Prevenzione dello Spam su VoIP basata su Web of Trust

Marco Cornolti

Dipartimento di Informatica
Università di Pisa

4 novembre 2009

http://www.cli.di.unipi.it/~cornolti – cornolti@cli.di.unipi.it

Organizzazione del tirocinio alla NEC

- NEC Europe ltd. – Network Research Division, Heidelberg (DE)
- Settembre - Dicembre 2008
- Supervisori: Saverio Niccolini, Nico d'Heureuse, Jan Seedorf (gruppo RTC)
- 577,5 ore lavorative

Lavoro svolto

Prima parte il test della Web of Trust anti-SPIT

- L'algoritmo per la ricerca del Trust Path
- Firma crittografica dei messaggi
- Integrazione in VoIP-SEAL
- GUI per la demo
- Test sulla performance

Seconda parte personalizzazione di VoIP-SEAL

- Design e parziale implementazione del sistema di personalizzazione.

Sulla prima parte del lavoro è stato redatto l'articolo Cornolti et al., *Detecting Trustworthy Real-Time Communications using a Web-of-Trust*, IEEE GLOBECOM 2009 (to appear).

Spam via e-mail e SPIT

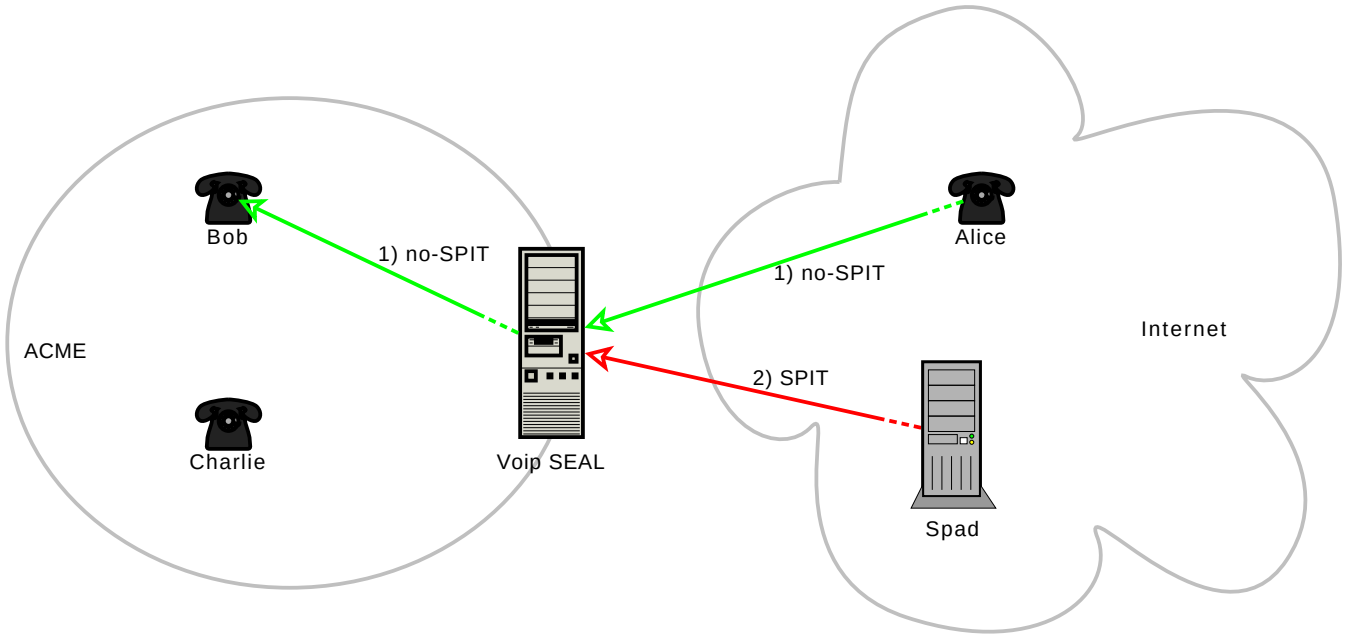
I due tipi di spam hanno caratteristiche comuni:

- Processo automatizzabile, nessun intervento umano
- Richiede solo il lato software, niente hardware particolari
- Costo molto basso, alta velocità (con 512kbit/sec, 40 telefonate contemporanee)
- Facile trovare indirizzi destinatari dello spam

Mentre per altri versi sono molto diversi:

- E-mail è asincona, VoIP è sincrona $\Rightarrow$ impossibile il filtro sui contenuti
- Analisi anti-SPIT deve essere fatta in tempo reale
- SPIT è persino più molesto per i target

Il server VoIP-SEAL

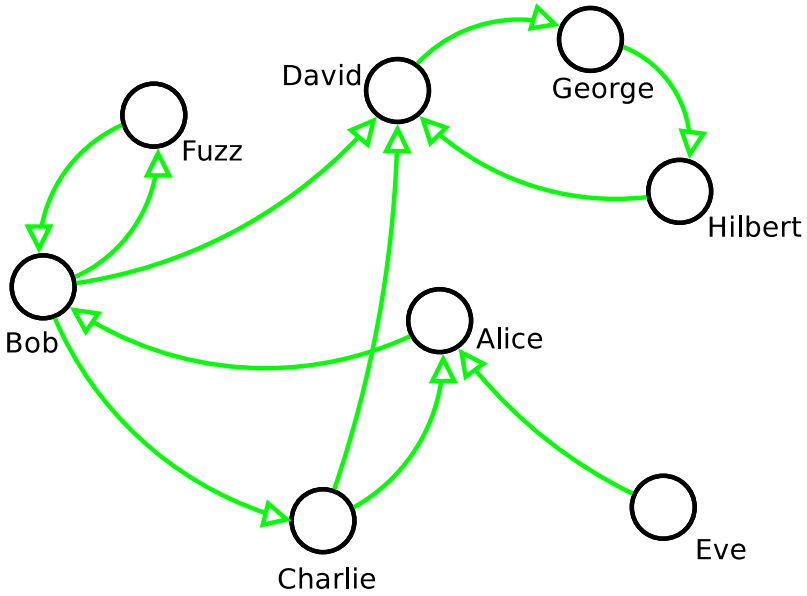


- È un server VoIP che fa anche il controllo anti-SPIT
- Perfettamente integrato nell'architettura SIP

I filtri anti-SPIT di VoIP-SEAL

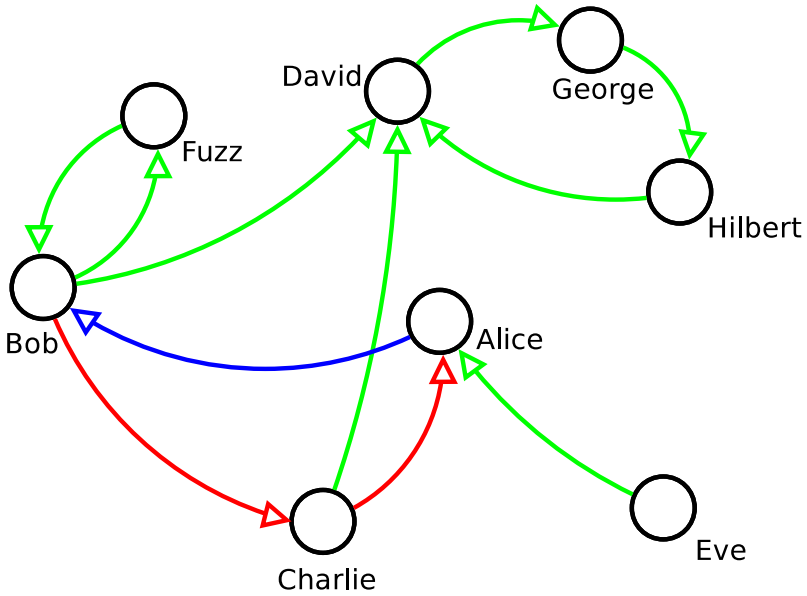
- Ogni modulo implementa un test sulla chiamata in arrivo
- I test restituiscono un punteggio sulla probabilità che la chiamata sia SPIT
- Tra i moduli vale la pena menzionare:
  - Modulo `Turing test`: composizione di numeri
  - Modulo `DDOS`: frequenza delle chiamate in arrivo

La Web of Trust (WoT)



- Concetto mutuato da PGP
- Relazioni di fiducia garantite crittograficamente
- Serve per potersi fidare di chiavi delle quali non si è mai verificata di persona l'identità

Il Trust Path



- Percorso *minimo* da un'identità ad un'altra
- Le relazioni di fiducia non sono simmetriche
- Lunghezza del TP $\sim$ livello di fiducia
- Per una chiamata $A \longrightarrow B$, bisogna cercare il TP $B \longrightarrow A$

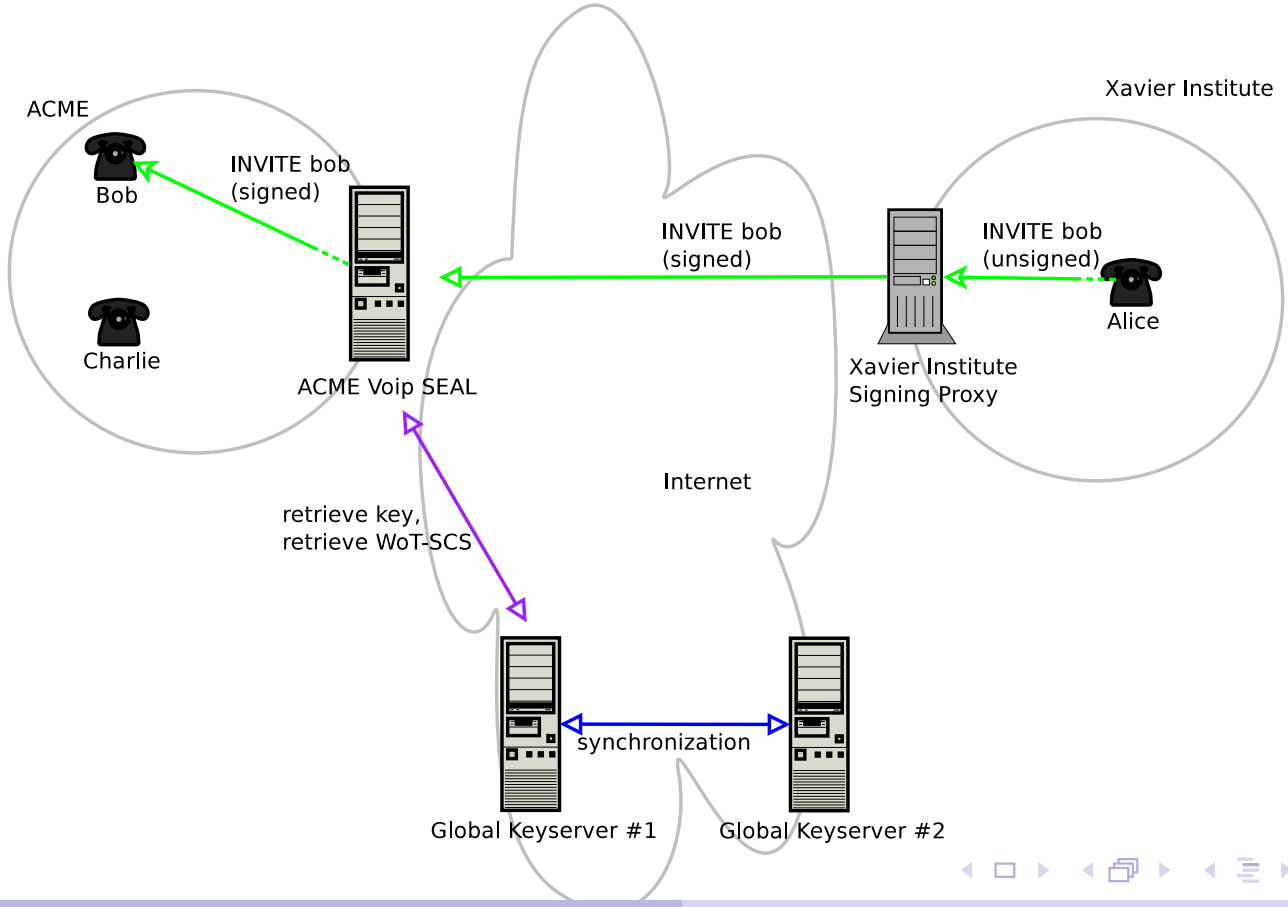
Operazioni del Modulo wot

- Il test della WoT è implementato in VoIP-SEAL tramite il Modulo wot

Operazioni del modulo WoT:

1. Verifica dell'autenticità della firma rispetto al messaggio
2. Eventualmente, download della chiave
3. Ricerca del Trust Path

Deployment del test WoT: W-BASA



Algoritmo di ricerca del Trust Path

- Ricerca del percorso minimo: un problema ampiamente studiato
- Si è cercato un algoritmo che avesse tempi di esecuzione bassi soprattutto per percorsi corti

m: media degli archi uscenti dai nodi (firme)
l: lunghezza del TP

$$BFS \in O(m^l) \text{ in tempo}$$

e, soprattutto:

$$BFS \in O(m^l) \text{ in spazio}$$

Basi di BIDDFS

Punti deboli di BFS

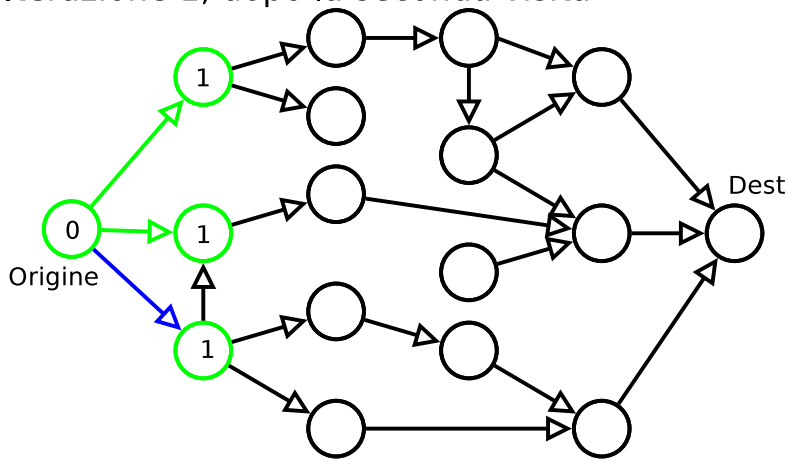
- non sfrutta l'informazione sulla Backward Star, facilmente estraibile dalla WoT
- ha bisogno di molto spazio ausiliario

BIDDFS (Bidirectional Iterative Deepening DFS)

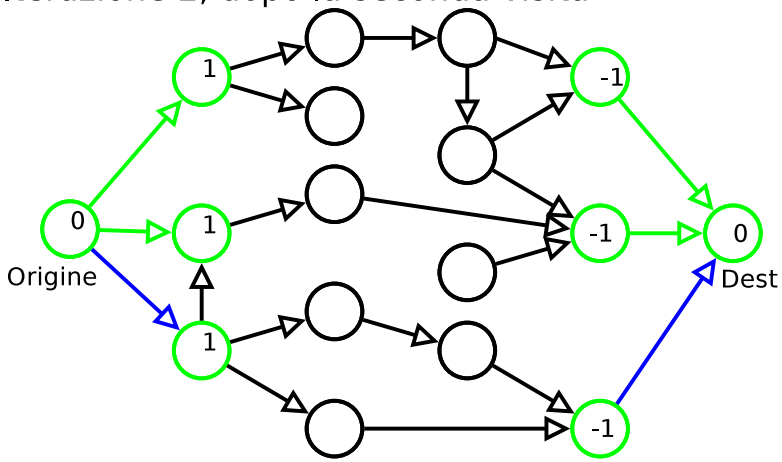
- Conduce delle visite iterative DFS a profondità limitata (in pratica, come BFS ma senza spazio ausiliario)
- Ad ogni iterazione, la profondità viene incrementata
- Due visite, bidirezionali: la prima dal nodo sorgente lungo la FS, la seconda dal nodo destinazione lungo la BS
- Questo riduce il tempo a $BIDDFS \in O(m^{l/2})$
- Cancellazione del grafo non necessaria, si usano timestamp.

BIDDFS: Esempio di ricerca (1/2)

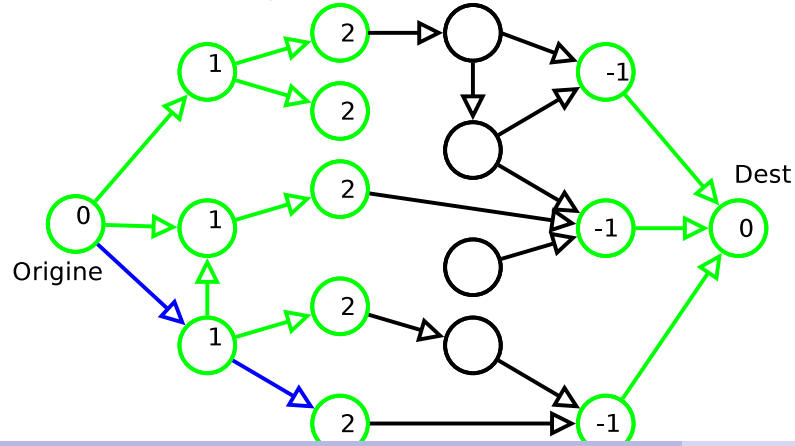
Iterazione 1, dopo la seconda visita



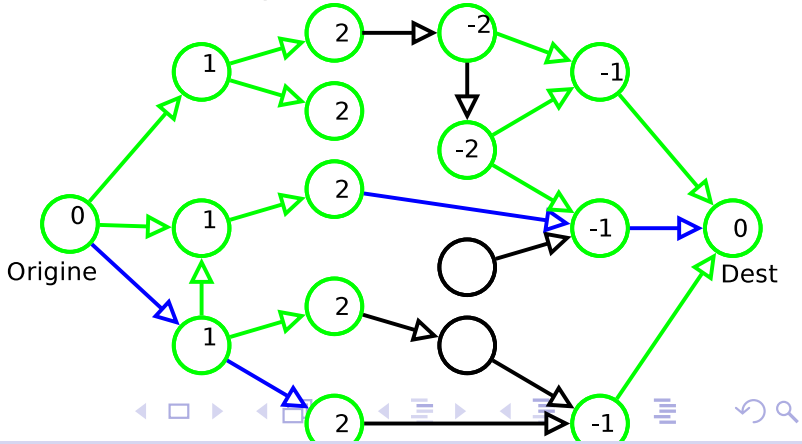
Iterazione 2, dopo la seconda visita



Iterazione 3, dopo la seconda visita

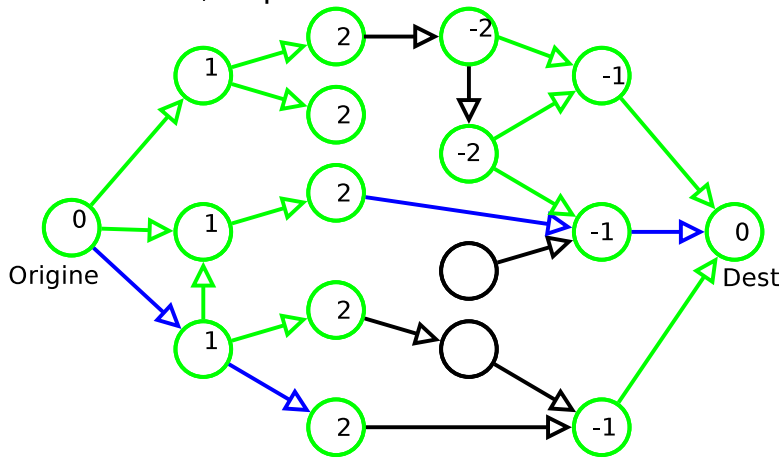


Iterazione 4, dopo la seconda visita

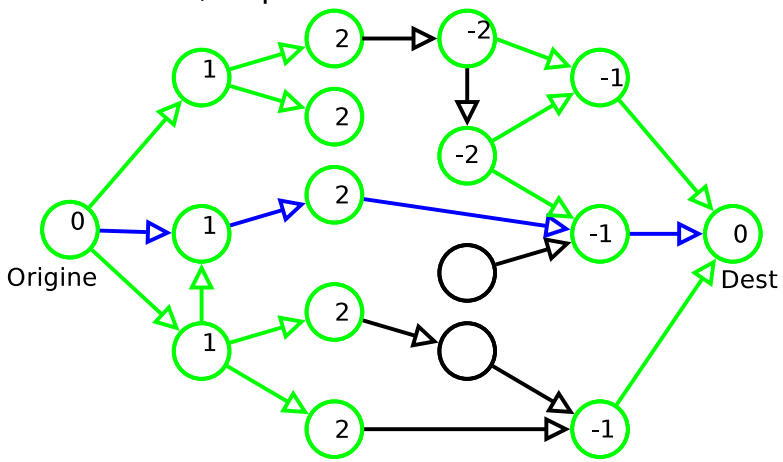


BIDDFS: Esempio di ricerca (2/2)

Iterazione 4, dopo la seconda visita



Iterazione 4, dopo la terza visita



- Ad ogni iterazione i, trova (se esiste) il TP lungo $l = i$.

Complessità in tempo (1/2)

- $BIDDFS \in O(l^{m/2})$
- Ma calcoliamo più esattamente il numero massimo di nodi visitati:

$$v(i,m) = \begin{cases} 2 \cdot \sum_{j=0}^{i/2}(m^j) & \text{per } i \text{ pari} \\ \sum_{j=0}^{\lfloor i/2 \rfloor}(m^j) + \sum_{j=0}^{\lceil i/2 \rceil}(m^j) & \text{per } i \text{ dispari} \end{cases}$$

$$V(l,m) = \begin{cases} \sum_{i=0}^{l} v(i) + \sum_{j=0}^{l/2}(m^j) & \text{per } l \text{ pari} \\ \sum_{i=0}^{l} v(i) + \sum_{j=0}^{\lceil l/2 \rceil}(m^j) & \text{per } l \text{ dispari} \end{cases}$$

Complessità in tempo (2/2)

Posto $m = 10$

l	$V(l, 10)$: n. di nodi visitati
1	25
2	47
3	269
4	491
5	2.713
6	4.935
7	27.157

Complessità in spazio

- Stack della funzione di visita, dimensione $O(s \cdot l) = O(k)$
- Strutture dati ausiliarie, dimensione $O(k)$
- $BIDDFS \in O(k)$

I test sulla performance

- Requisito: l'analisi deve avvenire in tempo reale
- Considerati i 7 scenari possibili
- Esperimenti condotti su un PC Athlon 2800 XP, 2 GB di RAM a 333MHz, sistema operativo Linux 2.6, Internet T3
- Vengono presentati qui solo i risultati fondamentali
  - Download di una chiave: 100msec, $\sigma_r = 53\%$
  - Verifica: 3.8 msec, $\sigma_r = 24\%$
  - Ricerca del TP: tempi analoghi alla verifica.
  - Nel caso più lungo, sono processabili in media 150 nuove chiamate/sec.
  - I requisiti di tempo reale sono ampiamente rispettati

Lavoro Futuro

- Studi più approfonditi sul comportamento di BIDDFS all'aumentare della dimensione della WoT
- Studi sull'occupazione della memoria all'aumentare della WoT
- Processo presso IETF per la standardizzazione di W-BASA