Cryptography 2

Matteo ROSSI
Politecnico di Torino

https://cybersecnatlab.it

License & Disclaimer

OLICYBER

CYBERSECURITY NATIONAL LABORATORY

Obiettivi

- Comprensione del problema dello scambio delle chiavi
- Comprensione dei concetti base di teoria dei numeri
- Comprensione base del funzionamento degli schemi Diffie-Hellman e RSA
- Comprensione base dei concetti di funzione di Hash, MAC e firma digitale

Prerequisiti

- Encoding e conversioni
- Matematica di base
- Modulo CR_1 – Cryptography 1

Argomenti

- Problema dello scambio delle chiavi
- Cenni di teoria dei numeri
- Problemi facili e problemi difficili
- Scambio di chiavi Diffie-Hellman
- Crittografia a chiave pubblica e RSA
- Integrità, Autenticazione e Non-ripudio

Argomenti

- Problema dello scambio delle chiavi
- Cenni di teoria dei numeri
- Problemi facili e problemi difficili
- Scambio di chiavi Diffie-Hellman
- Crittografia a chiave pubblica e RSA
- Integrità, Autenticazione e Non-ripudio

Problema dello scambio delle chiavi

- Ora che sappiamo come cifrare dei messaggi, la domanda naturale è: come facciamo a trasmettere le chiavi?

Problema dello scambio delle chiavi

- Ora che sappiamo come cifrare dei messaggi, la domanda naturale è: come facciamo a trasmettere le chiavi?

- Idea: ogni coppia di persone si scambia fisicamente una chiave e la utilizza per ogni comunicazione

Trusted Third Party

- Idea migliore: utilizzare un Trusted Third Party (TTP)
  - Un server centrale che ha una chiave condivisa con ogni persona

Trusted Third Party

- Idea migliore: utilizzare un Trusted Third Party (TTP)
  - Un server centrale che ha una chiave condivisa con ogni persona
  - Quando Alice e Bob vogliono comunicare, il server genera una chiave, che manda ad Alice e Bob cifrata con le rispettive chiavi condivise

Trusted Third Party

- Idea migliore: utilizzare un Trusted Third Party (TTP)
  - Un server centrale che ha una chiave condivisa con ogni persona
  - Quando Alice e Bob vogliono comunicare, il server genera una chiave, che manda ad Alice e Bob cifrata con le rispettive chiavi condivise
  - Alice e Bob iniziano a comunicare senza più passare dal server

Trusted Third Party

- Un TTP ha senso in ambienti chiusi (università o aziende)
- Se il TTP smette di funzionare nessuno può più comunicare
- Se il TTP viene attaccato, tutte le chiavi vengono esposte

Scambio delle chiavi

- Possiamo allora fare uno scambio di chiavi sicuro e "online"?

Scambio delle chiavi

- Possiamo allora fare uno scambio di chiavi sicuro e "online"?
  - Merkle Puzzles
  - Diffie-Hellman
  - RSA

Merkle Puzzles

- Protocollo ideato da Ralph Merkle nel 1974
- Basato sull'utilizzo dei block cipher per scambiare le chiavi
- Di difficile utilizzo pratico

Merkle Puzzles

- Idea:
  - Alice e Bob si accordano su un cifrario a blocchi
  - Alice manda una serie di "puzzle" a Bob
  - Bob ne sceglie uno, e manda ad Alice una parte della soluzione
  - La parte della soluzione non mandata da Bob è la chiave condivisa
  - Un attaccante deve risolvere tutti i puzzles per trovare la chiave che i due hanno scelto

Merkle Puzzles – Esempio

- Esempio:
  - Alice e Bob scelgono di usare AES-128
  - Alice sceglie 2^{32} chiavi del tipo $0 \ldots 0||P_i$ con P_i di 32 bit, e cifra 2^{32} messaggi del tipo "Soluzione $x_i||y_i$", con x_i e y_i interi random di 128 bit
  - Bob sceglie uno dei messaggi e prova tutte le possibili chiavi (2^{32} valori di P_i)
  - Quando trova un messaggio che inizia per "Soluzione" manda indietro x_i e Alice saprà cha y_i sarà la chiave condivisa

Merkle Puzzles – Esempio

- Esempio:
  - Un attaccante che vede tutti i puzzle e la risposta di Bob non sa quale di questi ha risolto
  - Deve provare tutti i puzzle: 2^{64} tentativi

Merkle Puzzles – Problemi

- Problemi:
  - Scambiarsi una chiave è troppo costoso
  - Il "gap" tra 2^{32} e 2^{64} è quadratico, vogliamo qualcosa di meglio: un gap esponenziale (es. n e 2^n)

Argomenti

- Problema dello scambio delle chiavi
- Cenni di teoria dei numeri
- Problemi facili e problemi difficili
- Scambio di chiavi Diffie-Hellman
- Crittografia a chiave pubblica e RSA
- Integrità, Autenticazione e Non-ripudio

Congruenze

- Dati tre interi a, b, n diciamo che a è *congruo* a b *modulo* n ($a \equiv b \bmod n$) se (equivalentemente):
  - $a - b$ è divisibile per n
  - a e b danno lo stesso resto se divisi per n

Congruenze – Esempi

- $32 \equiv 7 \bmod 5$
  - $32 - 7 = 25$ divisibile per 5 ($25/5 = 5$ resto 0)
  - $32 / 5 = 6$ resto 2 e $7 / 5 = 1$ resto 2

Congruenze – Esempi

- $32 \equiv 7 \bmod 5$
  - $32 - 7 = 25$ divisibile per 5 ($25/5 = 5$ resto 0)
  - $32 / 5 = 6$ resto 2 e $7 / 5 = 1$ resto 2
- $91 \not\equiv 18 \bmod 3$
  - $91 - 18 = 73$ non divisibile per 3 ($73 / 3 = 24$ resto 1)
  - $91 / 3 = 30$ resto 1 e $18 / 3 = 6$ resto 0

Congruenze

- Proprietà:
  - $a \equiv a \bmod n$
  - $a \equiv b \bmod n \Rightarrow b \equiv a \bmod n$
  - $a \equiv b \bmod n \text{ e } b \equiv c \bmod n \Rightarrow a \equiv c \bmod n$

Congruenze

- Se $a \equiv a' \bmod n$ e $b \equiv b' \bmod n$ allora:
  - $a + b \equiv a' + b' \bmod n$
  - $ab \equiv a'b' \bmod n$
  - $ka \equiv ka' \bmod n$ per qualsiasi k intero
  - Nota: questo non vale per la divisione

Inverso moltiplicativo

- Teorema di Bézout:
  - dati a e n con $\mathrm{MCD}(a,n) = 1$
  - esiste un unico intero b tale che $ab \equiv\ 1 \bmod n$
- Indichiamo questo intero con a^{-1} e lo chiamiamo *inverso moltiplicativo* di a

Funzione di Eulero

- Definiamo $\varphi(n)$ come il numero di interi k tra 1 e n tali per cui $\mathrm{MCD}(k, n) = 1$

Funzione di Eulero

- Definiamo $\varphi(n)$ come il numero di interi k tra 1 e n tali per cui $\mathrm{MCD}(k,n) = 1$

- Teorema di Eulero: dati a e n con $\mathrm{MCD}(a,n) = 1$, allora:

$$a^{\varphi(n)} \equiv 1 \bmod n$$

Funzione di Eulero - Proprietà

- $\varphi(1) = 1$
- $\varphi(p) = p - 1$ se p è un numero primo
- $\varphi(p^k) = (p-1)p^{(k-1)}$ se p è un numero primo
- $\varphi(pq) = (p-1)(q-1)$ se p e q sono numeri primi distinti

Funzione di Eulero - Proprietà

- $\varphi(1) = 1$
- $\varphi(p) = p - 1$ se p è un numero primo
- $\varphi(p^k) = (p-1)p^{(k-1)}$ se p è un numero primo
- $\varphi(pq) = (p-1)(q-1)$ se p e q sono numeri primi distinti
- In generale: $\varphi(ab) = \varphi(a)\varphi(b)$ se $\mathrm{MCD}(a,b) = 1$

Argomenti

- Problema dello scambio delle chiavi
- Cenni di teoria dei numeri
- Problemi facili e problemi difficili
- Scambio di chiavi Diffie-Hellman
- Crittografia a chiave pubblica e RSA
- Integrità, Autenticazione e Non-ripudio

Problemi facili e difficili

- Informalmente possiamo distinguere i problemi di teoria dei numeri in due tipologie:
  - Problemi "facili": problemi risolvibili con algoritmi efficienti
  - Problemi "difficili": problemi per i quali non si conoscono algoritmi efficienti

Problemi difficili

- Alcuni "problemi difficili" sono il cuore della crittografia moderna:
  - Facili da formulare
  - Impossibili da risolvere in pratica
  - Hanno un "problema facile" corrispondente
  - Permettono di creare schemi con dimostrazioni di sicurezza "forti"

Esempio: fattorizzazione

- Problema facile: prodotto di numeri (primi)
- Es: $13 \times 7 = 91$
- In generale $\approx n\log(n)$ operazioni per due interi a n cifre

- Problema difficile: fattorizzazione
- Es: $91 = ? \times ?$
- In generale sono necessarie $\approx e^{\sqrt[3]{n}}$ operazioni per fattorizzare numeri a n cifre

Esempio: fattorizzazione

- In pratica, nel caso della fattorizzazione di prodotti di due primi della stessa dimensione:
  - Per fattorizzare interi a 1024 bit sono necessarie circa 2^{70} operazioni
  - Per interi a 2048 bit sono necessarie circa 2^{90} operazioni
  - L'attuale record di fattorizzazione (al 2020) è di 829 bit

Esempio: logaritmo discreto

- Problema facile: elevamento a potenza modulare
- Es: $13^7 \equiv 9 \bmod 23$
- In generale $\approx nk\log(n)$ operazioni necessarie per un intero a n cifre e un esponente a k bit

- Problema difficile: logaritmo discreto
- Es: $9 \equiv 13^? \bmod 23$
- In generale, $\approx e^{\sqrt[3]{n}}$ operazioni per un modulo a n cifre

Argomenti

- Problema dello scambio delle chiavi
- Cenni di teoria dei numeri
- Problemi facili e problemi difficili
- Scambio di chiavi Diffie-Hellman
- Crittografia a chiave pubblica e RSA
- Integrità, Autenticazione e Non-ripudio

Scambio di chiavi Diffie-Hellman

“We stand today on the brink of a revolution in cryptography.”

– Whitfield Diffie and Martin Hellman, “New directions in Cryptography”, November 1976

Scambio di chiavi Diffie-Hellman

- Pubblicato nel 1976
- Basato sul problema del logaritmo discreto
- In grado di dare un "gap" esponenziale tra la complessità per gli utenti e quella per gli attaccanti

Scambio di chiavi Diffie-Hellman

- Step 1 - generazione dei parametri:
  - Alice e Bob scelgono un numero primo grande p
  - Alice e Bob scelgono un numero g tra 2 e $p-1$, detto generatore (spesso $g=2$)
  - Alice e Bob condividono pubblicamente questi parametri (chiunque può vederli)

Scambio di chiavi Diffie-Hellman

- Step 2 - generazione delle chiavi:
  - Alice sceglie un numero a tra 2 e $p-1$
  - Bob sceglie un numero b tra 2 e $p-1$
  - Alice calcola $A \equiv g^a \bmod p$ e Bob calcola $B \equiv g^b \bmod p$

Scambio di chiavi Diffie-Hellman

- Step 3 - scambio della chiave:
  - Alice condivide pubblicamente il valore di A
  - Bob fa lo stesso con B
  - Alice calcola $B^a \equiv g^{ab} \mod p$ e Bob calcola $A^b \equiv g^{ab} \mod p$
  - Alice e Bob condividono ora una chiave

Scambio di chiavi Diffie-Hellman

- Esempio:
  - Prendiamo $p = 37$ e $g = 2$
  - Alice genera il numero 7 e manda $2^7 \equiv 17 \bmod 37$
  - Bob genera il numero 21 e manda $2^{21} \equiv 29 \bmod 37$
  - Entrambi possono calcolare il numero
    $$2^{7 \cdot 21} \equiv 29^7 \equiv 17^{21} \equiv 8 \bmod 37$$
  - 8 è la chiave condivisa tra Alice e Bob

Quanto è sicuro Diffie-Hellman?

- Si crede che rompere Diffie-Hellman sia equivalente al recuperare uno dei numeri tra a e b, ovvero all'effettuare un logaritmo discreto
- In pratica, con p di 3072-bit si raggiunge una sicurezza comparabile a quella di un block cipher a 128 bit

Problemi

- I principali problemi di Diffie-Hellman sono:
  - Raggiungere alti livelli di sicurezza implica avere parametri molto grandi
  - A parità di dimensione, alcuni numeri primi sono più deboli di altri
  - Diffie-Hellman è vulnerabile ad attacchi attivi, come il *man-in-the-middle*

Man-in-the-middle

- Un attaccante che ascolta il protocollo può:
  - Intercettare g^a da Alice e sostituirlo con $g^{a'}$
  - Fare lo stesso con Bob, utilizzando un valore $g^{b'}$
  - Creare le chiavi $g^{a'b}$ e $g^{ab'}$ (Alice e Bob ora hanno chiavi diverse, ma non lo sanno!)
  - Decifrare ogni comunicazione, leggerla e cifrarla nuovamente con la chiave corretta

Argomenti

- Problema dello scambio delle chiavi
- Cenni di teoria dei numeri
- Problemi facili e problemi difficili
- Scambio di chiavi Diffie-Hellman
- Crittografia a chiave pubblica e RSA
- Integrità, Autenticazione e Non-ripudio

Fino ad ora…

- Alice e Bob vogliono comunicare:
  - Si accordano su un cifrario da utilizzare (es. AES-128)
  - Si scambiano una chiave (es. con Diffie-Hellman)
  - Iniziano a comunicare utilizzando il cifrario e la chiave

Crittografia a chiave pubblica

- Idea:
  - Sistemi che non richiedono lo scambio di chiavi
  - Ora una chiave è una coppia di valori (e, d) legati tra loro in un qualche modo
  - Gli utenti condividono e pubblicamente e tengono d segreto
  - Chiamiamo e chiave pubblica e d chiave privata

Crittografia a chiave pubblica

- La chiave pubblica e viene usata, insieme a un algoritmo di cifratura E, per cifrare i messaggi
- La chiave private d viene usata, insieme a un algoritmo di decifratura D, per decifrare i messaggi
- Il legame tra e e d, insieme alla struttura di E e D, permettono di effettuare cifratura e decifratura con chiavi diverse

Crittografia a chiave pubblica

- Se Alice vuole mandare un messaggio a Bob:
  - Alice prende la chiave pubblica e_{Bob} di Bob
  - Alice cifra il messaggio come $\mathrm{c} = E(e_{Bob}, m)$ e lo invia
  - Bob può decifrare il messaggio come $D(d_{Bob}, c)$
  - Nessun altro può decifrare il messaggio senza essere a conoscenza del valore di d_{Bob}

Crittografia a chiave pubblica

- Abbiamo quindi risolto il problema della cifratura?

Crittografia a chiave pubblica

- Abbiamo quindi risolto il problema della cifratura?
- Non proprio:
  - Gli schemi a chiave pubblica spesso si basano su algoritmi poco efficienti
  - Spesso le dimensioni delle chiavi devono essere maggiori di quelle del testo da inviare, e quindi non pratiche

Crittografia a chiave pubblica

- A cosa serve quindi la crittografia a chiave pubblica?
  - Mandare messaggi brevi
  - Scambiare chiavi
  - Applicare firme digitali

Il Sistema RSA

- Pubblicato nel 1977
- Primo schema a chiave pubblica
- Basato sul problema della fattorizzazione
- Utilizzato anche per le firme digitali

Il Sistema RSA

- Step 1 - generazione della chiave:
  - Alice genera due numeri primi grandi p e q
  - Alice calcola $N = pq$ e $\varphi(N) = (p-1)(q-1)$
  - Alice sceglie un esponente pubblico e e calcola l'esponente privato $d \equiv e^{-1} \bmod \varphi(N)$
  - La coppia (N, e) è la chiave pubblica, mentre la terna (p, q, d) è quella privata

Il Sistema RSA

- Step 2 – cifratura:
  - Bob effettua la cifratura di un messaggio m come

$$c \equiv m^e \bmod N$$

Il Sistema RSA

- Step 3 – decifratura:
  - Alice decifra il messaggio come

$$m \equiv c^d \bmod N$$

  - Infatti vale che:

$$c^d \equiv m^{ed} \equiv m^{k\,\varphi(N)+1} \equiv m \cdot \left(m^{\varphi(N)}\right)^k \equiv m \bmod N$$

- Nota: per funzionare correttamente il sistema richiede $\mathrm{MCD}(e, \varphi(N)) = 1$

Il Sistema RSA

- Esempio:
  - Scegliamo $p = 11$ e $q = 17 \Rightarrow \ N \ = \ 187$ e $\varphi(N) = 160$
  - Prendiamo $e = 3$ e calcoliamo $d \equiv \ 3^{-1} \equiv \ 107 \bmod \varphi(N)$
  - Prendiamo $m = 10$ e calcoliamo $c \equiv \ 10^3 \equiv \ 65 \bmod N$
  - Per decifrare: $c^d \ \equiv \ 65^{107} \ \equiv \ 10 \bmod N$

Il Sistema RSA

- RSA nel mondo reale
  - Per questioni computazionali si usa $e = 3$ o $e = 65537$
  - È necessario un sistema di padding
  - L'elevamento a potenza non è fatto direttamente $\bmod N$

Attacchi a RSA -1

- La scelta di $e = 3$ può essere vulnerabile
  - Low public exponent attack
- La scelta di e molto grande può essere vulnerabile
  - Wiener attack
- Due chiavi che condividono un fattore primo possono essere rotte facilmente
  - Common prime attack

Attacchi a RSA - 2

- Due chiavi con lo stesso modulo ma esponenti diversi possono essere vulnerabili
  - common modulus attack
- Un gruppo di e chiavi con lo stesso esponente, ma moduli diversi, può essere vulneabile
  - Hastad broadcast attack
- Diversi tipi di oracle a prima vista "innocui" possono rompere completamente RSA
  - Blinding, padding oracle, lsb oracle

Argomenti

- Problema dello scambio delle chiavi
- Cenni di teoria dei numeri
- Problemi facili e problemi difficili
- Scambio di chiavi Diffie-Hellman
- Crittografia a chiave pubblica e RSA
- Integrità, Autenticazione e Non-ripudio

Recap

- Confidenzialità
- Integrità
- Autenticazione
- Non-ripudio

Recap

- Confidenzialità
- Integrità
- Autenticazione
- Non-ripudio

Funzioni di hash

- Una funzione di hash:
  - Prende in input un messaggio M di lunghezza arbitraria
  - Produce in output una stringa di lunghezza fissata $H(M)$ detta digest (o, informalmente, hash) del messaggio
  - Nel processo non viene utilizzata nessuna chiave (chiunque può replicare il calcolo)

Funzioni di hash crittografiche

- Una funzione di hash si dice crittografica se (informalmente):
  - È difficile risalire a un possibile input dato un output
  - È difficile trovare delle collisioni: due messaggi M_1 e M_2 tali che $H(M_1) = H(M_2)$
  - Un piccolo cambiamento nell'input comporta un grande cambiamento nell'output (effetto valanga)

Integrità

- Possiamo usare le funzioni di hash per garantire integrità:
  - Inviamo insieme al messaggio anche il suo digest
  - Il ricevente riceve il messaggio e ne calcola indipendentemente il digest
  - Controllando che i due digest coincidano, il destinatario verifica che il messaggio sia rimasto integro

Integrità

- Problema:
  - Una funzione di hash può proteggerci da errori di trasmissione casuali
  - Cosa succede se un attaccante modifica volontariamente il messaggio?
    - L'attaccante modifica il messaggio
    - L'attaccante modifica opportunamente il digest
    - Il destinatario non può accorgersi delle modifiche

Recap

- Confidenzialità
- Integrità
- Autenticazione
- Non-ripudio

Message Authentication Codes

- Un Message Authentication Code (MAC) può essere visto come una funzione di hash con chiave:
  - Prende in input un messaggio M di lunghezza arbitraria e una chiave K
  - Produce in output una stringa di lunghezza fissata $MAC(K, M)$ chiamata tag

Message Authentication Codes

- MAC in pratica:
  - A partire da block cipher (CBC-MAC, NMAC)
  - A partire da funzioni di hash (HMAC)

Message Authentication Codes

- MAC in pratica:
  - Alice e Bob condividono una chiave
  - Alice manda insieme al messaggio anche il tag
  - Un attaccante non può modificarlo non conoscendo la chiave
  - In questo modo garantiamo sia integrità che autenticazione

Firme Digitali

- Informalmente: una firma digitale è l'equivalente "a chiave pubblica" di un MAC:
  - Non richiede uno scambio di chiavi
  - Utilizza una funzione di firma attraverso la chiave privata (solo il proprietario può firmare)
  - Utilizza una funzione di verifica attraverso la chiave pubblica (chiunque può verificare)

Firme Digitali

- Pro delle firme digitali:
  - Non serve scambiare delle chiavi
  - Possiamo garantire anche il non-ripudio

- Contro delle firme digitali:
  - Sensibilmente più lente dei MAC
  - Hanno in generale chiavi di dimensioni maggiori

Recap

Primitiva	Integrità	Authenticazione	Non-ripudio
Hash	Si	No	No
MAC	Si	Si	No
Firme Digitali	Si	Si	Si

Matteo ROSSI

Politecnico di Torino

Cryptography 2

https://cybersecnatlab.it