Tesi di Laurea

Università degli studi di Catania
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica Applicata, primo livello

Schemi di Firma per Network Coding Sicuro basati su RSA

Autore: Daniele Licitra
Relatore: Dario Catalano

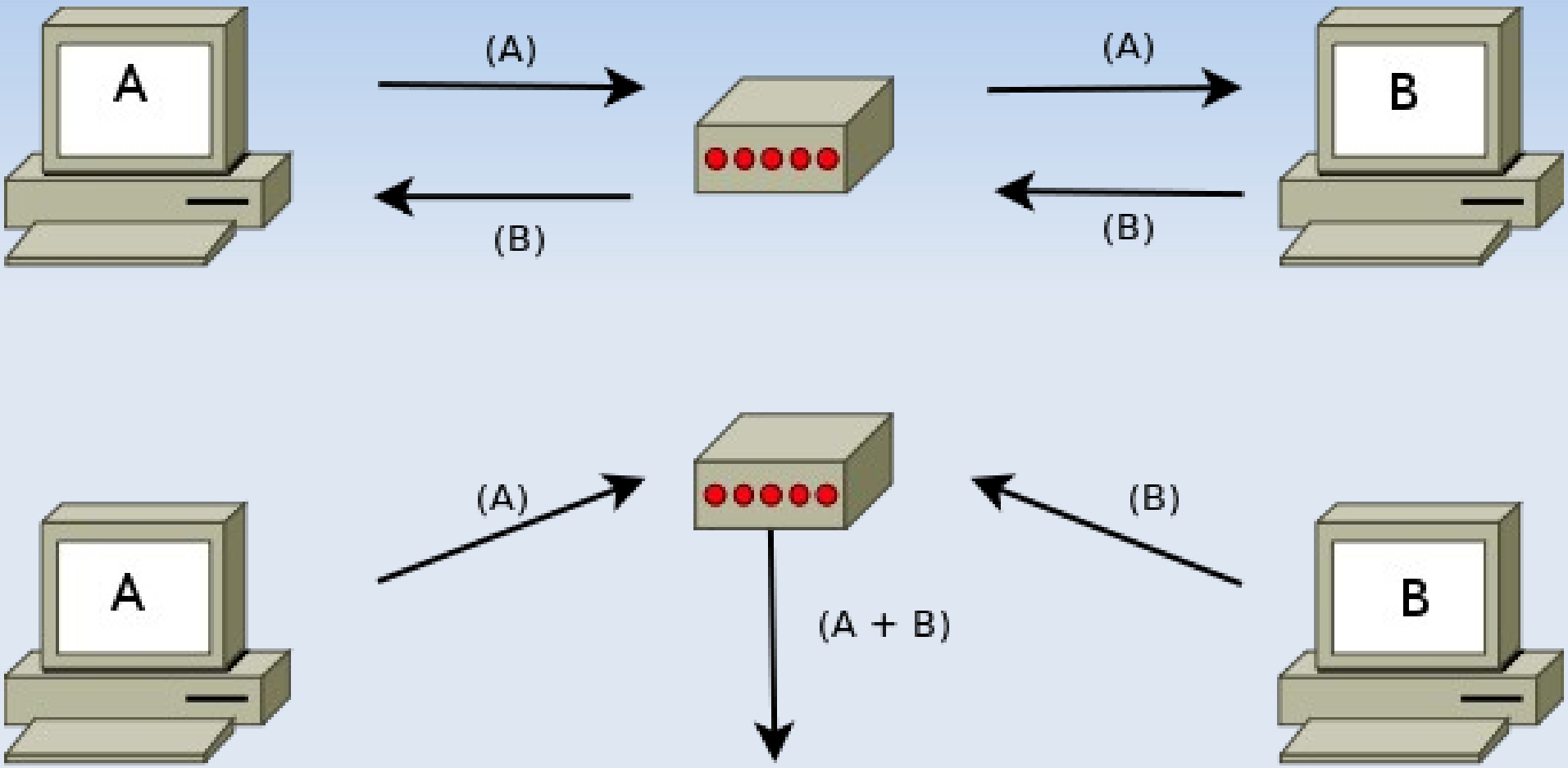
A.A. 2009/10

Network Coding

- Il Network Coding è un metodo generico di routing.
- Per **routing** si intende l'insieme delle scelte atte a trasferire un'informazione dal mittente al destinatario.
- Lo **Store-and-Forward** è il metodo di routing più noto.

- Nel **NetworkCoding:**
- i pacchetti sono constituiti da dei vettori;
- i nodi intermedi modificano i pacchetti combinando i vettori ricevuti.

Network Coding



Network Coding: funzionamento

- Il file $\overline{F}$ viene suddiviso in m vettori $v^{(i)}$ che vengono inviati dalla sorgente.
- I nodi intermedi ricevono l vettori e ne calcolano la combinazione lineare.
- La destinazione ricostruisce il file $\overline{F}$ dai vettori che riceve.
- Per i dettagli tecnici si rimanda alla tesi.

Applicazioni ed esempi

- Il Network Coding utilizzato in reti wireless ne incrementa notevolmente il *throughput* e l'affidabilità. Esempio: *Cope* è un protocollo per reti wireless che effettua la XOR dei vettori in transito.

- Nel P2P (peer-to-peer) riduce i tempi di trasferimento dei file. Esempio : *Avalanche* è un programma di file sharing simile a BitTorrent (in fase di sviluppo).

- Il Network Coding è applicato con notevoli benefici anche al routing multicast.

Pollution Attack

- Problema a cui sono soggette le reti e che produce più disagi in quelle che utilizzano il Network Coding;
- Attacco di tipo *Denial Of Service;*
- Un nodo corrotto o controllato da un avversario modifica un vettore in transito nella rete;
- Un vettore erroneo non è distinguibile da uno corretto;
- La destinazione che riceve un vettore erroneo non può ricostruire il file;

Pollution Attack: soluzioni

- Firma digitale per firmare il file originale (o hash di esso): i nodi intermedi non possono verificare i vettori;
- Firma digitale (o hash) su ogni vettore iniziale: i nodi intermedi, verificati e combinati i vettori, devono creare una nuova firma (o valore hash);

- Schemi basati sulle firme omomorfiche e schemi basati sull'hashing omomorfico

Firme omomorfiche

- Date le firme dei singoli vettori, è possibile calcolare la firma del vettore ottenuto dalla combinazione lineare di questi senza conoscere alcuna chiave.

- Proprietà: sia *Sign* uno schema di firma, per ogni vettore a,b e scalari α,β si ha

$$Sign(\alpha a+\beta b) = Sign(a)^{\alpha} Sign(b)^{\beta}$$

Nsig

- Autori: Gennaro, Katz, Krawczyk, Rabin [GKKR PKC10]
- Sicurezza: Random Oracle model e assunzioni di RSA
- Chiave pubblica ($N,e,g_1,\ldots,g_n$):
  - modulo RSA $N = pq$;
  - $g_1,\ldots,g_n$ elementi opportunamente scelti;
  - un primo e sufficientemente grande;
- Chiave privata *(d):*
  - esponente d : $ed = 1 \bmod \Phi(N)$;
- Valori hash: $h_1 = H(1,fid); \ldots; h_m = H(m,fid);$

Nsig: formule

- Firma: $Nsig(\vec{w}) = \left(\prod h_i^{u_i} \prod g_j^{v_j}\right)^d \bmod N$
- Verifica: $\sigma^e = \prod h_i^{u_i} \prod g_j^{v_j} \bmod N$
- Combinazione: $w = \sum \alpha_i w^{(i)}; \quad \sigma = \prod \sigma_i^{\alpha_i}$

NetPFSig

- Autori: Catalano, Fiore e Warinschi [CFW Eurocrypt'11]
- Sicurezza: dimostrata senza l'uso del *ROM,* si richiede che l'hash sia *division-intractable.*
- Chiave pubblica (A,N,φ_N):

  $A=\{g,g_1,\ldots,g_n,h_1,\ldots,h_m\}$ insieme di simboli variabili;

  $N = pq$, con p, q primi;

  $\varphi_N(fid, V, B^*, I, I_s)$ è una funzione usata per calcolare e, s_i.

NetPFSig: formule

- Firma: $x_i = \left(g^{s_i} \prod_j^m h_j^{u_j^{(i)}} \prod_j^n g_j^{v_j^{(i)}} \right)^d$
- Verifica: $x_i^e = g^{s_i} h_1^{u_1^{(i)}} \ldots h_m^{u_m^{(i)}} g_1^{v_1^{(i)}} \ldots g_n^{v_m^{(i)}}$
- Combinazione: $w = \sum \alpha_i w^{(i)}; \quad \sigma = \prod \sigma_i^{\alpha_i}$

Implementazioni

Sono state implementate delle librerie in C++ per la simulazione di una rete utilizzante gli schemi citati.

In particolare:

- Ncvector: rappresenta i vettori;
- MatrixQ: rappresenta la matrice utilizzata dal nodo destinazione. Implementa il calcolo dell'inversa;
- SourceNode: rappresenta il nodo sorgente;
- IntermediateNode: rappresenta un nodo generico della rete;
- TargetNode: rappresenta il nodo destinazione;

Implementazioni

- Queste classi simulano la rete in quanto operano su file anzichè sulla rete.
- Nell'implementazione, è stata scelta *Skein* come funzione hash.
- Skein è uno dei finalisti nella competizione per la scelta dello standard SHA3

Le classi di firma

Nsig

- Implementa lo schema di firma Nsig.
- I primi *p* e *q* sono scelti come primi-safe.

NetPFSig

- Implementa lo schema di firma NetPFSig.
- *e* è calcolato come *H(fid,1)*||*H(fid,2)* per ottenere un valore a 2048 bit.

Conclusioni

- Lo schema NetPFSig ha tempi di esecuzione peggiori di quelli di Nsig.
- I test e i dati proveniente dalle implementazioni mostrano che i tempi differiscono di un fattore minore di 1,6.
- Queste diapositive, la tesi e le librerie sono disponibili sul sito:

http://danielelicitra.altervista.org

Grazie per l'attenzione