Usi della crittografia: controllo d'integrità

Lezione 6 di Sicurezza dei sistemi informatici 1

Docente: Giuseppe Scollo

Università di Catania, sede di Comiso (RG)
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Studi in Informatica applicata, AA 2008-9

Indice

1. Usi della crittografia: controllo d'integrità
2. codici di controllo dell'integrità
3. hash: protezione dell'integrità
4. proprietà di funzioni di hash MDC
5. costruzioni iterative di funzioni di hash
6. codici di autenticità di messaggi (MAC)
7. algoritmi standard di hash

codici di controllo dell'integrità

problema: nella comunicazione attraverso un mezzo trasmissivo, come pure nella memorizzazione in un supporto fisico, una sequenza binaria è suscettibile di **alterazioni**, accidentali o meno, che possono corrompere l'informazione rappresentata

soluzione: estendere la sequenza con **informazione di controllo** di errore, efficace per la **rivelazione** di errori (entro limiti), e per la loro **correzione** (in limiti più ristretti), *e.g.:*

- **bit di parità**, aggiunto alla codifica ASCII dei caratteri
- **byte di controllo**, per sequenze di più byte; **somma di controllo** (***checksum***)
- **codici a ridondanza ciclica** (ingl.: ***Cyclic Redundancy Code*, CRC**)
- funzioni di ***hash*** crittografico (per la protezione da alterazioni intenzionali)

per configurazioni binarie di uguale lunghezza, **distanza di *Hamming* :**

- di **due configurazioni binarie:** numero di bit in cui differiscono
- di un **insieme finito di configurazioni binarie:**
  minima distanza di Hamming fra due configurazioni nell'insieme
- di un **codice** $c : A \rightarrow 2^n$ **:** distanza di Hamming di $c(A)$

un codice con distanza di Hamming **2n+1** può **rivelare** fino a **2n** errori, e ne può **correggere** fino a **n**, nella trasmissione o memorizzazione della codifica di un simbolo

hash: protezione dell'integrità

la crittografia è utile a garantire che ad un testo non siano state apportate **alterazioni indebite**, ovvero a rilevare il contrario, con i seguenti vantaggi rispetto ai codici di Hamming:

- ridondanza limitata, indipendente dalla lunghezza del testo
- rivelazione di alterazioni di qualsiasi entità, piccola o grande che sia
- margini di errore molto bassi

un **codice di *hash*** crittografico è una funzione del testo che gode di una **proprietà fondamentale:**

è molto improbabile che un'alterazione del testo ne dia uno con lo stesso hash

altre proprietà caratterizzano classi specifiche di funzioni di hash (v. appresso)

le funzioni di hash di solito codificano il testo, di lunghezza arbitraria, in un "riassunto" (*digest, checksum, hash*) di lunghezza fissa:

- si adopera spesso a tal scopo la **crittografia concatenata a blocchi**
  ovvero si frammenta il testo in blocchi della lunghezza richiesta e si adopera una crittografia a blocchi di lunghezza fissa (quale DES, AES, o altre più semplici) iterativamente, combinando (e.g. con XOR) ciascun blocco con l'output della crittografia concatenata dei blocchi precedenti

proprietà di funzioni di hash MDC

l'acronimo sta per *Manipulation Detection Code*

a differenza dei CRC, i codici MDC proteggono da **attacchi intenzionali** all'integrità

l'output $h(x)$ di una funzione di hash dev'essere **di facile calcolo**

l'output di una funzione MDC è di lunghezza fissa, per input di lunghezza arbitraria:
proprietà di **compressione** → sono inevitabili le **collisioni:** $x \neq y$, $h(x) = h(y)$

vulnerabilità: il "paradosso del compleanno"

una funzione di hash MDC generalmente gode di una o più delle seguenti **proprietà:**

- **(P1) unidirezionalità (*one-way*):** intrattabilità del calcolo di x, dato $h(x)$
- **(P2) resistenza debole alle collisioni:**
  dati x e $h(x)$, intrattabilità del reperimento di un $y \neq x$ tale che $h(x) = h(y)$
- **(P3) resistenza (forte) alle collisioni:**
  intrattabilità del reperimento di una qualsiasi coppia $x \neq y$ tale che $h(x) = h(y)$

costruzioni iterative di funzioni di hash

una semplice tecnica per la costruzione iterativa di funzioni di hash, detta di Merkle-Damgård, si basa su una **funzione di compressione** f che, applicata alla concatenazione di due sequenze binarie di lunghezze prefissate, produce un output di lunghezza uguale a quella di una delle due sequenze in input

l'iterazione si avvia con un **valore iniziale** h_0 prefissato dell'hash, di lunghezza uguale a quella prefissata dell'output, e con una suddivisione dell'input in blocchi aventi tutti la stessa lunghezza (generalmente diversa da quella dell'output)

> N.B.: l'ultimo blocco viene esteso a tal fine, e l'estensione può contenere l'informazione di lunghezza dell'originale

la costruzione iterativa è quindi così definita, dove ‖ designa concatenazione:

$$h_i = f(h_{i-1} \| x_i)$$

altre costruzioni iterative adoperano diversi operatori di combinazione, specie nel caso di crittografia con chiave, di cui appresso

codici di autenticità di messaggi (MAC)

la garanzia di integrità di un messaggio non ne assicura l'**autenticità** dell'**origine**

- la facilità di calcolo della funzione di hash costituisce una vulnerabilità alla trasmissione di messaggi di origine contraffatta

i codici MAC sono progettati per rimuovere tale vulnerabilità, proteggendo integrità e autenticità, mediante una **chiave segreta** usata per la generazione dell'output di hash:

- condivisa da mittente e destinatario
- parametro della **funzione di hash con chiave**, ovvero indice k di un'ordinaria funzione di hash h_k in una famiglia di tali funzioni

per essere efficace, un MAC deve soddisfare l'ulteriore proprietà di **resistenza al calcolo:**

- intrattabilità del calcolo di $h_k(x)$ da un insieme di coppie $(x_i, h_k(x_i))$, se non è nota k

si derivano algoritmi MAC da algoritmi MDC mediante varie costruzioni, e.g. HMAC, v. fonti HMAC per approfondimenti

algoritmi standard di hash

funzioni di hash **più in uso:**

- MD5 (hash da 128 bit)
- SHA-1 (hash da 160 bit)

entrambe ottenute mediante costruzione Merkle-Damgård

nel 2005 è stato dimostrato che né MD5 né SHA-1 posseggono la resistenza alle collisioni (proprietà P3)

SHA-1 è adoperato in vari protocolli standard di sicurezza: TLS, SSL, PGP, SSH, S/MIME, IPsec

le varianti SHA-2 sono algoritmicamente simili a SHA-1, quindi il NIST ha pubblicato un bando per la determinazione di un nuovo standard SHA-3, secondo modalità simili a quelle che hanno condotto alla selezione di AES

- **l'accettazione di proposte si è chiusa il 31/10/2008, e la prima selezione ne ha ammesse 51, in una lista pubblicata il 9/12/2008; questi algoritmi sono stati pubblicamente presentati ad una conferenza organizzata dal NIST nel febbraio 2009; la conclusione della selezione e la pubblicazione dello standard sono previste per il 2012**