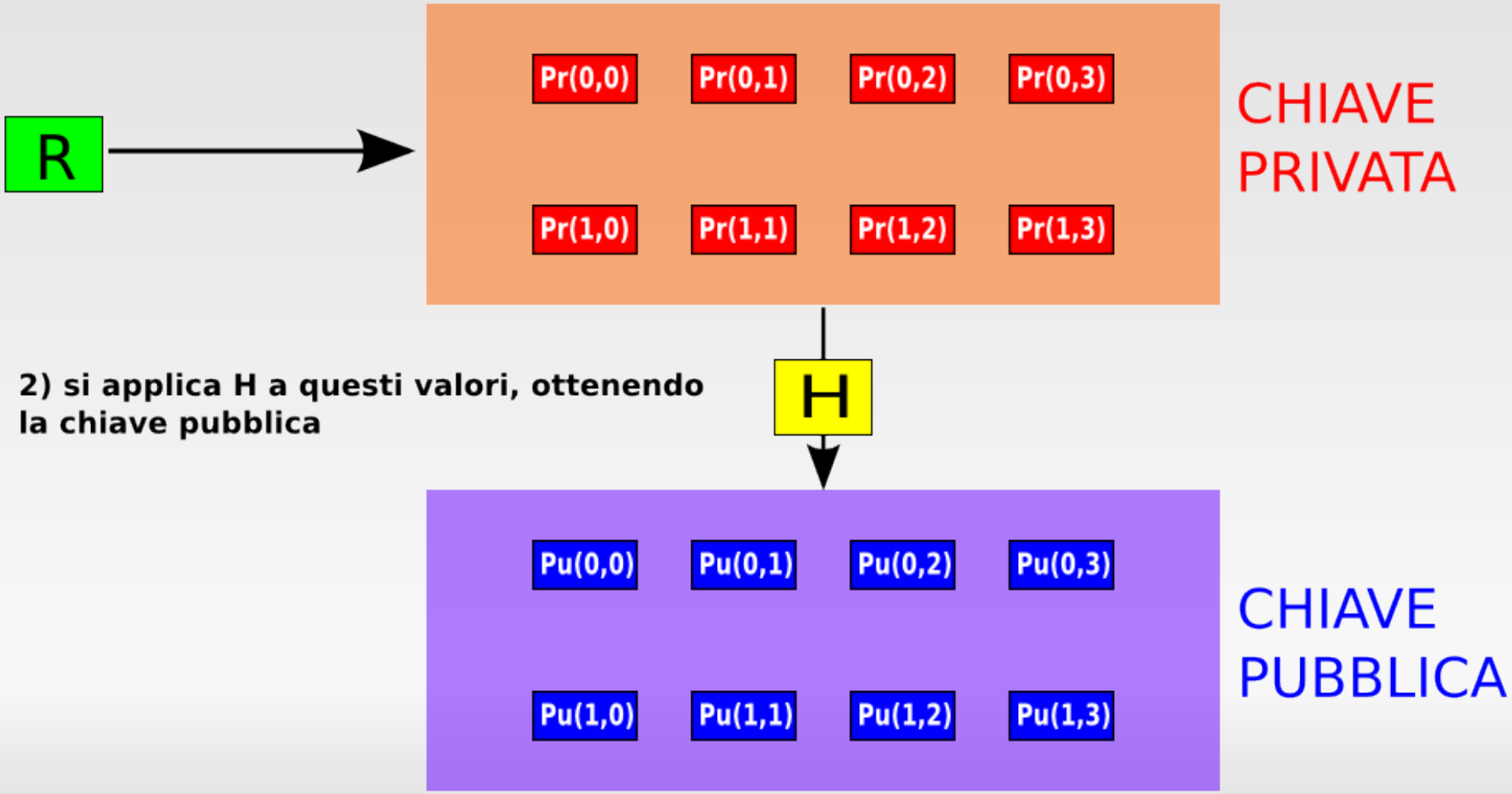
Teoricamente parlando la sola esistenza di questi due fondamentali componenti è tutta da dimostrare. Ai nostri scopi ci accontenteremo di strumenti standard quali SHA e PRNG classici. È importante però ricordare che deve essere posta la *massima* cura nella scelta di questi due componenti.

Supponiamo che H ed R producano stringhe di N bit. N deve essere sufficientemente grande da proibire la ricerca di collisioni. Nella spiegazione seguente assumiamo però N=4, per semplicità.

Lo schema di firma digitale Lamport

Generazione della coppia di chiavi

1) Si usa R per generare 2N valori casuali, che compongono la chiave privata

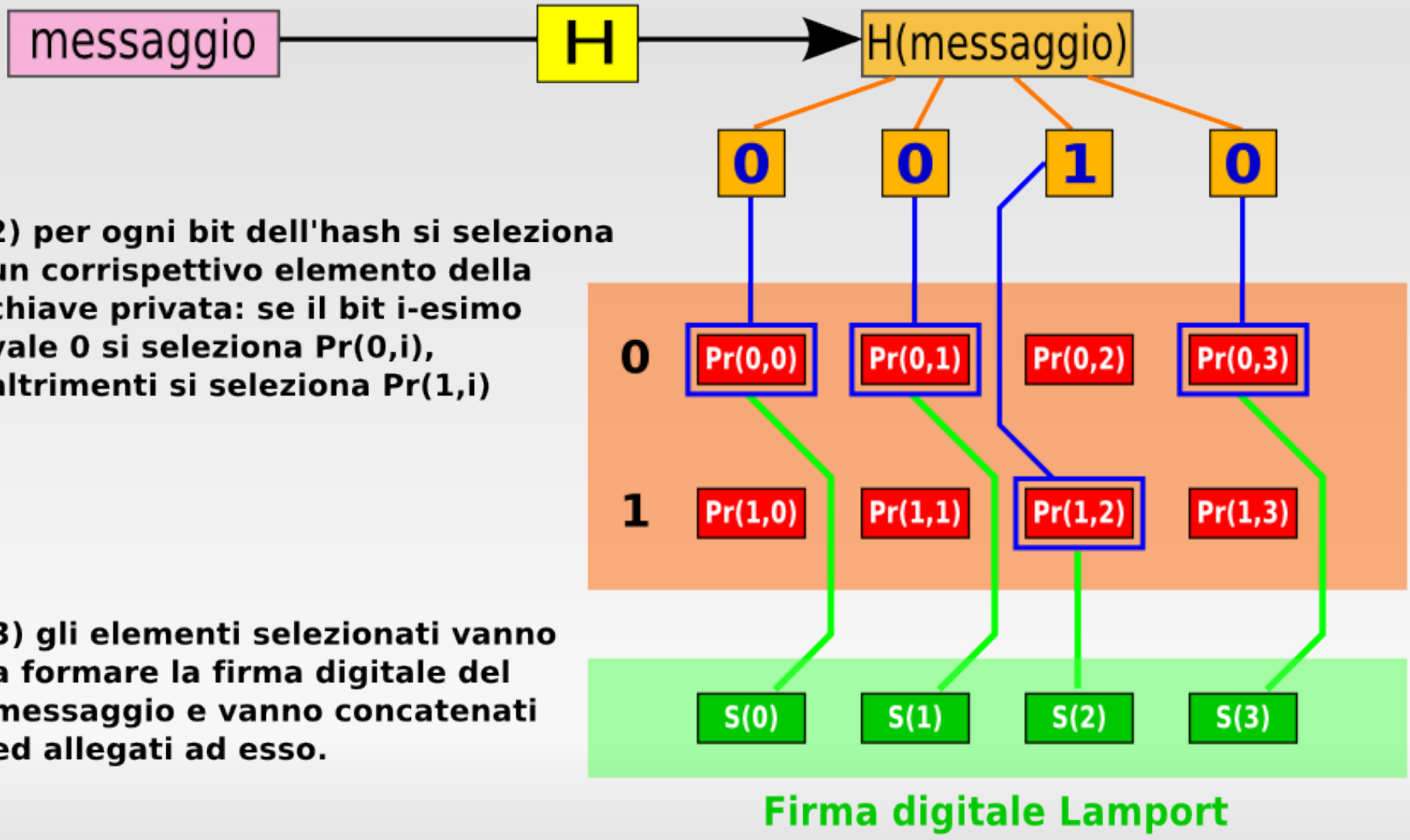


2) si applica H a questi valori, ottenendo la chiave pubblica

Lo schema di firma digitale Lamport

Firma di un messaggio

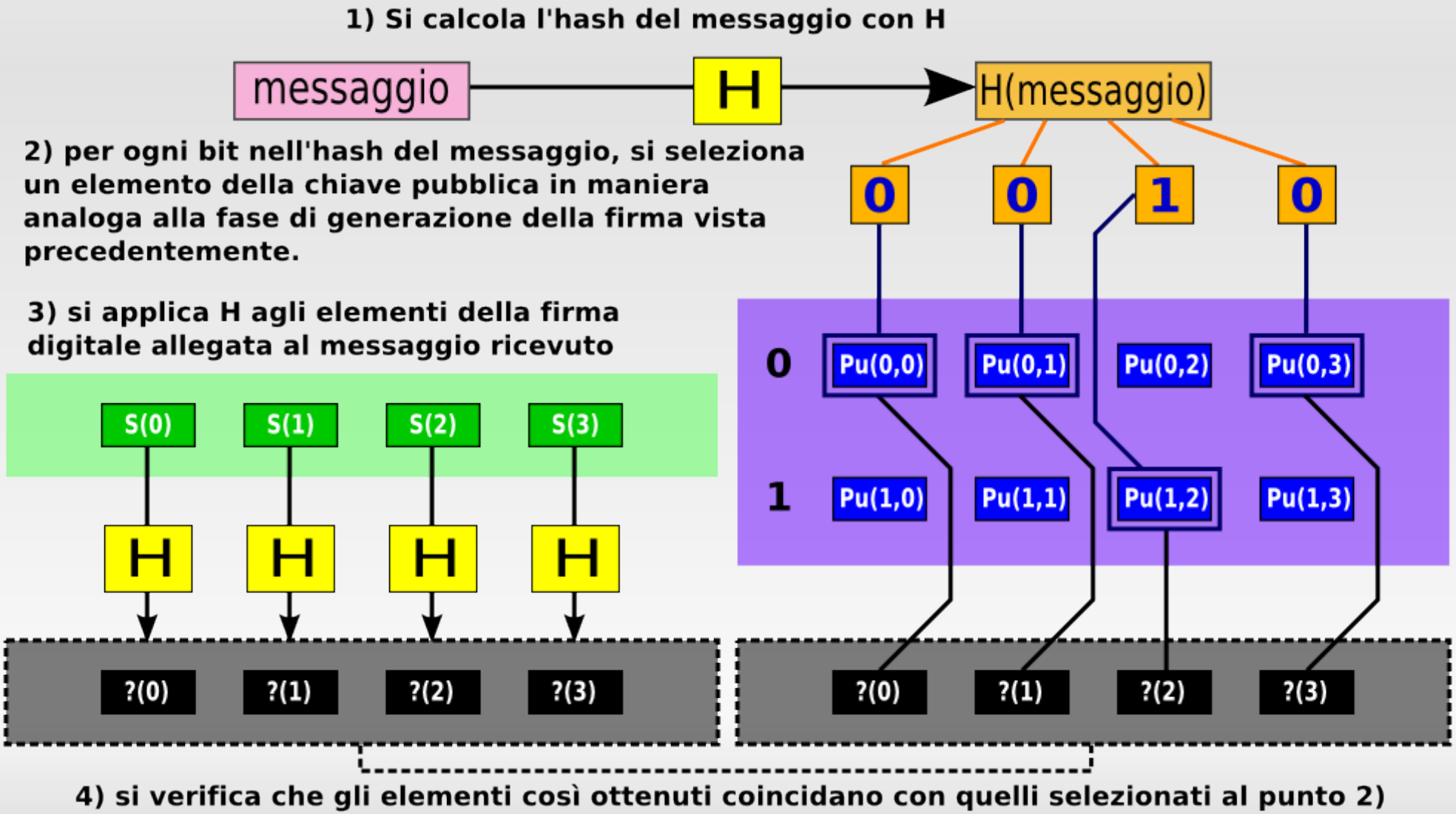
1) Si calcola l'hash del messaggio con H



Firma digitale Lamport

Lo schema di firma digitale Lamport

Verifica della firma



Lo schema di firma digitale Lamport

Vantaggi di questo schema:

- Fa uso solo di funzioni hash: per quanto visto prima un QC non riuscirebbe a forgiare facilmente firme fasulle

- Ciò implica anche che non ci sarà mai da temere nuove scoperte sensazionali nel campo della fattorizzazione e similari (come avviene invece per RSA etc), l'unico modo di attaccare questo schema è di trovare collisioni in H o di prevedere R, tutte cose che potrebbero comunque mettere in crisi qualsiasi altro schema noto

- Implementazione hardware e software incredibilmente facili, velocità di calcolo elevatissima (niente librerie aritmetiche richieste, meno possibilità di bug)

Lo schema di firma digitale Lamport

Problemi di questo schema:

- ogni coppia di chiavi può essere usata una sola volta, o un avversario potrebbe riutilizzare parte delle firme precedentemente usate per forgiarne di nuove (notare infatti che ogni firma rivela una metà casuale della chiave privata)

- le firme allegate ai messaggi sono lunghe N^2 bit: nel caso di H = SHA-512 ad esempio ogni firma occuperebbe 512 * 512 = 262144 bit = 32 KiB

- ogni chiave pubblica e ogni chiave privata è lunga $2N^2$ bit: nell'esempio precedente ciascuna occuperebbe 64 KiB (una coppia di chiavi: 128 KiB)

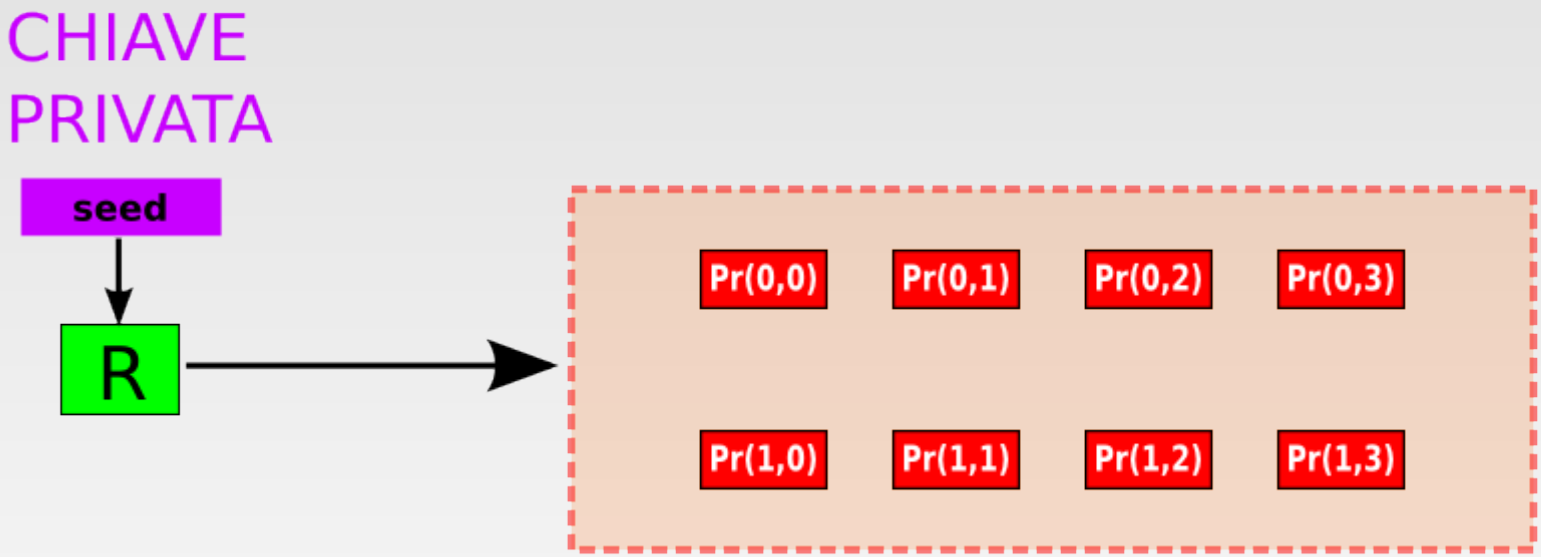
Sommario

- Parte 1: il computer Quantistico
- Parte 2: QC, crittografia e privacy
- Parte 3: lo schema di firma digitale Lamport
- Parte 4: varianti dello schema Lamport
- Conclusioni

Varianti dello schema Lamport

Variante 1: chiave privata corta

Invece di immagazzinare come chiave privata tutto l'output del PRNG R, basta salvare il seed crittografico di tale generatore. Quella sarà la nuova chiave privata.



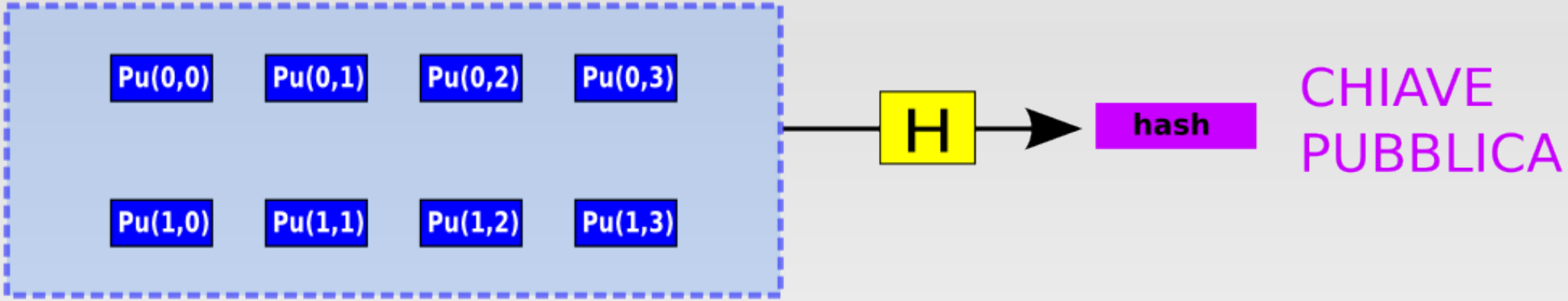
La dimensione della chiave privata passa da 2*N*N a un numero di bit variabile a seconda del PRNG e del grado di sicurezza desiderato.

L'unico svantaggio è dato dal fatto che ogni volta che si desidera firmare un documento bisogna ricalcolare i valori Pr(i,j).

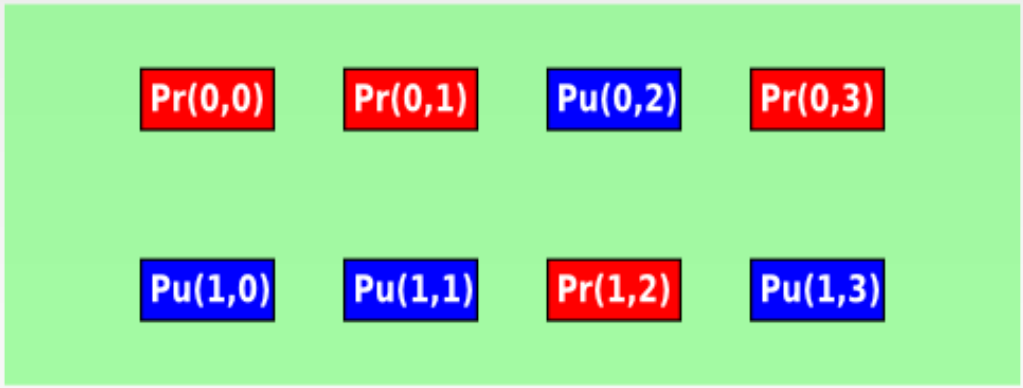
Varianti dello schema Lamport

Variante 2: chiave pubblica corta

Invece di pubblicare come chiave pubblica tutti i 2N valori Pu(i,j) se ne pubblica solo l'hash



A questo punto però bisogna includere nella firma Lamport, oltre ai valori Pr(i,j) selezionati col solito metodo, anche i valori Pu(i,j) non utilizzati. La firma diventa cioè del tipo:



Firma digitale Lamport

(notare che la dimensione della firma raddoppia)

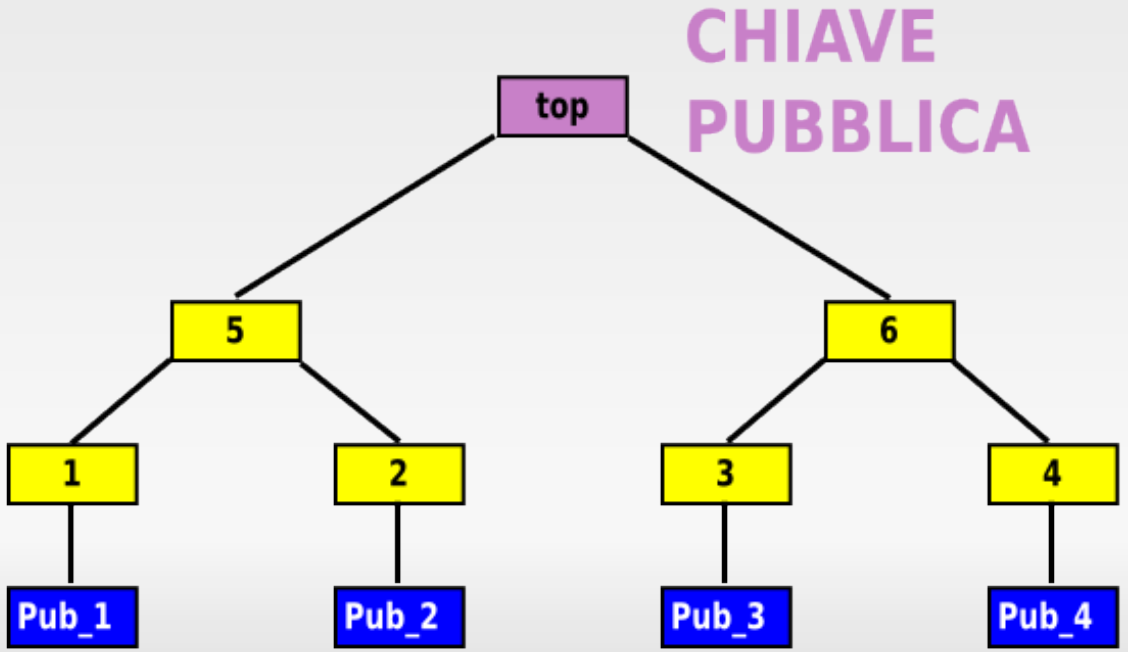
Per verificare la firma bisognerà prima trasformare i Pr(i,j) in Pu(i,j) (il verificatore può farlo tramite il solito processo di selezione basato sull'hash del messaggio) e poi verificare che l'hash di tutti i blocchi così ottenuti coincida con la chiave pubblica nota.

Varianti dello schema Lamport

Variante 3: firme multiple

Supponiamo di volere una chiave che possa fimare fino a 4 documenti diversi. Generiamo allora 4 diverse coppie di chiavi pubblica/privata:

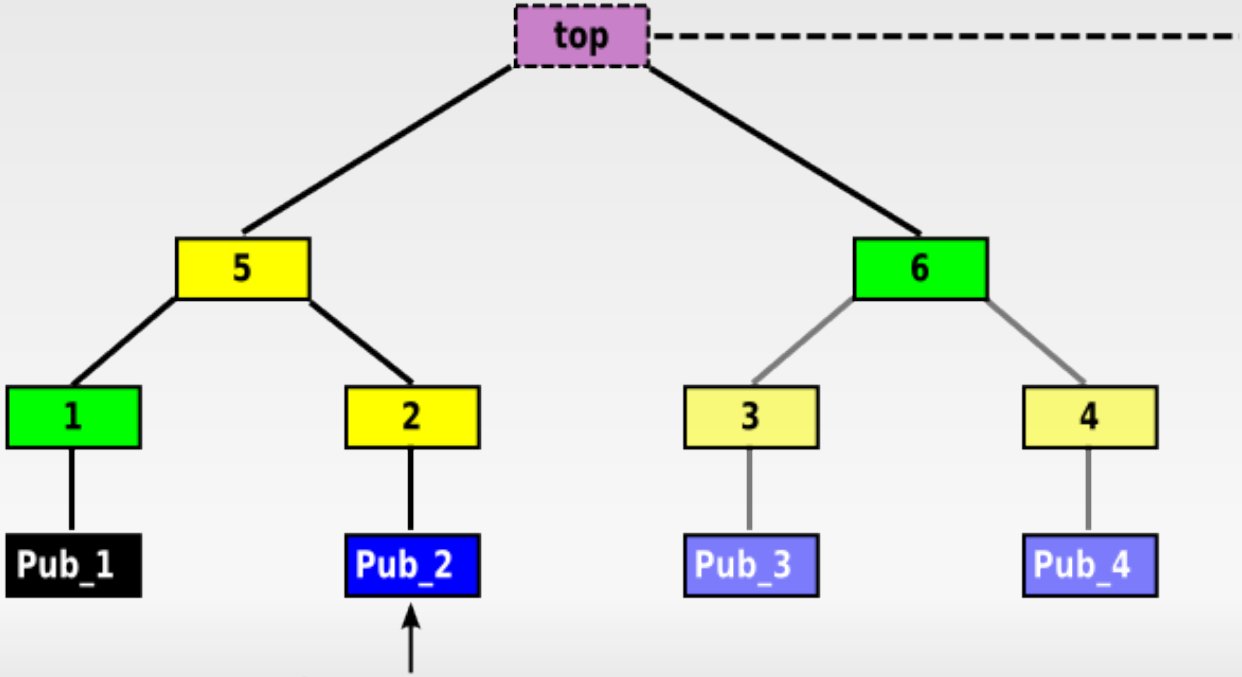
Ora costruiamo un Hash Tree (o Merkle Tree, in questo caso binario) usando come foglie gli hash delle chiavi pubbliche generate. Il "top hash" dell'albero sarà la vera chiave pubblica.



Varianti dello schema Lamport

Questa struttura può ora essere usata per firmare fino a 4 documenti: ogni volta dovremo scegliere una tra le 4 chiavi ed usarla per generare una firma. Ognuna delle 4 chiavi è monouso. Per generare una firma dovremo allegare al messaggio, oltre agli hash dati dalla normale procedura di firma già vista, alcuni nodi intermedi dell'albero, in modo da poter fornire al verificatore le informazioni necessarie a risalire (e controllare) al top hash.

Supponiamo ad esempio di aver già "bruciato" la chiave 1, e di voler firmare un messaggio con la chiave 2. Allora oltre alla firma generata dalla chiave 2 dovremo allegare al messaggio i nodi marcati in verde:



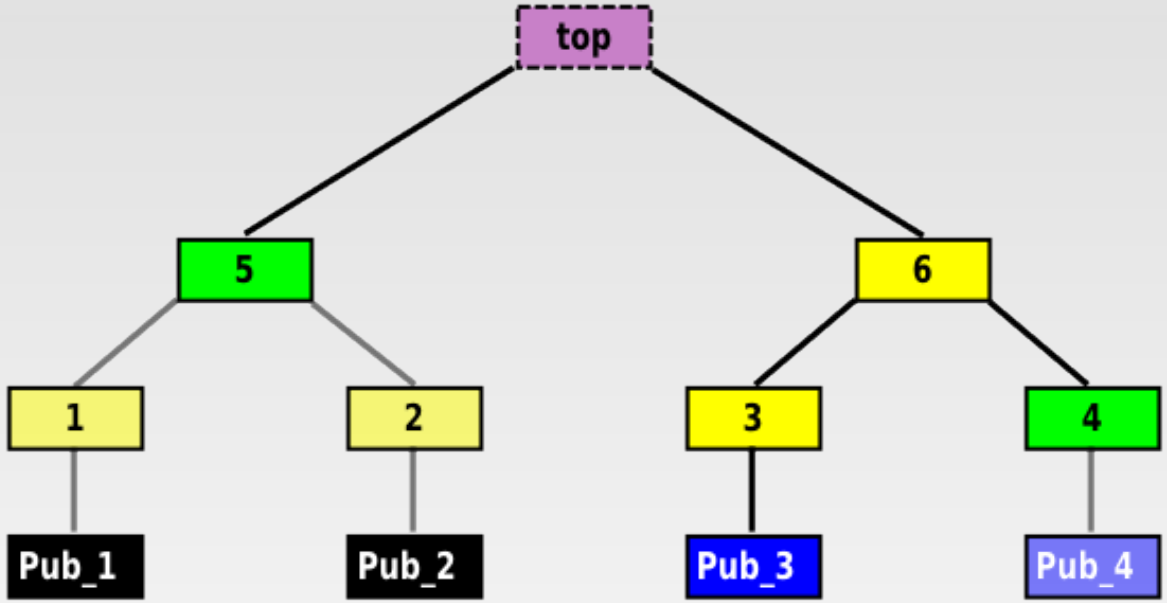
La firma è valida se il top hash ricavato coincide con la chiave pubblica nota.

CHIAVE PUBBLICA

(il verificatore può ricavare Pub_2 seguendo il metodo standard di verifica)

Varianti dello schema Lamport

Altro esempio: supponiamo di aver "bruciato" le chiavi 1 e 2 e di voler usare la chiave 3:



Vantaggi: si può pregenerare una chiave con un numero arbitrariamente alto di utilizzi (ricordiamo tralaltro che sarebbe buona usanza far "scadere" le chiavi dopo un certo periodo)

Svantaggi: la dimensione della firma aumenta leggermente.

Sommario

- Parte 1: il computer Quantistico
- Parte 2: QC, crittografia e privacy
- Parte 3: lo schema di firma digitale Lamport
- Parte 4: varianti dello schema Lamport
- Conclusioni

Conclusioni

- Il computer quantistico non è fantascienza, ed è in grado di mettere in crisi crittosistemi che finora si ritenevano sicurissimi. Le conseguenze per la sicurezza e la privacy sono evidenti.
- Il computer quantistico ha delle debolezze, sfruttando le quali si può provare a progettare nuove soluzioni, ad esempio il sistema di firma digitale Lamport.
- Il crittosistema Lamport ha degli svantaggi, che però possono essere di molto ridimensionati con opportune tecniche.
- Per quanto riguarda il futuro sarebbe buona cosa essere previdenti: c'è ancora troppa indifferenza verso questa tematica, e se ne parla ancora troppo poco in ambienti non accademici.
- Ci sono ancora molte cose che non si sanno sul quantum computer, vale la pena approfondire la ricerca in questo campo.

Fine.

Domande?