CRITTOGRAFIA

- Identificazione
- Autenticazione
- Firma digitale

Identificazione, autenticazione e firma digitale

Identificazione:

Un sistema di elaborazione, isolato o in rete, deve essere in grado di accertare l'identità di un utente che richiede di accedere ai suo i servizi.

Autenticazione:

Il destinatario di un messaggio deve essere in grado di accertare

- l'identità del mittente
- l'integrità del crittogramma ricevuto.

Identificazione, autenticazione e firma digitale.

Firma digitale

1. MITT non deve poter negare di aver inviato un messaggio m.
2. DEST deve essere in grado di autenticare il messaggio
3. DEST non deve poter sostenere che m' ≠ m è il messaggio inviato da MITT.

Tutto deve essere verificabile da una terza parte.

Relazioni tra le funzionalità

Non sono indipendenti, ma ciascuna estende le precedenti

- L'autenticazione di un messaggio garantisce l'identificazione del mittente.
- L'apposizione della firma garantisce l'autenticazione del messaggio.

Ogni funzionalità è utilizzata per contrastare gli attacchi attivi.

Esistono realizzazioni algoritmiche basate sui **cifrari asimmetrici** e simmetrici.

Funzioni hash

Una funzione hash $f: X \to Y$ è una funzione tale che

$$n = |X| \gg m = |Y|$$

$\exists\ X_1, X_2, \ldots, X_m \subseteq X$ disgiunti t.c.

$$X = X_1 \cup X_2 \cup \ldots \cup X_m$$

$$\forall\ i,\ \forall\ x \in X_i\ ,\quad f(x) = y$$

Una **buona** funzione hash deve assicurare che

I sottoinsiemi $X_1, \ldots, X_m$ abbiano circa la stessa cardinalità

due elementi estratti a caso da X hanno probabilità circa 1/m di avere la stessa immagine in Y

Elementi di X molto "simili" tra loro appartengano a due sottoinsiemi diversi

se X è un insieme di interi, due elementi con valori prossimi devono avere immagini diverse

Gestione delle collisioni

L'algoritmo che impiega la funzione hash dovrà affrontare la situazione in cui più elementi di X hanno la stessa immagine in Y.

Funzioni hash one-way

Se la funzione è applicata in crittografia, deve soddisfare le seguenti proprietà:

1. per ogni $x \in X$ è computazionalmente facile calcolare
$$y = f(x)$$
2. Proprietà one-way: per la maggior parte degli $y \in Y$ è computazionalmente difficile determinare $x \in X$ tale che
$$f(x) = y, \quad \text{i.e., } x = f^{-1}(y)$$
3. Proprietà claw-free: è computazionalmente difficile determinare una coppia di elementi x_1, x_2 in X tali che
$$f(x_1) = f(x_2)$$

Funzioni hash usate in crittografia: MD5 (Message Digest, versione 5)

- Si tratta di una famiglia di algoritmi, quello originale non fu mai pubblicato. Si pubblicarono MD2, seguito da MD4.
- In MD2 e MD4 furono trovate debolezze, e Ron Rivest propose MD5, nel 1992.
- Riceve in input una sequenza S di 512 bit e produce un'immagine di 128 bit: la sequenza è *digerita* riducendone la lunghezza ad un quarto.
- E' stato in seguito dimostrato che MD5 non resiste alle collisioni, e nel 2004 si sono individuate delle debolezze serie.
- Oggi la sua sicurezza si considera severamente compromessa.
- Lo stesso Rivest ha affermato (2005) che MD5 era da considerarsi chiaramente forzata dal punto di vista della resistenza alle collisioni.

Funzioni hash usate in crittografia: RIPEMD-160

- Versione "matura" delle funzioni MD
- Nata nel 1995 nell'ambito di un progetto dell'Unione Europea
- Produce immagini di 160 bit ed è esente dai difetti di MD5

Funzioni hash usate in crittografia: SHA Secure Hash Algorithm

- Progettata da NIST e NSA nel 1993, si adotta quando la proprietà di claw-free è cruciale per la sicurezza del sistema
- Opera su sequenze lunghe fino a 2^{64} bit e produce immagini di 160 bit
- È una funzione crittograficamente sicura: soddisfa i requisiti delle funzioni hash one-way, e genera immagini molto diverse per sequenze molto simili.
- La prima versione pubblicata (SHA-0) conteneva una debolezza, scoperta in seguito da NSA, che portò a una revisione dello standard.

Funzioni hash usate in crittografia: SHA Secure Hash Algorithm

- 1993

  SHA-0: proposta dal NIST nel 1993, presto ritirata a causa di una debolezza interna
- 1995

  SHA-1: progettato da NSA, uso raccomandato dal NIST
- 2001

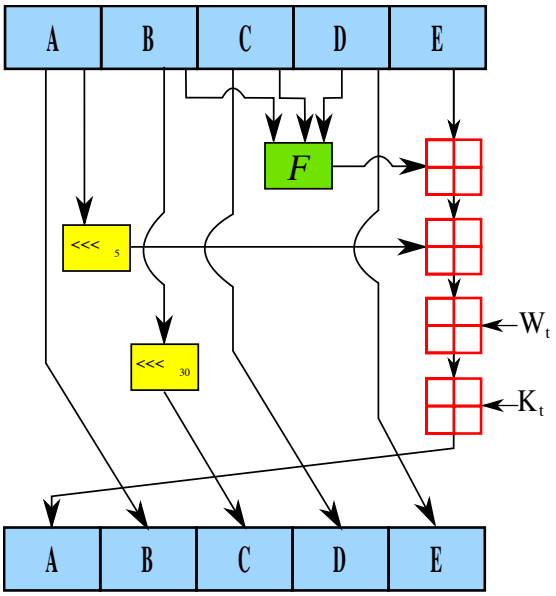
  SHA-2: quattro funzioni della famiglia SHA, progettate da NSA e pubblicate dal NIST, caratterizzate da digest più lunghi
- 2007

  A causa degli attacchi su MD5 e SHA-0, e di attacchi teorici su SHA-1, il NIST ha sollecitato proposte per un nuovo algoritmi hash. Ci sono stati 63 candidati.
- 2012

  Si è conclusa la valutazione, ed è stato selezionato una funzione hash, di progettazione non governativa → SHA-3 (team di analisti italiani e belgi, rilasciata ufficialmente nel 2015).

Funzioni hash usate in crittografia: SHA-1

- Opera su sequenze lunghe fino a 2^{64}-1 bit, e produce immagini di 160 bit.
- E' molto usata nei protocolli crittografici anche se non è più certificata come standard
- Tutte le altre funzioni hanno una struttura molto simile a SHA-1
- Opera su blocchi di 160 bit, contenuti in un buffer di 5 registri di 32 bit ciascuno, in cui sono caricati inizialmente dei valori pubblici
- Il messaggio *m* viene concatenato con una sequenza di padding che ne rende la lunghezza multipla di 512 bit
- Il contenuto dei registri varia nel corso dei cicli successivi in cui questi valori si combinano tra loro e con blocchi di 32 bit provenienti da *m*
- Alla fine del procedimento, i registri contengono SHA-1(m).



Un'iterazione all'interno della funzione di compressione di SHA-1. A, B, C, D ed E sono parole di stato a 32 bit; F è una funzione non lineare che varia; $\lll_n$ denota una rotazione del bit di sinistra di n posti; n varia per ogni operazione. $\boxplus$ denota l'addizione modulo 2^{32}. K_t è una costante.

W_t blocco di 32 bit ottenuto tagliando e rimescolando i blocchi di messaggio

Il contenuto dei registri varia nel corso dei cicli (all'inizio sono caricati valori fissi e pubblici) in cui questi valori si combinano tra loro e con blocchi di 32 bit provenienti dal messaggio W, nonché con alcuni parametri relativi al ciclo.
Alla fine del procedimento (quando è stato letto l'intero messaggio) i registri contengono l'hash SHA1(W)

Identificazione su canali sicuri

ESEMPIO: accesso di un utente alla propria casella di posta elettronica, o a file personali memorizzati su un calcolatore ad accesso riservato ai membri della sua organizzazione.

- l'utente inizia il collegamento inviando in chiaro login e password
- se il canale è protetto in lettura e scrittura, un attacco può essere sferrato solo da un utente locale al sistema:
  - ad esempio l'amministratore che ha accesso a tutti i file memorizzati (oppure un hacker)
- Il meccanismo di identificazione prevede una cifratura delle password, realizzata con funzioni hash one-way.

Cifratura password nei sistemi UNIX

Quando un utente U fornisce per la prima volta una password P

il sistema associa a U due sequenze binarie (che memorizza nel file delle password al posto di P):

- S (seme)

  prodotta da un generatore pseudocasuale

- Q = h (PS)

  h: funzione hash one-way

Cifratura password nei sistemi UNIX

Ad ogni successiva connessione di U, il sistema:

- recupera S dal file delle password,
- concatena S con la password fornita da U
- calcola l'immagine one-way della nuova sequenza: h(PS)
- se h(PS) = Q l'identificazione ha successo.

Un accesso illecito al file delle password non fornisce informazioni interessanti:

è computazionalmente difficile ricavare la password originale dalla sua immagine one-way

Protezione del canale

Se il canale è insicuro, la password può essere intercettata durante la sua trasmissione in chiaro.

Il sistema non dovrebbe mai maneggiare direttamente la password, ma una sua immagine inattaccabile.

Canale insicuro: identificazione

$< e, n >, < d >$ = chiavi pubblica e privata di un utente U che richiede l'accesso ai servizi offerti dal sistema S.

1. S genera un numero casuale $r < n$ e lo invia in chiaro a U.
2. U calcola
   $$f = r^d \bmod n \quad \text{(firma di U su r)}$$
   con la sua chiave privata e lo spedisce a S.
3. S verifica la correttezza del valore ricevuto calcolando e verificando se
   $$f^e \bmod n = r$$
   Se ciò avviene, l'identificazione ha successo

Canale insicuro: identificazione

- Le operazioni di cifratura e decifrazione sono invertite rispetto all'impiego standard nell'RSA
- Possibile perché le due operazioni sono commutative

$$(x^{e} \bmod n)^{d} \bmod n = (x^{d} \bmod n)^{e} \bmod n \ (=x)$$

- f può essere generata solo da U che possiede < d >
- Se il passo 3 va a buon fine, il sistema ha la garanzia che l'utente che ha richiesto l'identificazione sia effettivamente U, anche se il canale è insicuro

Canale insicuro: identificazione

Problema:

S chiede a U di applicare la sua chiave privata a una sequenza r che S stesso ha generato

potrebbe essere stata scelta di proposito per ricavare qualche informazione sulla chiave privata di U.

Protocollo alternativo a "conoscenza zero"

impedisce che da una comunicazione si possa estrarre più di quanto sia nelle intenzioni del comunicatore

Canale insicuro: autenticazione

DEST deve autenticare il messaggio accertando l'identità di MITT e l'integrità di m

MITT e DEST concordano una chiave segreta k.

Canale insicuro: autenticazione

MITT

- allega al messaggio un MAC (Message Authentication Code) A(m, k), allo scopo di garantire la provenienza e l'integrità del messaggio.
- spedisce la coppia < m, A(m, k) > in chiaro,
- oppure, cifra m e spedisce < C (m, k'), A(m, k) >
  - C: funzione di cifratura,
  - k': chiave pubblica o segreta del cifrario scelto.

Canale insicuro: autenticazione

DEST

- entra in possesso di m (dopo averlo eventualmente decifrato)
- essendo a conoscenza di A e k, calcola A(m, k)
- confronta il valore ottenuto con quello inviato da MITT per verificare che il MAC ricevuto corrisponda al messaggio a cui risulta allegato:
  - Se la verifica ha successo il messaggio è autenticato.
  - Altrimenti DEST scarta il messaggio.

Canale insicuro: autenticazione

MAC

- È un'immagine breve del messaggio, che può essere stata generata solo da un mittente conosciuto dal destinatario, previ opportuni accordi.
- Ne sono state proposte varie realizzazioni, basate su cifrari asimmetrici, simmetrici e sulle funzioni hash one-way

MAC con funzioni hash one-way

A(m, k) = h(mk), con h funzione hash one way

Risulta computazionalmente difficile per un crittoanalista scoprire la chiave segreta k

- h è nota a tutti, e m può viaggiare in chiaro o essere scoperto per altra via, ma k viaggia all'interno del MAC
- per recuperare k si dovrebbe invertire h

Il crittoanalista non può sostituire facilmente il messaggio m con un altro messaggio m'

- dovrebbe allegare alla comunicazione di m' il MAC A(m', k) che può produrre solo conoscendo k.

CBC + MAC

- Usando un cifrario a blocchi in modalità CBC, si può usare il blocco finale del crittogramma come MAC
- Il blocco finale è infatti funzione dell'intero messaggio

Firma manuale

1. è autentica e non falsificabile
   prova che chi l'ha prodotta è chi ha sottoscritto il documento;
2. non è riutilizzabile
   è legata strettamente al documento su cui è stata apposta;
3. il documento firmato non è alterabile
   chi ha prodotto la firma è sicuro che questa si riferirà solo al documento sottoscritto nella sua forma originale;
4. non può essere ripudiata da chi l'ha apposta
   costituisce prova legale di un accordo o dichiarazione.

Firma digitale

- Non può consistere semplicemente di una digitalizzazione del documento originale firmato manualmente
  un crittoanalista potrebbe "tagliare" dal documento digitale la parte contenente la firma e "copiarla" su un altro documento.
- Deve avere una forma che dipenda dal documento su cui viene apposta, per essere inscindibile da questo.
- Per progettare firme digitali si possono usare sia i cifrari simmetrici che quelli asimmetrici.