Funzioni hash: SHA

Paolo D'Arco
pdarco@unisa.it

Universitá di Salerno

Elementi di Crittografia

Contenuti

Funzioni hash usate in pratica

$$H : \{0,1\}^* \rightarrow \{0,1\}^\ell \qquad \text{collision resistant}$$

Deve essere impraticabile trovare $x, x' : H(x) = H(x')$

Output di ℓ bit $\Rightarrow$ il meglio che possiamo sperare di ottenere é che risulti impraticabile trovare collisioni usando meno di $2^{\ell/2}$ invocazioni di H.

Generalmente costruite in due passi:

1. una funzione di compressione per input di lunghezza fissata viene progettata
2. la funzione viene estesa a domini arbitrari, per esempio usando la trasformazione di Merkle-Damgard

Come costruire una buona funzione di compressione?

Davies-Meyer construction

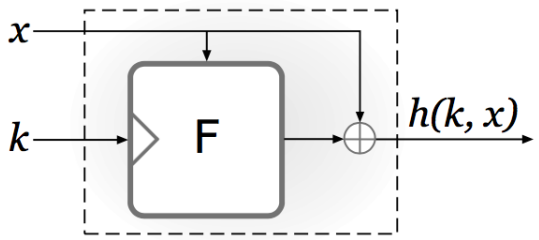
Un modo di costruire una funzione di compressione collision resistant da **un cifrario a blocchi** che soddisfa proprietá aggiuntive.

Sia

$$F : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^\ell$$

un cifrario a blocchi con lungh. chiave n e lungh. blocco ℓ. Definiamo

$$h : \{0,1\}^{n+\ell} \to \{0,1\}^\ell \qquad \text{come} \qquad h(k,x) \stackrel{def}{=} F_k(x) \oplus x.$$



Davies-Meyer construction

Non sappiamo come provare che h é collision resistant assumendo che F é una SPRP

$\Rightarrow$ potrebbe anche non essere possibile provarlo

Possiamo provare che é resistente a collisioni nel **Modello del cifrario ideale**

Ideal cipher model

Ideal cipher model (ICM in breve): tutte le parti hanno accesso ad un oracolo per F ed F^{-1}, permutazione con chiave totalmente casuale $\forall k, x$, risulta $F^{-1}(k, F(k, x)) = (k, x)$.

Le query sono l'unico modo per calcolare $F(k, x)$ ed $F^{-1}(k, y)$.

Si noti che:

- é un rafforzamento del random oracle model
- non é possibile tener conto di attacchi che sfruttano dipendenze di chiavi

  $F(K, \cdot)$ ed $F(K', \cdot)$ si comportano in modo indipendente anche se K e K' sono fortemente relate l'una all'altra

Davies-Meyer construction

- non ci sono chiavi deboli
- F_K dovrebbe comportarsi casualmente anche se K é nota

Qualsiasi cifrario reale F non é detto che soddisfi queste proprietá.

Tuttavia, é possibile dimostrare il seguente risultato

Teorema 6.5. Se F viene modellata come un cifrario ideale, allora la costruzione di Davies-Meyer dá una funzione di compressione resistente a collisioni. Ogni Adv che effettua $q < 2^{\ell/2}$ query all'oracolo, trova una collisione con probabilitá al piú $q^2/2^{\ell}$.

Dim. Si consulti il libro di testo.

Nota: é stato mostrato che ROM e ICM sono equivalenti, in accordo ad una nozione che estende la nozione di indistinguibilitá e si chiama *indifferenziabilitá*.

MD5

Funzione hash con output di 128 bit

- Progettata nel 1991 da R. Rivest
- Nel 1993 (Der Boer e Bosselaers) pseudo-collisione e nel 1996 (Dobbertin) prima collisione
- Nel 2004 un team cinese (Wang e altri) di crittoanalisti ha presentato un metodo efficiente per trovare collisioni
- L'attacco é stato migliorato diverse volte successivamente e oggi bastano pochi minuti su PC
- Possono essere anche trovate collisioni significative
- Non dovrebbe essere piú usata

Secure hash algorithm SHA

SHA-0, SHA-1 e SHA-2, standardizzate dal NIST

SHA-1

- Introdotta nel 1995. Ha 160 bit di output
- Nel Febbraio 2017 é stata trovata una collisione esplicita in circa 2^{63} passi. Non dovrebbe essere piú usata.

SHA-2 (due versioni)

- SHA-256 con 256 bit di output
- SHA-512 con 512 bit di output

Secure hash algorithm SHA

Tutte le funzioni SHA (SHA-1, SHA-256 e SHA-512) sono realizzate utilizzando lo stesso schema di progettazione

- viene progettata una funzione di compressione utilizzando la costruzione di Davies-Meyer ad un cifrario a blocchi
- il dominio viene esteso usando la trasformazione di Merkle-Damgard.
- il cifrario a blocchi, identificato retroattivamente nell'analisi della costruzione e denotato con i nomi SHACAL-1 (per SHA-1) e SHACAL-2 (per SHA-2), **non** é usato per la cifratura ma progettato esplicitamente per la costruzione

Il NIST ha annunciato una nuova competizione pubblica nel 2007 per una nuova funzione

Il 5 agosto del 2015 la funzione SHA-3 é stata rilasciata come ultima della famiglia SHA

SHA-3 (Keccak)

É completamente differente da tutti gli algoritmi SHA precedenti. Supporta output di 256 e 512 bit. SHA-3:

- utilizza una permutazione f senza chiave con una lunghezza di blocco molto grande, i.e., 1600 bit
- non utilizza la trasformazione di Merkle-Damgard
- sfrutta un approccio nuovo, denominato **sponge construction**, per gestire input di lunghezza arbitraria.
- puó essere analizzata nel **modello della permutazione casuale**

Nota: il modello della permutazione causale é un modello piú debole dell'ideal cipher model (fissando k nel secondo si ottiene il primo)

Sponge construction

Opera in due fasi: assorbimento (sponge) e spremitura (squeeze).

Nella prima, l'input $P = P_0 \dots P_{n-1}$, viene assorbito nello stato, r bit alla volta.

Nella seconda, l'output Z viene prodotto a seguito di trasformazioni di stato, r bit alla volta.