

E-GOVERNMENT = E' L'USO DELLA TECNOLOGIA PER L'INFORMAZIONE
(IN PARTICOLARE INTERNET) PER CREARE E SVILUPPARE SERVIZI TRA CLIENTI E GOVERNO.

↳ CORRISPONDENZA VIA E-MAIL

↳ ONLINE VOTING

↳ ...

LEC-01

INFORMATION SECURITY?

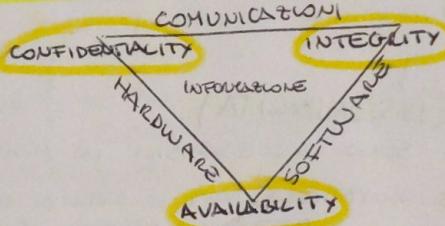
↳ COMPUTER SECURITY: PREVENTION e DETECTION DI AZIONI NON AUTORIZZATE DA PARTE DI UTENTI IN UN COMPUTER.
(IMPORTANTI: AUTORIZZAZIONE e SECURITY POLICY)

↳ NETWORK SECURITY: POLITICHE E SISTEMI ADOTTATI DALL'AMMINISTRATORE PER PROTEGGERE LA RETE E LE RISORSE DISPONIBILI IN RETE DA ACCESSI NON AUTORIZZATI.

↳ INFORMATION SECURITY: E' PIU' GENERICA, CENTRA CON L'INFORMAZIONE INDIPENDENTEMENTE DAL COMPUTER SYSTEM.

↳ PIROTEGGERE L'INFORMAZIONE E GLI INFORMATION SYSTEM DA ACCESSI NON AUTORIZZATI (MA ANCHE DALL'USO, DALLA MANIPOLAZIONE E DALLA DISTRIBUZIONE).

LA SICUREZZA E' UN PROBLEMA DI TUTTO IL SISTEMA:
SOFTWARE, HARDWARE, COMUNICAZIONI...
DAL PC PERSONALE ALLA RETE...



► SECURITY AS POLICY COMPLIANCE

(SICUREZZA COME CONFORMITA' A UNA POLICY)

SI E' ORIENTATI A UNA CONFORMITA' A UNA POLICY O A DEI GOALS

LA COMPUTER SECURITY SI OCCUPA DI PREVENTION e DETECTION LE AZIONI IMPROPRIE FATTE DA UTENTI DI UN COMPUTER SYSTEM.

↳ UNA PROSPETTIVA E' DATA DALL' INGEGNERIA DEL SOFTWARE

↳ SPECIFICA COSA IL TUO SISTEMA DOVREBBE FAR E'

↳ DISSEGNA E IMPLEMENTA SISTEMI CHE PORTANO A TERMINE LA TUA SPECIFICA.

↳ NEL CASO DELLA SICUREZZA LE SPECIFICHE PREVEDONO COMPORTAMENTO INCACCETTABILE

SECURITY POLICY: SPECIFICA COSA NEL NOSTRO SISTEMA E' O NON E' PERMESSO

PER QUANTO RIGUARDA LA SICUREZZA:

- SPECIFICATION (SPECIFICA): POLICY
- IMPLEMENTATION (IMPLEMENTAZIONE): MECCANISMO (IMPIEGATO NEL SISTEMA)
- CORRETTEZZA: COMPLIANCE (CONFORMITA' ALLE SPECIFICHE)
(CORRECTNESS)

↳ USATO X IMPOSTARE LA POLICY.

- LA SPECIFICA DOVREBBE VALERE NEL MECCANISMO IMPLEMENTATO IN TUTTI GLI AMBIENTI POSSIBILI.
- LA DOMANDA SUCA CORRETTEZZA DI UN SISTEMA PUO' NON ESSERE COSÌ FACILE (E' DIFFICILE).

↳ TRADITIONAL SECURITY PROPERTIES/GOALS

INFORMATION SECURITY IS CIA: - CONFIDENTIALITY

- INTEGRITY

- AVAILABILITY

VEDIAMOLE TUTTE: AUTHORIZATION (AUTENTICATION AND ACCESS CONTROL)

- CONFIDENTIALITY: L'INFORMAZIONE NON PUO' ESSERE VISIBILE/CAPITIBILE DA CHI NON HA L'AUTORIZZAZIONE.
- INTEGRITY: L'INFORMAZIONE NON PUO' ESSERE MODIFICATA, ALTERATA DA CHI NON E' AUTORIZZATO.
- AVAILABILITY: L'INFO DEVE ESSERE SEMPRE DISPONIBILE (NESSUN DANNEGGIAMENTO DELLE FUNZIONALITA').
- ACCOUNTABILITY (TRACCIABILITA'): LE AZIONI SONO SEMPRE TRACCIABILI A CHI LE HA COMPIUTE.
- AUTHENTICATION (AUTENTICAZIONE): I PRINCIPALI E L'ORIGINE DEI DATI POSSONO ESSERE IDENTIFICATI ACCURATAMENTE

CONFIDENTIALITY, PRIVACY, SECRECY

- LA CONFIDENZIALITÀ RIGUARDA LA LETTURA AUTORIZZATA DI DATI.
 - NON PARLAMO SOLO DI ACCESSO AI DATI MEMORIZZATI SUL SISTEMA, UN ATTACCHANTE NON DEVE POTER NESSUNO ACQUISIRE INFORMAZIONI IMPLIQUE DAL SISTEMA (es. STATISTICHE USO DATI)
 - LA CONFIDENZIALITÀ PRESUME UNA NOTIONE DI PARTE AUTORIZZATA
- ↳ UNA SECURE POLICY DICE CHI E COSA PUÒ FARE CON I DATI,
ACCEDERE AI
- PRIVACY: CONFIDENTIALITY PER GLI INDIVIDUI SINGOLI
 - SECRECY: CONFIDENTIALITY PER LE ORGANIZZAZIONI
- D LA PRIVACY PUÒ
ESSERE USATA
TALVOLTA NEL
SENSO DI ANONYMITY

ANONYMITY, ≠ PRIVACY,
L'IDENTITÀ VERA NON È CONSCIA SI PUÒ SCOPRIRE COSA LE ALTRE PERSONE POSSONO CONOSCERE

LE EMAIL DI DEFAULT NON HANNO ENCRYPTION (SE VOGLIAMO IMPLEMENTARE LA CONFIDENTIALITY AGGIUNGHIAMO ENCRYPTION E CONTROLLO DEGLI ACCESSI).

INTEGRITY

I DATI NON VENGONO ALTERATI IN MODO MALEVOLO (DA PRINCIPAL MALEVOLO) ATTACCHANTI

↳ QUESTO PRESUPPOSTO CI SUA UNA SECURITY POLICY CHE DICE COME E CHI È AUTORIZZATO A MODIFICARE I DATI.

↳ PER IMPLEMENTARE SULLE MAIL POSSIAMO AGGIUNGERE LE FIRME DIGITALI E/O IL CONTROLLO DELL'ACCESSO.

AVAILABILITY (DISPONIBILITÀ)

DATI E SERVIZI SONO ACCESSIBILI IN MODO AFFIDABILE E IN TEMPI CONGRUI.

↳ SI INTENDE PROTEGGERE DATI E SERVIZI DA:

- ATTACCHI MALEVOLO (es. DOS → DENIAL OF SERVICE)
- ATTACCHI ACCIDENTALI (es. DIPENDENTI DISTURBI...)
- EVENTI AMBIENTALI (es. INCENDI, INONDAZIONI...)

↳ SOLITAMENTE VENDONO DIMENTICATI!

DOS → E' DIFFICILE PROTEGgersi DA ATTACCHI CHE ESauriscono LE RISORSE DEL SISTEMA

- E' DIFFICILE DISTINGUERE TALVOLTA TRA:
USO ELEVATO MA LEGITIMO ↔ DOS

IDENTIFICARE MINACCIE

IMPLEMENTARE MECCANISMI PER DIFENDERSI

SFIDE CHE CI PONEMO PER COPRIRE TUTTE LE MINACCIE.

TRACCABILITÀ

LE AZIONI SONO REGISTRATE E POSSONO ESSERE TRACCiate A CHI LE HA CAMPIUTE (CHI NE È RESPONSABILE)

↳ IDEA: MANTENERE TRACCIA DI QUANTO SUCCIDE (SECURE AUDIT TRAIL)

↳ UNA FORMA DI TRACCABILITÀ È LA NON-REPUDIATION

MANTENERE QUESTI FILE NON È SEMPRE, SE UN SISTEMA È COMPROMESSO ANCHE QUESTI FILE POSSONO ESSERE COMPROMESSI (SOLUZIONI POSSIBILI: SERVER SEPARATO FILE DI SOLO LETTURA...)

AUTENTICAZIONE

DATI E SERVIZI SONO DISPONIBILI SOLO A CHI È AUTORIZZATO

↳ VERIFICA DELL'IDENTITÀ DI PERSONE O SISTEMI

↳ MOLTO IMPORTANTE PER IMPLEMENTARE IL CONTROLLO DELL'ACCESSO

↳ PER IMPLEMENTARE L'AUTENTICAZIONE, SI USANO DEI METODI BASATI SU:

↳ QUALcosa CHE ABBIANO (es. DOCUMENTO O CARTA)

↳ QUALcosa CHE CONOSCINO (es. PASSWORD)

↳ QUALcosa CHE SONO (es. IMPRONTA DIGITALE)

POSSONO ESSERE COMBINATI X SICUREZZA EXTRA

PROPRIETÀ

POLICIES

MECHANISMS

ALTO NUOVO/ASTRAZIONE

"AND" DI DIVERSE PROPRIETÀ

DA BASSO NUOVO E OPERAZIONALE (PRACTICO)

DISTINZIONE NON CHIARA.

PREVENTION

DETECTION

RESPONSE

PREVENIRE DI ESSERE ATTACCATO (es. FIREWALL)

IN CASO DI BRECCIA DI SICUREZZA, CI INTERESSA ESSERE SICURI DI

→ INTRUSION DETECTION SYSTEM
LOGGING AND MACS

RISPONDERE ALL'ATTACCO!
(es. RESTORE BACKUP)

SECURITY AS RISK MINIMIZATION

→ E' UN APPROCCIO BEN DEFINITO NELLA PRATICA (APPROCCIO SCELTO + COMUNEMENTE)

- SICUREZZA E' LA PROTEZIONE DELLE RISORSE DALLE MINACCE

↳ SI VOGLIE PROTEGGERE LE RISORSE (SONO IMP. PER L'AZIENDA), SOPRATTUTTO QUELLE RISORSE CHE SONO DI VALORE.

↳ ANCHE GLI ATTACCATI FANNO UNA ANALISI DELLE RISORSE PIÙ DI VALORE E VOLGONO ABUSARNE.

I PROPRIETARI VAGLIANO LE VARIE MINACCE E QUALE RISCHIO POSSONO PORTARE → PER OGNIUNA DI ESSE ADOTTARE UNA CONTRAMISURA.

↳ PROBLEMA: NON POSSIAMO PROTEGGERCI DA TUTTE (COSTI/FATTIBILITÀ)

ANCHE ACCIDENTALI
NON BISOGNA
DIMENTICARSI

SOLITAMENTE CI CONCESSIONANO SULLE MINACCE ~~MALEVOLI~~ MALEVOLI (HACKERI)

RIMANGONO
ALCUNI
RISCHI RESIDUI

VAGLIAMO IL RISCHIO

RISCHIO = POSSIBILITÀ DI ABUSO X IMPATTO SUL SISTEMA

LA NOZIONE DI RISCHIO E' CENTRALE. → IN CONTRASTO

POLICY CHE INDICA QUALI AZIONI SULLE RISORSE SONO NON AUTORIZZATE.

ESEMPI DI ATTACCO = INJECTION, CROSS-SITE SCRIPTING,

CHI SONO QUESTI THREAT AGENTS (AGENTI MALEVOLI)?

- SENZA ↴
• AZIONI NON INTENZIONALI DA PARTE DI DIPENDENTI
• HACKERI GUIDATAI DA SFIDA PERSONALE
• EX DIPENDENTI
• CRIMINALI, TERRORISTI, AGENTI DI SPIONAGGIO ESTERI.

• UNA VULNERABILITÀ È UNA DEBOLEZZA CHE PUÒ ESSERE SFRUTTATA DA UNA MINACCIA (ATTACCO) PER CAUSARE DANNO.

↳ DA DOVE ARRIVANO LE NOSTRE VULNERABILITÀ?

- AMBIENTE FISICO: PC IN MANI NEMICHE
- AMBIENTE DI RETE: INTERNET
- AMBIENTE SOFTWARE: OS, DRIVERS, ...

• SUPERFICIE D'ATTACCO = DOVE AVVIENE/PUÒ AVVENIRE L'ATTACCO

COME PROCEDIAMO: RISK ANALYSIS AND REDUCTION STEPS

ES MAIL CON PGP

- 1-ANALISI DEI RISCHI ESISTENTI
 - IDENTIFICARE LE RISORSE DA PROTEGGERE
 - IDENTIFICARE I RICHI PER LE RISORSE (RICHIESTE DI COMPRENSIONE DELLE VULNERABILITÀ DEL SISTEMA, DEGLI ATTACCONTI E DELLE MINACCIE)
- 2-ANALISI DELLE SOLUZIONI DI SICUREZZA PROPOSTE
 - CONTRAMISURE, CI SONO ALTRI RISCHI? CI SONO NUOVI RISCHI?

⚠ TRADEOFF TRA ① RIDURRE AL MINIMO IL RISCHIO DI ABUSO
② COSTRUIRE UN SISTEMA CHE SODDISFI UNA SPECIFICA

INGEGNERIA DELLA SICUREZZA: è difficile

- FAR SI CHE IL SISTEMA SI COMPORTI NEL MODO SPECIFICATO → INGEIERIA DEL SW.
- IMPEDIRE A UN SISTEMA DI COMPORTARSI IN MODI NON SPECIFICATI → INGEIERIA DELLA SICUREZZA
- IMPLEMENTARE UNA SOLUZIONE ATTIVAMENTE: ANALISI, MODELLO, RICERCA DEI RISCHI, ...
- APPROCCIO OLISTICO = LA SICUREZZA DEVE ESSERE CODIFICATA CON IL SISTEMA, NON AGGIUNTA!

PUNTI
• CHIARE GLI OBIETTIVI PRIMA DELLO SVILUPPO → RIDURRE LA COMPLESSITÀ • VERIFICARE CHE LE RICHIESTE DI SICUREZZA SONO SODDISFAVTE.

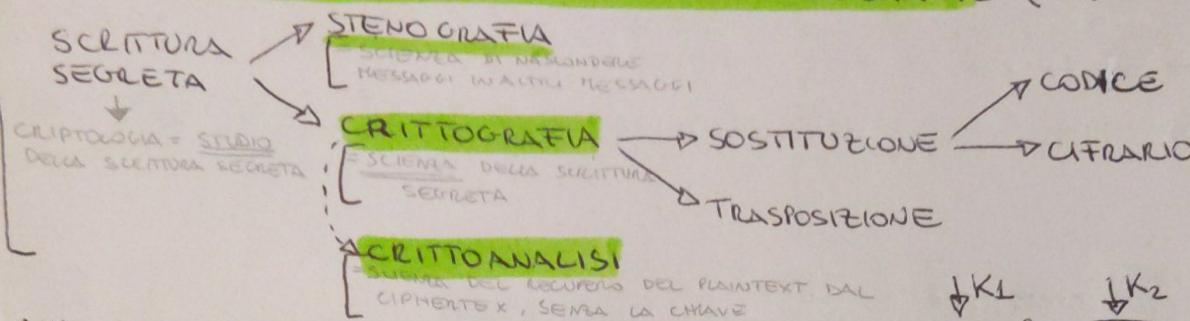
METHODI FORMALI: TECHNICHE E TOOL BASATI SULLA MATEMATICA E SULLA LOGICA CHE SUPPORTANO LA SPECIFICA, LA COSTRUZIONE E L'ANALISI DI SISTEMI HW E SW.

↳ MOLTO BUONI PER COSTRUIRE MODELLI, DOCUMENTAZIONE NON AMBIGUA E VALIDARE RICORSIVAMENTE

- COME TRASFORMARE CANALI INAFFIDABILI IN CANALI AFFIDABILI?**
- **CONFIDENTIALITÀ** = LE INFO TRASMESSE RIMANGONO SEGRETE
 - **INTEGRITÀ** = INFORMAZIONI NON CORROTTE (O RILEVO ALTERNZIONI)
 - **AUTENTICAZIONE** = I PRINCIPALI SANO CON CHI STANNO PARLANDO
 - **ALTRI OBIETTIVI** = TEMPESTIVITÀ, INOSSERVABILITÀ

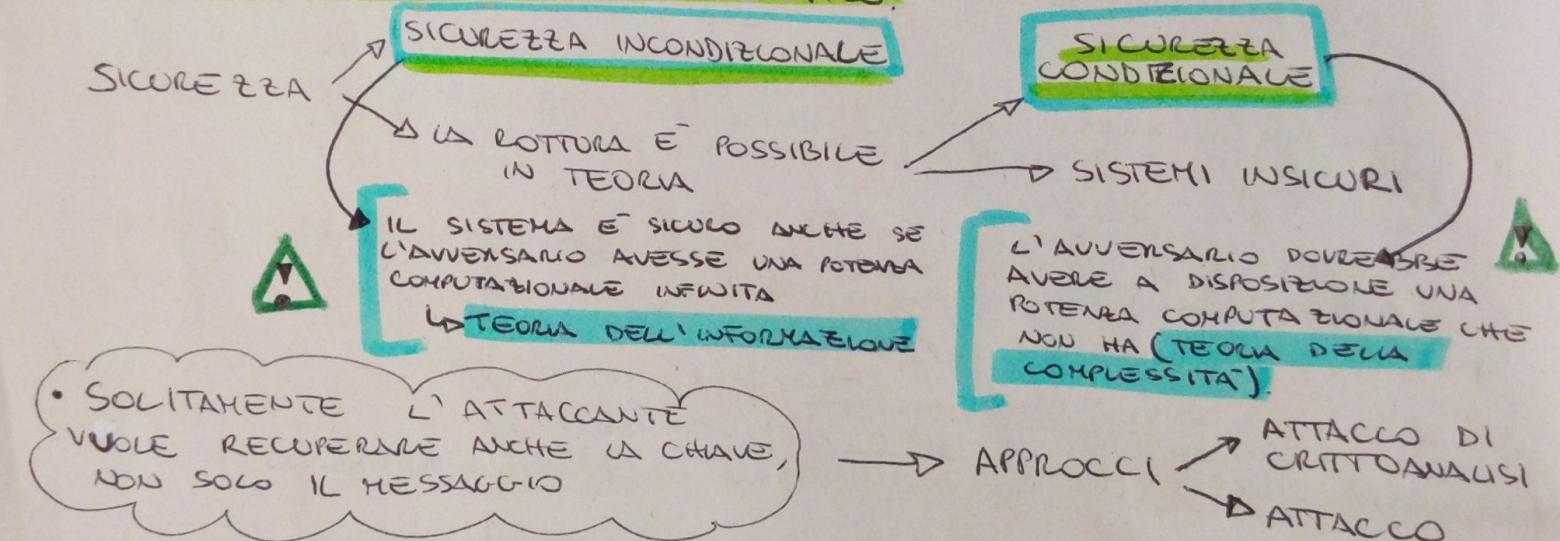
LEC-02

LA CRITTOGRAFIA È LA TECNOLOGIA ABILITANTE (TRASFORMA CANALE NON SICURO IN SICURO)



• IN UNO SCHEMA GENERICO ABBIAMO $M \rightarrow E \xrightarrow{C} D \xrightarrow{H} H$, NEGLI ALGORITMI A CHIAVE SIMMETRICA $K_1 = K_2$ SONO FACILMENTE DERIVABILI UNA DALL'ALTRA, MENTRE NEGLI ALGORITMI A CHIAVE PUBBLICA ABBIAMO UNA CHIAVE PUBBLICA CHE PUÒ ESSERE PUBBLICATA SENZA COMPROMETTERE LA CHIAVE PRIVATA.

LA SICUREZZA DEVE DIPENDERE DALLA SEGRETEZZA DELLA CHIAVE E NON DALLA SEGRETEZZA DELL'ALGORITMO.



► INFORMAZIONI CHE UN ATTACCANTE HA ALL'INIZIO:

- SOLO TESTO CIFRATO
- TESTO CIFRATO E RELATIVO PLAINTEXT CHOOSEN PLAINTEXT
- PLAINTEXT VIENE SCELTO → IL TESTO CIFRATO RELATIVO VIENE FORNITO
- CHOOSEN ADAPTIVE PLAINTEXT → IL CRITTOANALISTA PUÒ MODIFICARE IL PLAINTEXT PER VEDERE CHE VARIA LA CRITTAZIONE
- CHOOSEN DIFFERENTEXT → IL CRITTOANALISTA PUÒ SCEGLIERE IL CIPHERTEXT DA DECODIFICARE

► COME CREARE UNA DEFINIZIONE DI SIUREZZA!

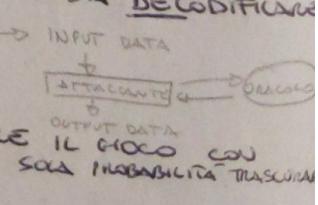
- SPECIFICARE UN ORACOLO (TIPO DI ATTACCO, TRA QUESTI)
- DEFINIRE COSA L'UTENTE DEVE FARLE PER VINCERE IL GIOCO
- IL SISTEMA È SICURO SE NESSUN ATTACCANTE EFFICIENTE PUÒ VINCERE IL GIOCO

► UN ESEMPIO È LA CONVENTIONAL ENCRYPTION

- NESSUN DATO IN INPUT PER IMPEDIRE L'ATTACCANTE
- L'ATTACCANTE PUÒ FARE CHOOSEN PLAINTEXT ATTACK DI TIPO

↳ CIFRA UN MESSAGGIO CHE TU HO DATO IN

↳ CIFRA UN MESSAGGIO A CASO



SIMMETRIC KEY ENCRYPTION (SINGLE-KEY, SHARED-KEY)

CONSIDERA LO SCHEMA DI DECODIFICAZIONE $\{E_e | e \in K\}^{-1} \{D_d | d \in K\}$

↪ LO SCHEMA E' A CHIAVE SIMMETRICA SE E' COMPUTAZIONALMENTE FACILE DETERMINARE d DA e . IN PRATICA $d = e$

↪ NELL'ETÀ PRIMA, DUE INDIVIDUI HANNO UNA CHIAVE CONDIVISA CON CUI CRITTANO I MESSAGGI DA UN'ALTRA ALL'ALTRO.

↪ ANCORA MOLTO USATI POICHÉ VELOCI

BLOCK CIPHER

ENCRIPTION SCHEME IN CUI IL MESSAGGIO VIENE DIVISO IN BLOCCHI DI LUNGHEZZA FISSATA T .
VIENE CRIPTATO UN BLOCCO ALLA VOLTA

STREAM CIPHER

BLOCK CIPHER DOVE LA LUNGHEZZA DEL BLOCCO E' 1.

CODES (CODICI)

FUNZIONANO SU PAROLE DI VARIA LUNGHEZZA

SUBSTITUTION CIPHERS

↪ es ALGORITMO DI CESARE → OGNI LETTERA VIENE TRASPOSTA DI UN NUMERO DI POSIZIONI (SOLO 26 CHIAVI)
↪ BRUTE FORCE (SI PUO' VEDERE CHE UN MODO X CAPIRE QUANDO IL TESTO E' IN CHIARO)

↪ MONO-ALFABETICO → LA CHIAVE E' UNA DELLE 26! PERMUTAZIONI DELLE LETTERE DELL'ALFABETO
↪ CRIPTOANALISI (SI PONE CON L'ANALISI DELLE FREQUENZE X OGNI LINGUA)

↪ CRIPTOANALISI CON CRIPTOANALISI DELLE FREQUENZE RENDO LA CODIFICA NON UNIVOCATA
↪ CRIPTOANALISI DI SOSTITUZIONE OMOFONICO → RIPARZO OGNI LETTERA CON UNA STRINSA PRESA DA UN INSIEME $H(3)$. LA CHIAVE SONO GLI INSIEMI.
↪ L'ANALISI DELLE FREQUENZE E' PIU' DIFFICILE

↪ CRIPTOANALISI DI VICENZE → SOFISTICATO DI QUELLO DI CESARE,
CHIAVE $K = K_1 K_2 K_3 = 3, 7, 8$
CRIPTANTE MESSAGGIO
$$\begin{array}{c} \text{H} \\ \text{H} \\ \text{H} \\ \downarrow \\ \text{H} \end{array} \quad \begin{array}{c} \text{H} \\ \text{H} \\ \text{H} \\ \downarrow \\ \text{H} \end{array} \quad \begin{array}{c} \text{H} \\ \text{H} \\ \text{H} \\ \downarrow \\ \text{H} \end{array}$$

↪ CRIPTOALFABETICO → PASSO IN PIU' RISPETTO A VICENZE PER OGNI BLOCCO NON VIENE PIU' STERZO UNO SHIFT DIVERSO MA UNA PERMUTAZIONE DIVERSA
→ LA CHIAVE E' UNA DIM DEL BLOCCO E LE PERMUTAZIONI.

↪ ONE-TIME-PADS (VERNAME CIPHER) → SI USA LO XOR PER COMPLICARE LA SINCRONIZZAZIONE DELLO STREAM DA PARTE DI TERZI NON VOLUTI
↪ $\text{XOR}(\text{MESSAGE}, \text{KEY}) = \text{CH}$
 $\text{XOR}(\text{CH}, \text{KEY}) = \text{MESSAGE}$
→ PROBLEMA: LA PASSWORD E' CONDIVISA E DUNQUE DIFFICILE DA SCELIRE
→ IL MESSAGGIO CRIPTATO NON CONTIENE NESSUNA INFO SUL MESSAGGIO IN CHIARO!
→ VERDE TATO IN CRIPTOLOGIA

► TRASPOSITION CIPHERS

- ↳ I CARATTERI VENGONO SCIFATI SECONDO UN SISTEMA
- ↳ SONO COMUNI LE ANALISI DI FILE OPIENA POICHÉ LE LETTERE SONO LE STESSSE CAPOGGIANDO SOLO DI POSIZIONE
- ↳ SCYTALE → CINTURA ~~ROTANTE~~ CON LETTERE AVOLTA ATTORNO A UN BASTONE, LA CHIAVE È UN DIA DI BASTONE

► COMPOSITE CIPHERS

- ↳ CRIPTARE SOLO TRAMITE L'USO DI SOSTITUZIONI → SOLO TRAMITE L'USO DI TRASPOSIZIONI NON È SICURO
- ↳ SOLUZIONE! LI COMBINIAMO!
- 2 SUBSTITUTION = 1 SUBSTITUTION
- 2 TRANSPOS = 1 TRANSPOS
- 1 TRANS + 1 SOST = HARDER CIPHER!
- ↳ VENGONO INVENTATE LE CIPHER MACHINE

- ↳ SHANNON → IDEA DELLA COMBINAZIONE DEI DUE

- PRIME BASI DEL CRIPTALE → BLOCCHE
 - SUBSTITUTION → CONFUSION OF MESSAGE }
 - PERMUTATION → DIFFUSION OF MESSAGE }
- { DOBBLIO NASCONDENDO IL PIÙ POSSIBILE LE PROPRIETÀ DEL MESSAGGIO OGNI VOLTA

- ↳ FEISTEL → IL TESTO IN CHIAVO VIENE DIVISO IN DUE META' HL, HR
- LE DUE PARTI VENGONO CIFRATE IN n ROUND E Poi VENGONO RIMESSSE ASSIEME PER FORMARE IL MESSAGGIO FINALE CIFRATO
- OGNI ROUND:
 - APPLICO A HR UNA ROUND FUNCTION CON UNA CHIAVE
 - FACCIO LO XOR CON HL
 - SCAMBIO LE DUE META'

↳ DES (DATA ENCRYPTION STANDARD)

- VARIANTE DI FEISTEL A 16 ROUND.
- DIVIDO IL TESTO IN CHIAVO VIENE DIVISO IN BLOCCHE DA 64 bit
- LA SICUREZZA STA NELLA ROBUSTEZZA DELLA CHIAVE
 - NECESSARIO ALMENO UNA CHIAVE DA 128 bit.

↳ BRUTE-FORCE ATTACK (QUANDO TESTO NERPURO E ONE)

↳ ATTACCO MATEMATICO USANDO LA LINEAR CRITTOANALISI, SI RIDUCE LO SPAZIO DELLE CHIAVI DA 2^{56} A 2^{43}

- ↳ DOUBLE-DES : DUE DES IN CASCATA (NON COMPUTAZ + DIFFICILE DI DES)
- IL NUMERO DI CHIAVI NON È 2^{112} COME SI PENSEREbbe
- OGNI CHIAVE A UN MAN IN THE MIDDLE TRA DUE
 - MI RIDUCO A UNO SPAZIO DI ~~2^{56}~~ 2^{56}

► TRIPLE-DES → METTE IN CASCATA 3 DES (IN UN MODO NON BANALE)

- VENGONO USATE 2 CHIAVI (K_1 e K_2)
- IL PRIMO È IL TERZO DES → K_1
- IL SECONDO DES → K_2

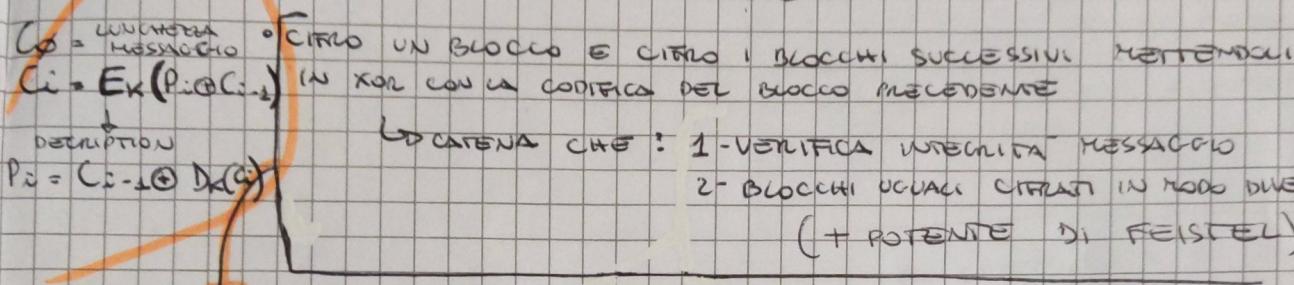
► ATTACCO UNICO POSSIBILE

→ BRUTE FORCE IN KEY SPACE 2^{112}

- MANTENERE COMPATIBILITÀ CON DES, USANDO $K_1 = K_2$

► AES (ADVANCED ENCRYPTION STANDARD)

► NON HA UNA STRUTTURA DI FEISTEL, BLOCCI LUNghi 128 bit
VENGONO PROCESSED IN PARALLELO



N.B! SI CHIAMA CHIPER-BLOCK CHAINING
PASSO IN AVANTI RISPIETTO A ELECTRONIC CODEBOOK

DIVISO IN BLOCCI È LI CIFRA
UNO A UNO SENZA TENERE CONTO
DEGLI ALTRI.

NUOVO PROBLEMA: DOVE PIAZZO L'ENCRYPTION?

LINK-ENCRYPTION (LAYER 1-2 ISO OSI)

- LA CRIPTAZIONE VIENE FATTA INDIPENDENTEMENTE SU OGNI LINK
- IL TRAFFICO DEVE ESSERE CRIPTATO E DECRYPTATO SU OGNI LINK
- RICHIEDE COPPIE DI CHIAVI

- PROTEGGE I FLUSSI DI TRAFFICO DAL MONITORAGGIO

→ PIÙ IN ALTO ANDIAMO NELLA PIÙ È

- MENO INFORMAZIONE È CRIPTATA
- PIÙ SICUREZZA MA + COMPLESSA (E CON + OLTRE CHIAVI)

END-TO-END ENCRYPTION (LAYERS 3-4-6-7)

- LA CRIPTAZIONE VIENE FATTA DA SCONESE E DESTINAZIONE
- HA BISOGNO DI UN DISPOSITIVO SU OGNI FINE LINK
- CHIAVI CONDIVISE TRA LE DUE PARTI DELLA COMUNICAZIONE.

- PROTEGGE I DATI CRIPTANDOLI TRA LE 2 PARTI (GLI HEADER DEVONO PERMettere IN CHIAVE PER IL ROUTING CORRETTO)

DEALMENTE
LI VOLGONO
ENTRAMBI!

- RICHIEDE VARIE COPIE DI CHIAVI (INTERMEDIATE NODES)
- CRIPTAZIONE PUÒ ESSERE FATTA HARDWARE
- TRANSPARENTE ALL'UTENTE
- TUTTO VIENE CRIPTATO.

• RICHIEDE SOLO UNA COPIA DI CHIAVI

• SOLO SW.

- L'UTENTE DEVE IMPLEMENTARE LA CRIPTAZIONE
- L'UTENTE SCEGLIE COSA CRIPTARE

APPLICATION-LEVEL ENCRYPTION

- LINK ROUTER E GATEWAY VEDONO CRIPTATA SOLO LA PARTE DI DATA

TCP-LEVEL ENCRYPTION

- LINK E ROUTERS VEDONO CRIPTATA
- GATEWAY VEDONO TUTTO IN CHIAVE

LINK-LEVEL ENCRYPTION

- SUL LINK CRIPTO NELLE LIP
- I CONTROL E I GATEWAY VEDONO TUTTO IN CHIAVE

SIMMETRIC ENCRYPTION

- STESSO ALGO E STESSA CHIAVE PER CRIPTAZIONE E DECRYPTAZIONE
- IL MITTENTE E IL DESTINATARIO DEVONO CONDIVIDERE CHIAVE E ALGORITMO.
- **X LA SICUREZZA**
- LA CHIAVE DEVE RIMANERE SEGRETA
- E' IMPOSSIBILE O IMPRACTICABILE DECRYPTARE UN MESSAGGIO SE NESSUNA ALTRA INFO E' DISPONIBILE.
- LA CONOSCENZA DELL'ALGORITMO + ESEMPI DI TESTI CIFRATI DEVONO ESSERE INSUFFICIENTI A DETERMINARE LA CHIAVE

PUBLIC-KEY ENCRYPTION

LEC 03

- UN ALGORITMO PER CRIPTARE E UNO PER DECRYPTARE CON UNA COPPIA DI CHIAVI UNA PER CRIPTARE E UNA PER DEC.
- IL MITTENTE DEVE avere UNA CHIAVE E IL DESTINATARIO DOVE avere L'ALTRA.
- **X LA SICUREZZA**
- UNA DELLE DUE CHIAVI DEVE RIMANERE SEGRETA
- **II II**
- LA CONOSCENZA DELL'ALGORITMO + UNA CHIAVE + ESEMPI DI TESTI CIFRATI DEVONO ESSERE INSUFFICIENTI A DETERMINARE L'ALTRA CHIAVE.

IDEA: QUANDO ALICE

1: RIESCE A DETERMINARE L'AUTENTICITA' DEL MESSAGGIO SCAMBIAZO SUL CANALE NON SICURO.

• LA CHIAVE PUBBLICA PERMETTE DI AVERE

→ SEGRETEZZA
AUTENTICAZIONE
AUTENTICAZIONE E SEGRETEZZA (MA PRIMA AUTENTICAZIONE).

2: LA CRIPTOGRAFIA A CHIAVE PUBBLICA PERMETTE AD ALICE DI STABILIRE UN CANALE CONFIDENZIALE (SEGRETO) CON BOB.

► DI COSA ABBIAMO BISOGNO X FARE CRIPTOGRAFIA A CHIAVE PUBBLICA:
 • DEVE ESSERE COMPUTAZIONALMENTE FACILE PER B OBTENERE LA COPPIA DI KEY.
 • DEVE ESSERE COMPUTAZIONALMENTE FACILE PER A CONOSCERE PUB_B PER CRIPTARE M
 $C = E(PUB_B, M)$

• DEVE ESSERE " " FACILE PER B DECRYPTARE C CON PR_B
 $M = D(PR_B, C) = D(PR_B, E(PUB_B, M))$
 • DEVE ESSERE COMPUTAZIONALMENTE INFATTIBILE PER UN AVVERSARIO
 • DA PUB_B DETERMINARE PR_B
 • DA PUB_B E C DETERMINARE M

(• UTILE MA NON NECESSARIO: LE DUE CHIAVI POSSONO ESSERE SCAMBIALI)
 $M = D(PUB_B, E(PR_B, M)) = D(PR_B, E(PUB_B, M))$

TABELLA FICA

ALGORITMO	ENCRYPTION/DEC.	DIGITAL SIGNATURE	KEY EXCHANGE
RSA	✓	✓	✓
CURVA ELLITICA	✓	✓	✓
DIFFIE-HELLMAN	✗	✗	✓
DSS (DIGITAL SIGNATURE ALGORITHM)	✗	✓	✗

ONE WAY FUNCTION

- $f: X \rightarrow Y$ IS A ONE-WAY FUNCTION,
 - IF f IS "EASY" TO COMPUTE FOR ALL $x \in X$
 - f^{-1} IS "HARD" TO COMPUTE

A TRAPDOOR ONE WAY FUNCTION

- IS A $f: X \rightarrow Y$ ONE-WAY, WHERE GIVEN K (TRAPDOOR INFO) E' POSSIBILE TROVARE PER $y \in \text{Im}(f)$, UN $x \in X$ CON $f_K(x) = y$

PUBLIC KEY CRYPTOANALYSIS

BRUTE FORCE ATTACKS

↳ SOLUZIONE: CHIAVI + GRANDI

↳ PROBLEMA: E' ONEROSSO CRIPPIARE CON CHIAVI + GRANDI

↳ IMPLICAZIONE: CHIAVE PUBBLICA CONFINATA TRA KEY MANAGER

E FIRMA DIGITALE.

COMPUTARE CHIAVE PRIVATA DA PUBBLICA

↳ NESSUNA PROVA CHE SIA INFATTIBILE

ATTACCO A UN MESSAGGIO (FATTIBILMENTE DECIFRABILE)

↳ SUPPONIAMO DI AVERE UN MESSAGGIO CORTO M CRIPPIATO CON PU

↳ ATTACCANTE COMPUTA TUTTI GLI $Y_i = E(PU_A, X_i) \cdot H(X_i)$ PUNT

E SI FARÀ APPENA $Y_i = C$ (SNIFFATO).

↳ SOLUZIONE: APPENDERE BIT RANDOM A M .

PRIME FACTORIZATION

↳ MOLTIPLICARE I NUMERI E' FACILE, FATTORIZZARE I NUMERI SEMBRA DIFFICILE

RSA

→ LA SICUREZZA DERIVA DALLA DIFFICOLTÀ DI FATTORIZZARE GRANDI NUMERI.

→ LE CHIAVI SONO FUNZIONI DI UNA COPPIA DI NUMERI PRIMI CON ≥ 100 CIFRE.

→ È IL PIÙ COMUNE DOGLI VANTAGGI.

→ RISOLVE IL PROBLEMA DELLA GESTIONE DELLE CHIAVI

→ SVANTAGGIO:

- BASSE PRESTAZIONI → È PESANTE

- HA BISOGNO DI CHIAVI GRANDI (512 bit sono pochi, 1024 sono ragionevoli)

- VULNERABILE A PLAINTEXT E TIMING ATTACK

COME FUNZIONA, I PASSI:

1. GENERAZIONE DELLE CHIAVI

- GENERO 2 NUMERI PRIMI DISTINTI $p \neq q$ (con ≥ 100 digits)

- COMPUTO $n = pq$ e $\phi = (p-1)(q-1)$

- SCELGO e t.c. $1 < e < \phi$, ED È RELATIVAMENTE PRIMO CON ϕ

- COMPUTO $d = e^{-1} \bmod \phi$

- PUBBLICO (e, n) E MANTENGO (d, n) PRIVATA, SCARPO $p \neq q$

2. CRIPTAZIONE

- DIVIDO M IN BLOCCHE $M_1 M_2 \dots M_k$

- COMPUTO $C_i = M_i^e \bmod n$

3. DECRYPTAZIONE

- COMPUTO $M_i = C_i^d \bmod n$

→ È INFATTIBILE ARRIVARE A d DATO (e ED n)

→ È DIFFICILE QUANTO LA FATTORIZZAZIONE

→ DATO IL PROCESSO ATTUALE NEL FATTORIZZARE,

n DEVE AVERE ALMENO 1024 bit

→ DI PROGRESSI NELLA TEORIA DEI NUMERI POSSONO RENDERE RSA INSICURO.

IL PROBLEMA DI DISTRIBUZIONE DELLE CHIAVI

→ IDEA: USARE GLI ALGORITMI A CHIAVE PUBBLICA PER SUPPORTARE QUELLI + VELOCI A CHIAVE SIMMETRICA.

→ CON RSA

- DEVO CRIPTARE m E HO LA CHIAVE PUBBLICA (e, n)

- SCELGO K RANDOM

- $c = (K^e \bmod n, E_K(m))$

- DEVO DECRYPTARE c CON (d, n)

- DIVIDO c IN (c_1, c_2)

- $K = c_1^d \bmod n \quad m = D_K(c_2)$

→ ESEMPIO SSL

→ PROBLEMA: SE (d, n) È COMPROMESSO POSSO RECUPERARE K DAL TRAFFICO PRECEDENTEMENTE SNIFFATO!

↳ CON DIFFIE HELLMAN

SI SCAMBIANO
q e un PRIMITIVE ROOT α

(A)

GENERA $X_A < q$

CALCOLA $Y_A = \alpha^{X_A} \mod q$

Y_A

(B)

GENERA $X_B < q$

CALCOLA $Y_B = \alpha^{X_B} \mod q$

CALCOLA $K = (Y_A)^{X_B} \mod q$

Y_B

CALCOLA $K = (Y_B)^{X_A} \mod q$

↳ A DIFFERENZA DI RSA VI E'

PERFECT FORWARD SECRECY

↳ UNA CHIAVE DI SESSIONE NON E' COMPROMESSA SE
UNA CHIAVE PRIVATA VIENE SCOPERTA IN FUTURO.

↳ A DIFFERENZA DI RSA, SI PUO' FARE ANCHE PER UN GRUPPO (3+)

↳ DEBOLEZZA DI DIFFIE-HELLMAN LA CHIAVE SARA' FORMATA DA
 Y_A, X_B, P_C

↳ LE CHIAVI NON SONO AUTENTICATE

↳ POSSIBILE MAN IN THE MIDDLE ATTACK

(A)

SCEGLIE α, q

GENERA $X_A < q$

CALCOLA $Y_A = \alpha^{X_A} \mod q$

$Y_A[\alpha, q]$

(M)

GENERA $X_M < q$

CALCOLA $Y_M = \alpha^{X_M} \mod q$

$K_1 = (Y_A)^{X_M} \mod q$

GENERA $X_B < q$

CALCOLA X_B

$Y_B = \alpha^{X_B} \mod q$

$K_2 = (Y_M)^{X_B} \mod q$

Y_M

GENERA
 $K_2 = (Y_B)^{X_M} \mod q$

Y_B

CALCOLA

$K_1 = (Y_H)^{X_A} \mod q$

↳ CON MASSEY-OHURA

↳ CRIPTAZIONE SENZA CHIAVI CONDIVISE, BASATO SUL PROBLEMA DEL
LOGARITMO DISCRETO

↳ PRINCIPALI SI SCAMBIANO UN PRIMO p E SCELGONO CASCUO UN
t.c. e, d mod $(p-1) = 1$

• $A \rightarrow B : m^e \mod p$

• $B \rightarrow A : m^{e \cdot eb} \mod p$

• $A \rightarrow B : m^{e \cdot ebda} \mod p (= m^{eb})$

• $B \rightarrow A : m^{e \cdot ebda \cdot db} \mod p (= m)$

► MESSAGE INTEGRITY

↳ I DATI NON POSSONO ESSERE ALTERATI O ABBIANO GLI STRUMENTI X CAPIRLE SE SONO STATI ALTERATI.

↳ NEI SISTEMI OPERATIVI → ACCESS CONTROL

↳ NELLE RETI APERTE → DOBBIAMO USARE LA CRIPTOGRAFIA

IDEA: FUNZIONE DI HASH X CREA UNA IMPRONTA DEL MESSAGGIO

↳ DEVE AVERE LE PROPRIETÀ DI:

- ① COMPRESSIONE → h MAPPA X DI LUNGHEZZA ARBITRARIA IN $h(x)$ Dⁿ bit
- ② COMPUTABILE IN TEMPO POLINOMIALE

↳ $h(x)$ È ANCHE CRIPTOGRAFICA SE

- PRE-MESSAGE-RESISTANCE → ③ È ONE-WAY → DATO y È DIFFICILE COMPUTARNE UN X CON $h(x) = y$
- 2nd-PRE-MESSAGE-RESISTANCE → ④ È DIFFICILE TROVARE UN INPUT CON LO STESSO HASH
- ⑤ COLLISION RESISTANCE: È DIFFICILE ESISTERE DUE INPUT CON LO STESSO HASH

↳ UNA APPLICAZIONE È LO STORE DELLE PASSWORD

↳ È RICHIESTA SOLO LA ③ POICHÉ LA PW È COMBINATA CON IL SALT.

↳ UN'ALTRA APPLICAZIONE SONO GLI HASH FIRMATI

↳ È RICHIESTA 2nd-PREIMAGE RESISTANCE e AUTENTICATED MAC

MODIFICATION
DETECTION
CODE

► MESSAGE AUTHENTICATION

↳ MESSAGE AUTHENTICATION CODE (MAC)

- MAC AND DIGITAL SIGNATURES SONO DUE TECNICHE X FARE MESSAGE AUTHENTICATION

- UN MAC ALGORITMO È UNA FAMIGLIA DI FUNZIONI HASH h_K PARTECIPATE DA UNA CHIAVE SEGRETA K.

↳ LESE h_K DEVONO ESSERE COMPUTATION-RESISTANT

QUESTA PROPRIETÀ CI DICE CHE DEVE ESSERE INFATTIBILE COMPUTARE DA ZERO O PIÙ COPPIE $(x_i, h_K(x_i))$ UNA NUOVA COPPIA $(x_i, h_K(x_i))$ PER UN NUOVO INPUT $x \neq x_i$

O DELLO D'USO:

(A) M, MAC
 $MAC = h_K(M)$

(Z) M', MAC'
ALTERNA MAC E M

(B) VERIFICA CHE $MAC' = h_K(M')$

COSA GARANTISCE

- MESSAGE AUTHENTICATION? SI, PER I MESSAGGI VERIFICATI
- NON-REPUDIATION? NO, NON SE K È SIMMETRICA (A e B CONDIVIDONO K)
- FRESHNESS? NO, NESSUNA DETECTION DI REPLAY MESSAGES

(Z) M, MAC

(B) ✓

↳ POSSIAMO PENSARE A DELLE REALIZZAZIONI IN DES

MESSAGE AUTHENTICATION 2

DIGITAL SIGNATURES

M, S MESSAGGI E FIRME, SA FIRMA DA PARTE DI A CON IL SUO SEGRETO
 V_A VERIFICA FIRMA, VERIFICA CHE SIA LA FIRMA DI A
 $S = Sa(m)$ E TRASMETTE (m, s)

* FIRMA: A FIRMA $m \in M$, $s = Sa(m)$ E TRASMETTE (m, s) . ACCETTA IL

* VERIFICA: B VERIFICA LA FIRMA DI A CON $m = Va(m, s)$. ACCETTA IL

MESSAGGIO SOLO SE

* REQUISITI DI SEGRETERIA
 E' DIFFICILE PER UN PRINCIPAL $\neq A$ TROVARE PER UN $m \in M$,

UN $s \in S$ t.c. $V_A(m, s) = \text{true}$

L'PUO' ESSERE OVVIAMENTE BASATA SULLA CRITTOGRAFIA
 (REVERSIBILE) A CHIAVE PUBBLICA

$$\hookrightarrow \text{se } Dd(Ee(m)) = Ee(Dd(m)) = m$$

$$\underline{\underline{M=C}}$$

\hookrightarrow COSTRUIAMO UNO SCHEMA X LA FIRMA
 CON LA CHIAVE PUBBLICA.

① LET $M \in C$, lo spazio dei messaggi e delle firme, con $M = C$

② (e, d) è la coppia di chiavi per lo schema a chiave pubblica

③ DEFINISCI LA FUNZIONE DI FIRMA SA PER ESSERE Dd , $s = Dd(m)$

④ V_A È $V_A(m, s) = \begin{cases} \text{true} & Ee(s) = m \\ \text{false} & \text{OTW} \end{cases}$

ATTENZIONE QUESTO SCHEMA AVIETTE UN FOLLOW ON ATTACK

① UN ATTACCANTE B SELEZIONA S RANDOM E COMPUTA $m = Ee(s)$

② VISTO CHE $M = C$ PUO' INVIARE (m, s)

③ LA VERIFICA RITORNA TRUE ANCHE SE A NON HA FIRMATO m !

A
 A vuole FIRMANE m
 $s = Dd(m)$
 ↓ PRIVATA DI A

FORGERY

m, s

B
 B vuole VERIFICARE CHE m SIA DI A,
 { true $Ee(s) = m$ ✓
 { false OTW → PUBBLICA DI A ✓

B
 NUOVO FARSI PASSARE
 PER A
 • s' ER S
 • $m' = Ee(s')$

m', s'

{ true $Ee(s) = m$ ✓
 { false OTW ✓

SOLUZIONE: H DOVE H SONO I MESSAGGI FIRABILI
 → SUFFICIENTEMENTE PICCOLO CAPOLO! E' UN PROBLEMA!

LA VERIFICA DIVENTA $V_A(s) = \begin{cases} \text{true} & Ee(s) \in M \\ \text{false} & \text{OTW} \end{cases}$

LOPOI RECUPERO $m = Ee(s)$

DRSA PREVEDE UNA REALIZZAZIONE DELLE FIRME DIGITALI
 (LA COPPIA PUO' ESSERE CRISTATA PER GARANTIRE ANCHE CONFIDENZIALITA').

LA CRITTOGRAFIA SIMMETRICA / ASIMM.
 ASSICURA LA CONFIDENZIALITA'.

SOMMENTO LA SICUREZZA INCONDIZIONATA E CHIAVI DISTRIBUITE CORRISPONDENTEMENTE

LA CRITTOGRAFIA ASIMM. ASSICURA

SEMPLIFICA LA KEY DISTRIBUTION

LA MA C'E' COMUNQUE BISOGNO DI UN

KEY MANAGEMENT

LECT 8h

- KEY MANAGEMENT SI RIFERISCE A:
 - DISTRIBUZIONE DI CHIAVI CRITTOGRAFICHE
 - IL MECCANISMO DI ASSOCIARE UNA IDENTITÀ A UNA CHIAVE
 - LA GENERAZIONE E NECESSARIO MANTENIMENTO E REVOCO DELLA CHIAVE.

PKI → P. KEY INFRASTRUCTURES

↪ PERMETTONO DI RICONOSCERE IL PRO. DI UNA CHIAVE

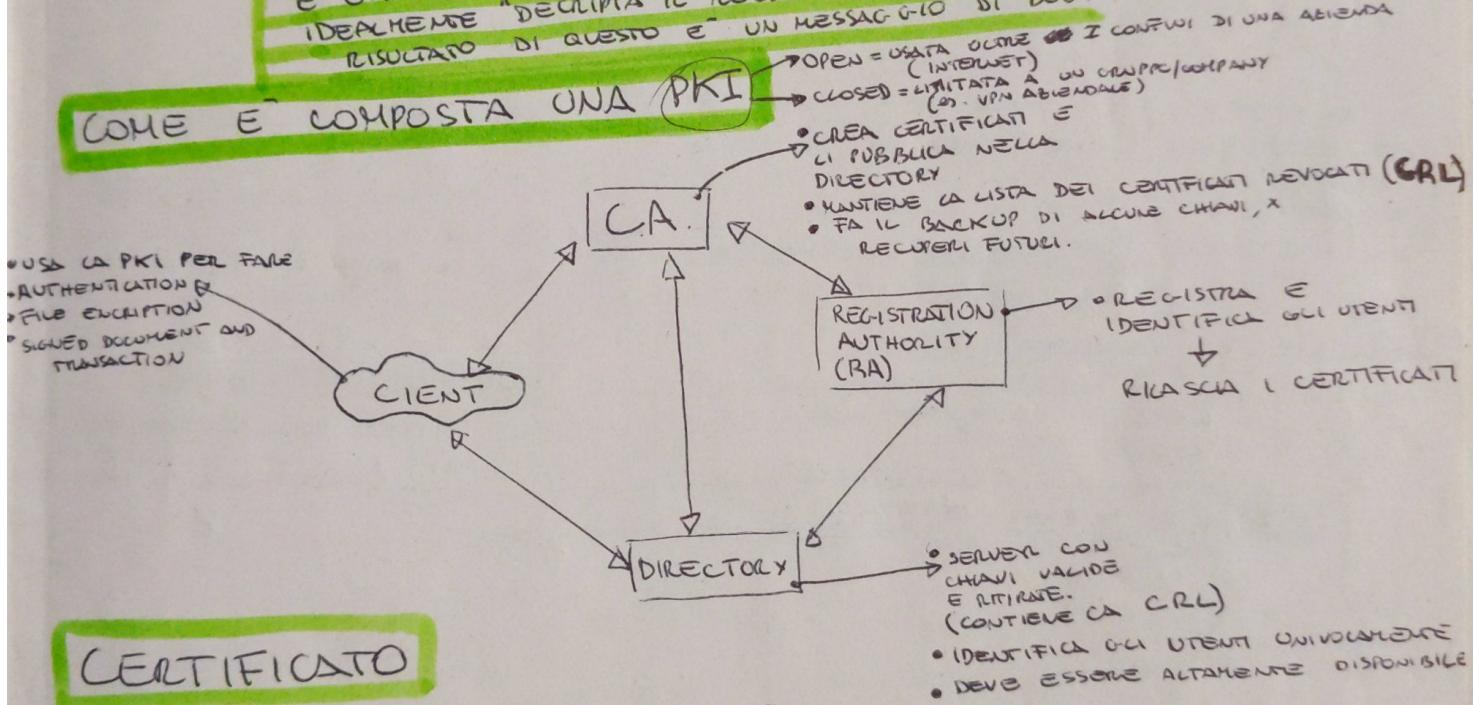
↪ PER FAIRE PARTE DI UNA PKI, ALICE

- GENERA CHIAVE PUBBLICA E PRIVATA
- CONSEGNA LA PRIVATA PUA A UNA C.A. CERTIFICATION AUTHORITY
 - LA C.A. VERIFICA CHE ALICE SA CHI DICE DI ESSERE
 - E POI EMISSIONE UN CERTIFICATO CHE DICE CHE:
PUA È ALICE

↪ IN EQUAL MODO
ALICE PUÒ VERIFICARE L'IDENTITÀ DI BOB TRAMITE UN CERTIFICATO REPERITO

↪ IDEA: BOB INVIA UN MESSAGGIO AD ALICE CON IL SUO CERTIFICATO,
ALICE DECRYPTA IL CERTIFICATO CON LA CHIAVE PUBBLICA DELLA C.A.
E OTTIENE LA CHIAVE PUBBLICA DI BOB, CON QUESTA CHIAVE
IDEALMENTE "DECRYPTA" IL MESSAGGIO ED È COSÌ SICURA CHE IL
RISULTATO DI QUESTO È UN MESSAGGIO DI BOB.

COME È COMPOSTA UNA PKI



CERTIFICATO

↪ È UN TOKEN CHE COLLEGA UNA IDENTITÀ A UNA CHIAVE, QUESTO TOKEN DEVE ESSERE
FIRMATO DALLA C.A.

$$\text{CACCIA} = M \parallel E(H(M), PR_{\text{CERT. AUTH.}})$$

↪ PROBLEMA, SE IL RICEVENTE DI UN MESSAGGIO DI ALICE NON CONOSCE
LA C.A? 2. APPROCCIO.

↪ COSTRUIRE UNA GERARCHIA DI CERTIFICATI. (LA ROOT È NOTA A TUTTI)

↪ ALTRO METODO BASATO SULLA FIDUCIA CHE CLASSE UNICO HA DI UNA C.A.

↪ X.509: STANDARD CHE DEFINISCE COME DEVE ESSERE FATTO UN CERTIFICATO.

↪ È BASATO SU CRITTOGRAFIA A CHIAVE PUBBLICA (RACCOMANDATO RSA), HASH E
FIRME DIGITALI.

↪ IL CUORE È IL CERTIFICATO A CHIAVE PUBBLICA ASSOCIAZO A OGNI
UTENTE.

COME È COMPOSTO X.509 (I CAMPI)

- VERSIONE DI X.509.
- SERIAL NUMBER: UNICO TRA TUTTI, O MEGLIO DEVE ESSERE UNICO LA COPPIA (NOME DELL'EMITTENTE, SERIAL NUMBER).
- SIGNATURE ALGORITHM IDENTIFIER: ALGORITMO USATO X FIRMARE IL CERTIFICATO
- ISSUER NAME: È IL NOME IN X.500 DELL'EMITTENTE (CA)
- PERIODO DI VALIDITÀ
- SUBJECT NAME: NOME DEL POSSESSORE (UTENTE) DEL CERTIFICATO, QUELLO A CUI APPARTIENE LA CHIAVE PUBBLICA.
- SUBJECT PUBLIC-KEY INFO: ALGORITMO, PARAMETRI E LA CHIAVE.
- SIGNATURE: CONTIENE L'HASH DEGLI ALTRI CAMPI, FIRMATO DALLA CA CON LA SUA CHIAVE PRIVATA.

SUPPOSIANO CHE UNA CA ~~REGGIA~~ FIRMA IL CERTIFICATO DI A:

$$CA(A) = MIE(PKA, H(M))$$

UDIL RICEVENTE X VERIFICARLE LA VALIDITÀ DEL CERTIFICATO FA L'HASH DEI CAMPI E VERIFICA CHE SIA LO STESSO HASH CHE È CONTENUTO NEL RELATIVO CAMPO DEL CERTIFICATO, PRECEDENTEMENTE DECRIPTATO CON LA CHIAVE PUBBLICA DELLA CA.

MODELLI DI FIDUCIA

UD FIDUCIA DIRETTA

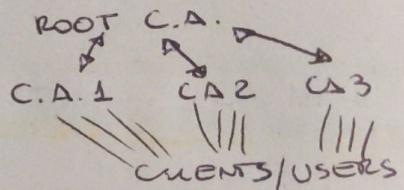
UD FIDUCIA GERARCHICA

UD WEB OF TRUST

TUTTI GLI UTENTI SONO ISCRITTI ALLA STESSA CIE UNA FIDUCIA COMUNE VERSO CA.

PER GRANDI COMUNITÀ DI UTENTI CI SONO PIÙ CA.

MODELLO:



UD COMPRENDE SIA IL DIRECT CHE IL GERARCHICAL
UD CERTIFICATO VIENE FIRMATO DA PIÙ UTENTI/ENTITÀ.

UD USATO NEL PGP (PRETTY GOOD PRIVACY)

PGP \neq X.509
CERTIFICATES CERTIFICATES

• 1: PGP KEY HA MULTIPLE SIGNATURES (ANCHE UNA SELF-SIGNING), UN UTENTE FIRMA SOLO I CERTIFICATI DI CHI CONOSCE.

• 2: L'E' LA NOZIONE DI TRUST (FIDUCIA) CHE È ESPlicita IN OGNI CERTIFICATO.
(UNO STESSO LEVEL DI TRUST PUÒ AVERNE SIGNIFICATO DIVERSO PER OGNI UTENTE)

IDEA: TI FIDI DELLA MIA OPINIONE SU ALTRE CHIAVI E' VALIDA, SOLO SE MI CONSIDERI AFFIDABILE, ALTRIMENTI LA MIA OPINIONE SULLA VALIDITÀ DI ALTRE CHIAVI NON È DA CONSIDERARE.

UD QUANDO FINO LA CHIAVE, INSERISCO ANCHE UN TRUST LEVEL

UD QUANDO SO CHE UNA CHIAVE È STATA COMPROMESSA, NON LA FIRMA O LA FIRMO METTENDO TRUST LEVEL A 0.

UD ABBIAMO DUNQUE UN MODO X REVOCARE CERTIFICATI

UD ABBIAMO DUNQUE UN REPUTATION SYSTEM (UN SISTEMA DOVE OGNIUNO HA UNA REPUTAZIONE AI SUOI "VICINI")

UD. IN CUI A VOGLIE PARLARE CON B E A SI FIDA DI X1 MA X2 NON CONOCE B

X1 << X2 >> X2 << B >>

CHAIN OF CERTIFICATES

UD IDEA DELLA CROSS-CERTIFICA
LE CA. SI SONO SCAMBiate LE CHIAVI PUBBLICHE E SI CONDIVISCONO DI LORO.

UD X.509 SUGGERISCE CHE LE CA POSSANO ESSERE DISPOSTE IN UNA GERARCHIA.

UD ABBIAMO I FORWARD CERTIFICATES (CERTIFICATI DI X FIRMATI DALLE ALTRE CA)

UD "REVERSE CERTIFICATES"

(X CERTIFICA FINO ALLE ALTRE CA)

UD IDEA: LE CATENE NON SONO LUNGHESSIMO MASSIMO 40 NODI E RAGGIUNGE TUTTO INTERNET

UD C'E' UN CONCETTO DI TRUST MA NON E' ESPlicito NEL CERTIFICATO

KEY / CERTIFICATE REVOCATION AND RECOVERY

CRL (CERTIFICATE REVOCATION LIST) FIRmate e mantenute dalla CA.

• CRL (CERTIFICATE REVOCATION LIST) FIRmate e mantenute dalla CA.

↳ PUBBLICATE SULLA DIRECTORY.

↳ I CLIENTS CONTROLLANO LE CRL PERIODICAMENTE O USANO UN VALIDATION SERVICE

↳ CHE LI CONTROLLA IN MODO CENTRALIZZATO.

↳ OGNI CA MANTIENE UNA LISTA DEI CERTIFICATI REVOCATI MA NON DI QUELLI SCADUTI (IL PERIODO DI VALIDITÀ È SPICATO NEL CERTIFICATO)

RAGIONI PER LA REVOCA

↳ LA PRIVATE KEY SI ASSUME ESSERE STATA COMPROMESSA

↳ L'UTENTE NON È PIÙ CERTIFICATO DALLA CA.

↳ IL CERTIFICATO DELLA CA. SI ASSUME ESSERE STATO COMPROMESSO

Ogni CRL CONTIENE:

• IL NOME DELL'EMITTENTE

• LA DATA IN CUI LA CRL È STATA CREATTA

• LA DATA IN CUI LA PROSSIMA CRL VERRÀ RILASCIATA

• UNA ENTRY PER OGNI CERTIFICATO REVOCATO

IDEA: SALVARE LA CHIAVE PRIVATA SU UN SERVER PUÒ ESSERE PERICOLOSO MA NON SALVARLA PUÒ ESSERE ANCHE DI PIÙ.

↳ KEY ESCROW SYSTEM È UN SISTEMA CHE PERMETTE A UNA TERZA PARTE DI RECUPERARE UNA CHIAVE.

↳ PER QUESTIONI DI RENDITA DELLA STESSA

↳ PER QUESTIONI LEGALI (RICHIEDA DA PARTE DI UNA AUTORITÀ).

NAMING AND IDENTITY

↳ UN PRINCIPAL È UNA IDENTITÀ UNIVOCÀ

↳ UNA PKI FA IL BINDING DELLA CHIAVE PUBBLICA DI ALICE CON IL SUO NOME.

↳ UN PROBLEMA: COME IDENTIFICARLO IN MODO UNIVOCO ALICE?

CI SONO MOLTI MODI, MA È DIFFICILE, ES INDIRIZZO MAIL

È UNIVOCO MA NON CONTIENE ABBASTANZA INFO SUL SOGGETTO.
(INOLTRE GLI INDIRIZZI EMAIL POSSONO ESSERE CONDIVISI)

↳ USIAMO UN INSERIMENTO DI INFO: NOME, DATA DI COMPLEANNO, (CAMBIANO NEL TEMPO)
ADDRESS...

↳ USIAMO UN NUMERO UNIVOCO: ES
NUERO DI CODICE FISCALE (NON È GLOBALE...)

↳ DOBBIAMO AVETE DELLE POLICIES PER L'ASSEGNAZIONE DEI NOMI
E PER LA RISOLUZIONE DEI NOMI

→ E QUINDI? CHE SI FA? L'X.509 UTILIZZA NOMI DISTINTI SECONDO LE SPECIFICHE.

↳ QUESTI NOMI DISTINTI VENGONO USATI SIA PER IDENTIFICARE GLI EMITTENTI (CAs) CHE GLI UTENTI (SOGGETTI)

↳ C'È BISOGNO DI DUE CRITERI FONDAMENTALI:

↳ PERMANENZA (RISPETTO AL PASSATO) → NON CONTIENE INFO VOLUTA

↳ UNICITÀ (È UN AND DI INFORMAZIONI) → UNICI NEL TEMPO

LE CAs GARANTISCONO A UN CERTO LIVELLO LE IDENTITÀ DEI PRINCIPAL A CUI EMETTONO UN CERTIFICATO!

• CA AUTHENTICATION POLICY: DESCRIVE IL LIVELLO DI AUTENTICAZIONE RICHIESTO DA UNA CA. PER ASSICURARSI CHE UN PRINCIPAL SIA PROPRIO CHI DICE DI ESSERE.

• CA ISSUANCE POLICY: DESCRIVE I PRINCIPAL VERSO I QUALI UNA CA. RILASCIERÀ IL CERTIFICATO. DATA UNA IDENTITY LA CA. RILASCIERÀ UN CERTIFICATO A TALE PRINCIPAL?

I PROTOCOLLI

OBBIETTIVO: METTERE ASSIEME DELLE PRIMITIVE DI CRITTOGRAFIA
PER OTTENERE DEI PROTOCOLLI SICURI E COLMARE
QUELLE MANCANZE DEI SINGOLI BLOCCI.

LEZ. 03

- RSA } CRITTAZIONE E DISTRIBUZIONE IN MA RICHIEDONO
DIFFIE HELLMAN } CANALI INSICURI SENZA PROBLEMA AUTENTICAZIONE
- FIRME DIGITALI } GARANTISCONO MA NON LA TIMELESS
L'AUTENTICAZIONE DEL MESSAGGIO
(REPLAY-ATTACK POSSIBILE)

COME CREARE UNA COMUNICAZIONE SICURA?

LA PRIMA POSSIBILITÀ: APPLICAZIONI SICURE SU CANALI INSICURI

USAREPGP PER CRITTAZIONE/FIREWALL LE MAIL

USAREKERBEROS X SUPPORTARE DIVERSE APPLICAZIONI

LA ALTRA POSSIBILITÀ: LA SICUREZZA SUGLI ALTRI LAYER È ANCORA POSSIBILE

PROTOCOLLO = INSIEME DI REGOLE CHE CONSENTONO LO SCAMBIO DI MESSAGGI TRA
DUE O PIÙ PRINCIPALI.

L'È UN ALGORITMO DISTRIBUITO CON ENFASI SULLA COMUNICAZIONE.

PROTOCOLLO CRITTOGRAFICO = USA MECCANISMI CRITTOGRAFICI PER RAGGIUNGERE
GLI OBIETTIVI DI SICUREZZA

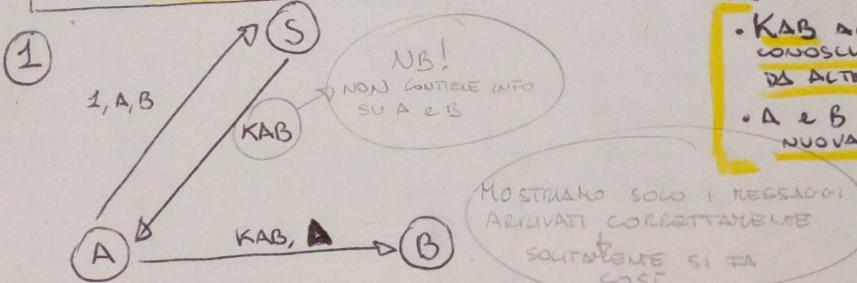
COSTRUIRE UN PROTOCOLLO PER STABILIRE UNA CHIAVE

- **SECURAMO UNO SILENZIO**
 - UN INSIEME DI UTENTI, A 2 A 2 HANNO UNA CHIAVE DI SESSIONE
 - UN SERVER (SICURO)

IL RUOLO DI S È SOLITAMENTE QUELLO
DI GENERARE UNA CHIAVE DI SESSIONE
TRA A E B KAB (E COMUNICARLA IN
MODO SICURO A entrambi)

OBIETTIVI:

- KAB ALLA FINE DEL PROTOCOLLO SARÀ CONOSCUTA DA A E DA B (MA NON DA ALTRI) CON POSSIBILE ECCEZIONE DI S.
- A E B DEVONO SAPERE CHE KAB È NUOVA GENERATA.



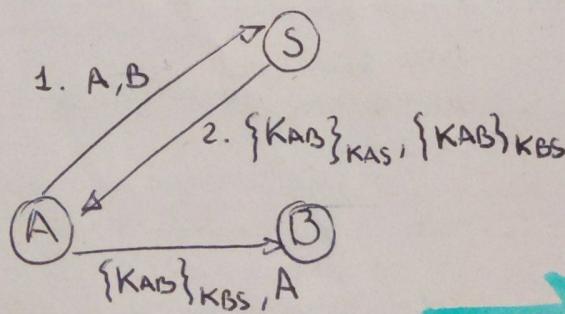
• PROBLEMA = KAB DEVE ESSERE COMMUNICATA AD A E A B MA NON AD ALTRE PARTI.

SECURITY ASSUMPTION 1

L'AVVERSARIO È CAPACE DI ORIGINARE TUTTI I MESSAGGI INVIAI IN
UN PROTOCOLLO SICURO.

THE ADVERSARY HAS THE COMPLETE CONTROL
OVER THE NETWORK

- ASSUMMO CHE S INIZIALMENTE CONDIVIDA UNA CHIAVE CON TUTTI I PARTECIPANTI



CONTROLLO
SU APPUNTI
(CONTINUO)

SECURITY ASSUMPTION 2

L'ATTACCANTE È CAPACE DI ALTERARE TUTTI I MESSAGGI USANDO TUTTE LE INFO DISPONIBILI.

LUI PUÒ INOLTRE RIDIREZIONARE UN MESSAGGIO A QUALUNQUE ALTRO PRINCIPAL (O ANCHE SEMPLICEMENTE INTERCETTARLO). QUESTO INCLUDE L'ABILITÀ DI GENERARE E INVIARE MESSAGGI COMPLETAMENTE NUOVI.

PERFECT CRYPTOGRAPHY ASSUMPTION

I MESSAGGI CRIPARI POSSONO ESSERE LETTI DAL LEGITTIMO RICEVENTE CHE HA LE CHIAVI PER DECRIPTARLO.

SECURITY ASSUMPTION 3

L'AVVERSARIO PUÒ ESSERE UN LEGITTIMO PARTECIPANTE (UN INSIDEN), UN ESTERNO (OUTSIDER) O ENTRAMBI.

SECURITY ASSUMPTION 4

L'AVVERSARIO È CAPACE DI OBTENERE KAB USATA IN UNA "SUFFICIENTEMENTE VECCHIA" ESECUZIONE DEL PROTOCOLLO.

[ATTACCHI E PROTOCOLLI
SU FOGLI SEPARATI ...]

L'IDENTITÀ DI UN INDIVIDUO DEVE AVERE
-PERMANENZA (IL RISPETTO AL PASSATO)
-UNITÀ (È UN AND DI INFORMAZIONI)

LEC05

LAYERED SECURITY AND SECURITY PROTOCOLS

DIFFIE HELLMAN → È PRIVO DI AUTENTICAZIONE

05/06

- LE PRIMITIVE DI CRITTOGRAFIA POSSONO ESSERE COMBINATE
- IMPORTANTE E' IMPORTANTE PER CIO' PROTOCOLLO RICORDARE QUALE PROPRIETÀ IL PROTOCOLLO SI FESTA A SODDISFARE.
- USEREMO LA NOTAZIONE ALICE & BOB

PREAMBOLI

- IL CANALE DI COMUNICAZIONE PUÒ ESSERE LOGICO O FISICO
- IL PROTOCOLLO DI SICUREZZA PUÒ STARE/SI PUÒ IMPLEMENTARE IN UN LIVELLO O NELL'ALTRIO, OPPURE SI PUÒ IMPLEMENTARE A PIÙ LIVELLI.
↳ DEL TCP/IP (O DELL'ISO/OSI)

NON MI ACCORDO DI USARE UN SOLO LAYER DI SICUREZZA.

INVECE

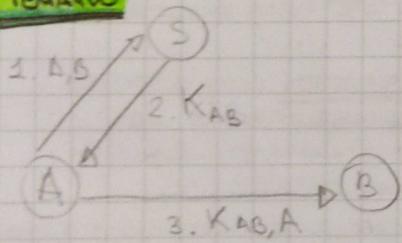
COS'È E COME SI FA UN PROTOCOLLO? → UN PROTOCOLLO CI SEMBRA SICURA, POI TUTTI CHIAMA I PROBLEMI E METTENDO DELLE PEGGIO

PROTOCOLLO: INSOMMA DI REGOLE CHE DETERMINANO LO SCAMBIO DI MESSAGGI TRA LE PARTI,
O SEMPRELLAMENTE UN ALGORITMO DISTRIBUITO CON ENTRAMBI SUA COMUNICAZIONE

↳ DI PROTOCOLLI DI SICUREZZA USANO ALGORITMI CRPTOGRAFICI

- CREARE UN ACCORDO SULLE CHIAVI → PROPRIETÀ (OGGETTO)
- (2 PRINCIPALS E UN DISTRIBUTORE DI CHIAVI → SERVER S)
 - CHIAVE CONFIDENZIALE, CONOSCIUTA DA A, B E DAL SERVER CHE L'HA GENERATA.
 - CHIAVE SIA FRESH (NON UNA VECCHIA)
 - CHIAVE SIA "AUTENTICATA"

1° TENTATIVO



- NON MI INTERESSA SE UN MESSAGGIO VIENE MAL PERCEPITO O NON ARRIVA, DI QUESTE COSE (ACK...)
- SE NE OCCUPANO ALTRI LIVELLI
- ASSUMO CHE L'ALTRO PRINCIPALE SA DISPONIBILE A RICEVERE I MESSAGGI, CHE ABbia L'ALGORITMO ATTIVO A ricevere la mia comunicazione.

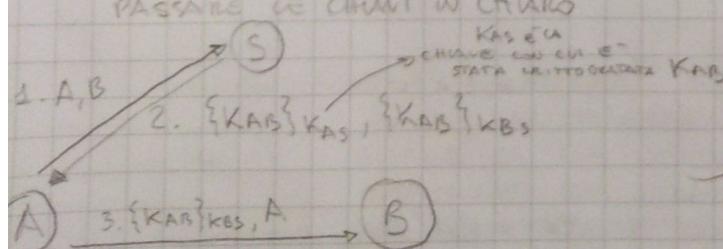
↳ RAPPRESENTATO IN ALTRO MODO:

1. A → S: A,B
2. S → A: KAB
3. A → B: KAB,A

→ CHIUNQUE SI METTE IN MEzzo CARPISE LA CHIAVE KAB CHE PASSA IN CHIARO

2° TENTATIVO

LO CERCO DI NON FAR PASSARE LE CHIAVI IN CHIARO



IL NOME DEL RICEVENTE
MISTE' NEL MESSAGGIO

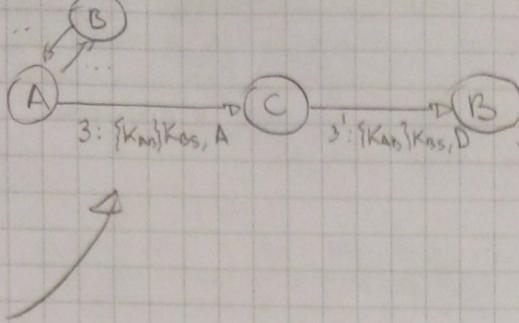
→ UN QUARTO CHE SI METTE IN MEzzo NON PUO' CARPIRE LA CHIAVE PRIVATA

Ma ci sono infiniti modi con cui un attaccante può violare il tuo protocollo

DA NON PUO' DECIFRARE KAB? KBS POICHÉ SEMPRE LA CHIAVE PRIVATA DI B

↓
CI SONO ANCORA PROBLEMI!
MAN IN THE MIDDLE!

CHARLIE (L'ATTACCANTE FA QUESTO)

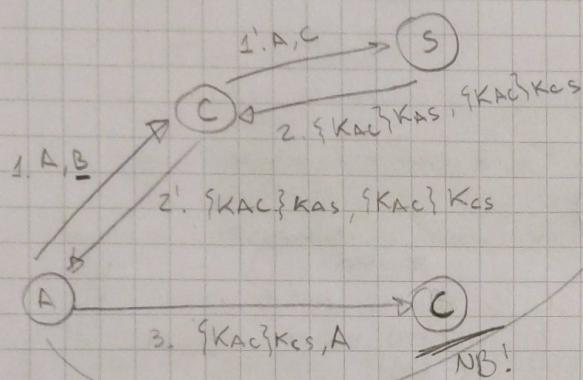


PROBLEMA: B CREDÉ DI AVERE UNA CHIAVE
PER PARLARE IN MODO SICURO CON D E NON
CON A.

D È SOTTO IL CONTROLLO DELL'ATTACCANTE
E FA PUNTA DI ESSERE A.

1° MODO PER
METTERSI IN MEZZO

2° MODO PER METTERSI
IN MEZZO



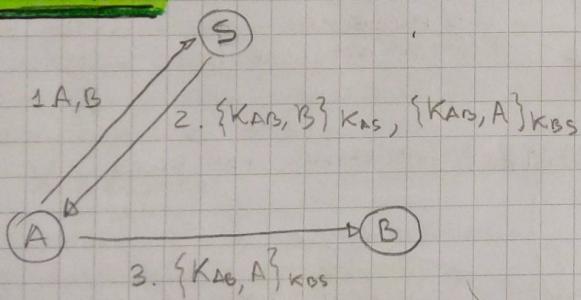
SI USA PER FARE DENY OF SERVICES
(POICHÉ IN REALTA' C E B NON POSSONO
LEGGERE I MESSAGGI DI B: LA CHIAVE È
COMUNQUE SEGRETA E NON VIOLATA).

CREDE DI AVERE UNA COMUNICAZIONE
SICURA (TRAMITE UNA CHIAVE SEGRETA)
CON B!

E' BEN PIÙ GRAVE DI PRIMA, POICHÉ
ORA C PUÒ CHIUDERE A FARLE B E
LEGGERE TUTTI I MESSAGGI SEGRETI DI A.
(CHE A CREDÉ DI INVIERE A B).

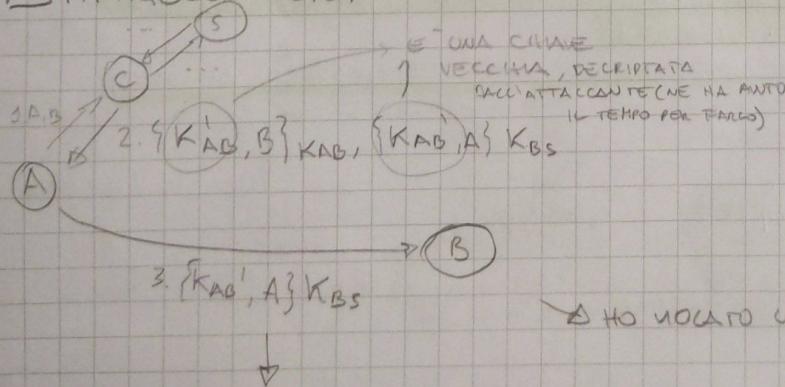
IDEA X POSSIBILE RISOLUZIONE: IL
SERVIZIO DEVE INCLOCCHIARE IN MODO CIFRATO
L'INFORMAZIONE CHE LA CHIAVE È PER
C E NON PER B.

3° TENTATIVO



ORA L'ATTACCO DI PRIMA NON
E' PIÙ POSSIBILE, UN ATTACCANTE
NON PUÒ MODIFICARE IL MESSAGGIO FINO AL
CONCUSSO POICHÉ MODIFICANDO UN BIT
IL MESSAGGIO CIFRATO NON PUÒ
PIÙ ESSERE DECIFRATO.

ATTACCO REPLY



PROBLEMA: FAR E' QUESTO TIPO
DI OPERAZIONE E' PESANTE, POICHÉ
LA CHIAVE E' CHIAVE PUBBLICA PRIMATA LONG
E POSSO TRASMETTERE ALLORA DUE
CHIAVI DI SESSIONE E UNA VOLTA
CHE HO PASSATO LA CHIAVE DI
SESSIONE TRAMITE IL CANALE CIFRATO
A B, A E B COMUNICHERANNO
TRAMITE QUESTA CHIAVE.

HO VOLUTO UNA SECRETARIA

- AUTENTICAZIONE (POICHÉ IN REALTA'
QUESTA CHIAVE E' PASSATA DA C)

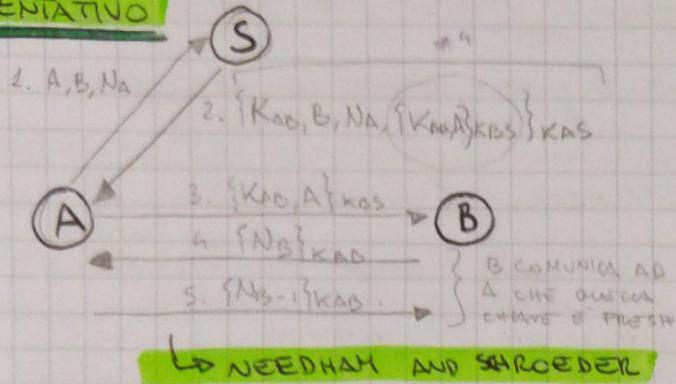
IDEA DI RISOLUZIONE: TENERE LE CHIAVI SEGRETE

→ TROPPO PESANTE

- ATTACCO ALLA CHIAVE UN TIMESTAMP CHE ME INDICA QUANTO VECCHIA
E QUESTA CHIAVE: NONCE

- IN UN REPLAY ATTACK COSA E' ANDATO male? CHE LA RISPOSTA E' VECCHIA (LA RISPOSTA E' ATTACCARE ALLA CHIAVE UN DATETIME RAPPRESENTATO DAL NOSTRO NONCE). IL NONCE LO ATTACCO GI' AL MESSAGGIO DI RICHIESTA COSÌ QUANDO VEDO RITORNARE IL MESSAGGIO DAL SERVER SO CHE LA RISPOSTA E' PROPIA RELATIVA ALLA RICHIESTA CHE HO FATTO "POCO FA".

4° TENTATIVO



• IL NONCE NON SOLO MI ASSICURA CHE LA CHIAVE E' FRESH MA MI ASSICURA CHE IL PACCHETTO PER B E' FRESH.

↳ PER QUESTO TACCO, QUESTA COSA E'
E NON QUESTA

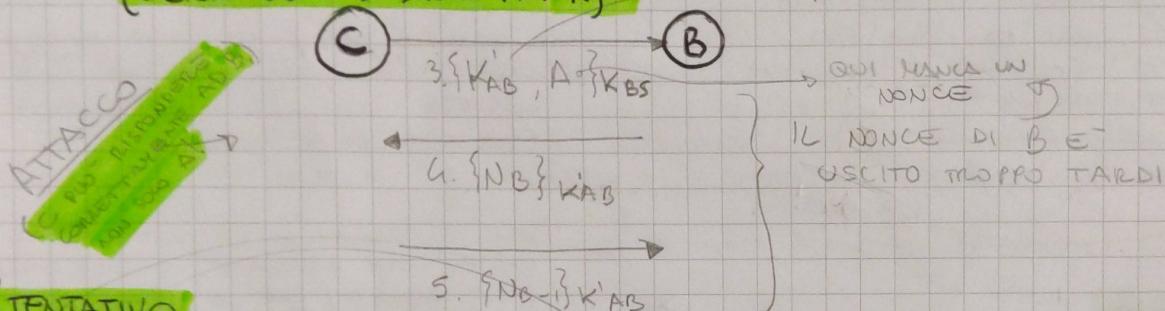
{KAB, B, Na} KAS, {KAB} KBS

↳ QUESTO HA FUNZIONATO PER ALCUNI FINCHE' QUALCUNO E' RIUSCITO A FARNE UN MAN IN THE MIDDLE.

(DENNING AND SACK ATTACK)

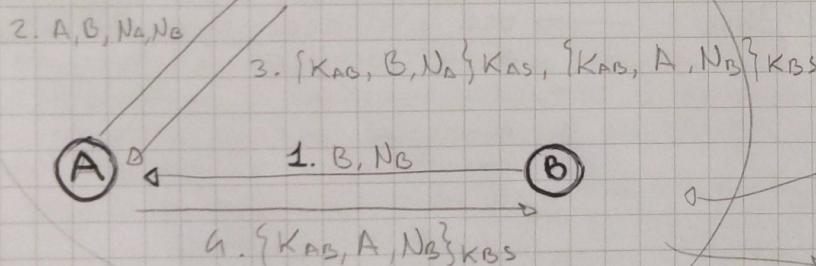
IL MAN IN THE MIDDLE C

L'HA DESCRIPPIATA IN MIEI DI LAVORO



5° TENTATIVO

SOLUZIONE:
(FINALE)



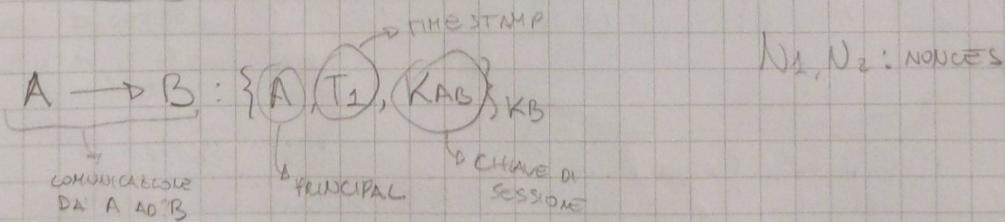
UN'ULTIMA COSA
CHE POSSO ANCORA AGGIUNGERE UN ACK

PROBLEMA A NON SA SE B ABbia EFFETTIVAMENTE RICEVUTO IL MESSAGGIO

IDEA: I NONCES SONO LA SOLUZIONE AI REPLAY ATTACK!
MA VANO USATI BENE!

(RIESCO A EVITARE I REPLAY ATTACK CON MENO MESSAGGI DI PRIMA)

NOTAZIONE:



N₁, N₂: NONCES

LA COMUNICAZIONE
CHE TRAVERSO E'
ASINCRONA

OUVIAMENME

SENDER e RECEIVER NON SONO PARTE
DEL MESSAGGIO SE NON SPECIFICATO!

► PROTOCOLLO NSPK (NEEDHAM-SCHROEDER PUBLIC KEY PROTOCOL)

↳ CONSENTE UNA SORTA DI PING AUTENTICATO

1. $A \rightarrow B : \{NA, A\}^K_B$
 2. $B \rightarrow A : \{NA, NB\}^K_A$
 3. $A \rightarrow B : \{NB\}^K_B$
- } NON C'È INTERAZIONE CON IL SERVER

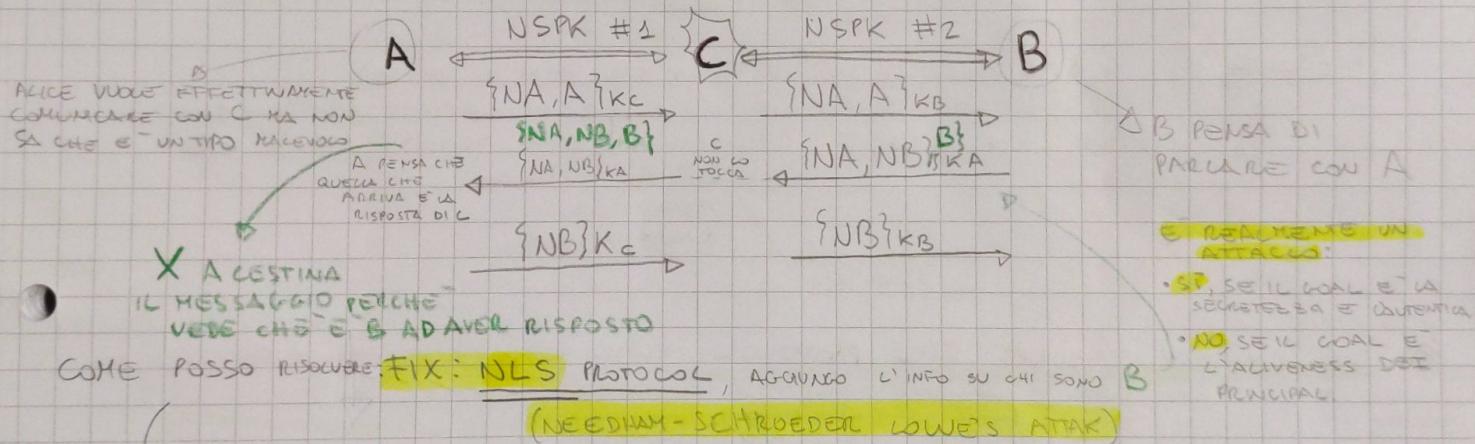
COSA DEVE GARANTIRE QUESTO PROTOCOLLO:

- MESSAGGI AUTENTICATI
- IL MESSAGGIO DEVE ESSERE FRESH (TIMELINESS)
- SE UNA TERRA (NON TRASMETTO NULLA DI SEGRETO MA SE POSSONO VIOLARE NONCES POTREBBERE VIOLARE L'AUTENTICAZIONE DEI MESSAGGI)

↳ GLI ATTACANTI POSSONO ESSERE ATTIVI O PASSIVI

↳ QUANDO PARLAVAMO DI DOLEV-YAO ATTACKER MODEL SI PARLA DI UN ATTACCANTE CHE PUÒ FAIRE QUALESiasi COSA SULLA RETE. È LA DEFINIZIONE DELL'ATTACCANTE PIÙ PUENTE, MA ATTENDERE NON PUÒ VIOLARE LE CHIAVI SE NON SONO CHIAVI VECCHIE!

① -> ATTACCO MAN-IN-THE-MIDDLE A NSPK (LOWE'S ATTACK)



② -> ATTACCO TYPE-FLAW ATTACK

↳ CONTIENE UN ORACLE ATTACK: ATTACCO CHE PREVEDE UN'ISTANZIAMENTO DEL PROTOCOLLO CON UN PRINCIPAL PER FARSI DARE INFORMAZIONI UTILI A FAR E UN ATTACCO.

→ TIPI DI ATTACCHI → ERRORE SLIDE! NON È IL PARAGRAFO SESSION ATTACK

- MAN IN THE MIDDLE: QUALCUNO STA IN MEZZO TRA DUE PRINCIPALI E FA QUALCOSA PER INVALIDARE IL PROTOCOLLO.
- REPLY ATTACK
- MASQUERADE ATTACK → UN ATTACCANTE FA FINTA DI ESSERE QUALCUNALTRO
- REFLECTION ATTACK → RITORNARE ALL'ORIGINE INFORMAZIONI CHE VENGONO TRASMESSE
- ORACLE ATTACK
- TYPE FLAW ATTACK

• PARALLEL ATTACK: BISOGNA ESSERE MAN IN THE MIDDLE MA BISOGNA
CUCIARE PIÙ RUN DEL PROTOCOLLO PER SPOSTARE INFORMAZIONI
RICEVUTE DA UNA RUN SULL'ALTRA, TRAMITE QUESTA OPERAZIONE
VIENE STUDIATO UN ATTACCO.

L'OBBIETTIVO DI TUTTI I PROTOCOLLI CHE CI SARANNO
E CAPIRE I VARI TIPI DI ATTACCO

10/04

► THE OTWAY-REES PROTOCOL (AUTENTICATED KEY DISTRIBUTION WITH)

PROPRIETÀ: KEY AUTHENTICATION, FRESHNESS

(FATTA DAL SERVIZIO)

E SEGRETERIA DELLE CHIAVI

M₁: A → B : I, A, B, {N_A, I, A, B} K_{AS}

M₂: B → S : I, A, B, {N_A, I, A, B} K_{AS}, {N_B, I, A, B} K_{BS}

M₃: S → B : I, {N_A, K_{AB}} K_{AS}, {N_B, K_{AB}} K_{BS}

M₄: B → A : I, {N_A, K_{AB}} K_{AS}

CHIAVE DI SESSIONE

GARANZIA

DI FRESHNESS

MAN IN THE MIDDLE TRA A e B

Z(B) → SIGNIFICA CHE Z SI MASCHERA
DA B

• PRIMA TIP. DI ATTACCO (REFLECTION / TYPE-FLAW)

M₁: A → Z(B) : I, A, B, {N_A, I, A, B} K_{AS}

M₄: Z(B) → A : I, {N_A, I, A, B} K_{AS}

{K_{AB}} = |I, A, B|

• B NON È STATO CONVINTO

• A PENSA DI AVER SCAMBIAZI LA CHIAVE CON B (MA NON È VERO)

(PERDENDO AUTENTICAZIONE, SEGRETERIA) NON PERDO LA FRESHNESS

• SECONDA TIP. DI ATTACCO (MAN IN THE MIDDLE TRA B e IL SERVER)

L'ATTACCALENTE RIESCE AD AVERE IL COMPLETO CONTROLLO DELLA
(VEDI SULLE SLIDE) COMUNICAZIONE CHE AVVIENE TRA A e B

FALLISCE:

- SEGRETERIA

- AUTENTICAZIONE

K_{AB}

(A e B HANNO GIÀ UNA CHIAVE DI SESSIONE)

► THE ANDREN SECURE RPC PROTOCOL → COAL: ~~SECRETARIA FUORI~~ (SULLE SLIDE)

(SOTTO DI PING AUTENTICATO)

M₁: A → B : A {N_A} K_{AB}

M₂: B → A : {N_A+1, N_B} K_{AB}

M₃: A → B : {N_B+1} K_{AB}

M₄: B → A : {K_{AB}, N_B} K_{AB}

• ATTACCO TYPE-FLAW (MAN IN THE MIDDLE TRA A e B)

M₃: A → Z(B) : {N_B+1} K_{AB}

M₄: Z(B) → A : {N_A+1, N_B} K_{AB}

{FUNZIONA POGLI METACOLPOSS
DEGLI ACCORTI!}

- LA CHIAVE È NON È PIÙ AUTENTICATA

- LA SEGRETERIA È INVECE ANCORA PRESERVATA

{QUALE È LO SVANTAGGIO CREATO
DALL'ATTACCALENTE? DOS!

A CREDÈ DI AVERE PORTATO A
TERMINE IL CAMBIO DI CHIAVE CON B
E DUNQUE A INizia A USARE UNA
NUOVA CHIAVE MA B NON CÒ L'HA
(B NON RIESCE PUÒ DECRIPPIARE I
MESSAGGI DI A = LA COMUNICAZIONE SI
INFERISCE IMMEDIATAMENTE)

► KEY EXCHANGE WITH CA (DENNING & SUCCO)

DOVE ESSERCI:

- SEGRETERIA

- AUTENTICAZIONE

M₀: A → S: A, B
 M₁: S → A: C_A, C_B
 M₂: A → B: C_A, C_B {{T_A} K_{AB}}_{K_A} }_{K_B}

→ CERTIFICATI (NOME, CHIAVE PUBBLICA...)

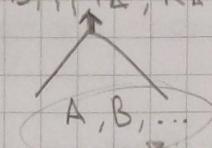
→ PUÒ LIMITARE L'USO DELLA CHIAVE DI SESSIONE

- ATTACCO (MAN IN THE MIDDLE)

A vuole parlare con Z (AUTENTICATO CHE PENSÀ NON RISPETTA IL PROTOCOLLO)

RIMANGONO
UNICO

M₀: A → S: A, B
 M₁: S → A: C_A, C_Z
 M₂: A → B: C_A, C_Z {{T_A, K_{AZ}}_{K_A} }_{K_Z}
 M₃: A → Z: C_A, C_Z {{T_A, K_{AZ}}_{K_A} }_{K_Z}
 M₄: Z → B: C_A, C_B {{T_A, K_{AZ}}_{K_A} }_{K_B}



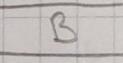
C_A e C_B VENGONO
TRASFERITE PER TRASPORTE
LE INFO SULLE
CHIAVI PUBBLICHE.
 C = {B_B, K_B}_{K_S}

→ DADESSO ZOE PUÒ SIMULARE ALICE
 E BOB INVIERÀ A ZOE LE INFORMAZIONI
 CHE INVECE CREDÈ DI INVIERE
 A ALICE (BOB CREDÈ CHE
 ZOE SIA ALICE).

COME POSSIAMO RISPARMIARE?
 BASTA AGGIUNGERE LE PARTI!

► ALTRO ESEMPIO (ATTACCO DI BINDING) SU UN ALTRO PROTOCOLLO

M₀: A → S: A, B, NA
 M₁: S → A: S, {S, A, NA, K_B}_{K_S}



FIX PER
RISPARMIARE

M_{1.1}: A → Z(S): A, B, NA
 M_{2.1}: Z(A) → S: A, Z, NA
 M_{2.2}: S → Z(A): S, {S, A, NA, K_Z}_{K_S}
 M_{1.2}: Z(S) → A: S, {S, A, NA, K_Z}_{K_S}

• L'ATTACCANTE SI SPACCIA
PER B

SRUTTARE UN
PRINCIPAL PER
DARE UN'IDEEA
CHE L'ATTACCANTE
NON SAVERE
FARE

► ESEMPIO DI PARALLEL SESSION ATTACK (WITH ORACLE) (SUCCÈ SLIDE)

UTILIZZA DUE RUN, L'ATTACCANTE SI SPOSTA DA UNA PARTE
ALL'ALTRA IN BASE ALLE NECESSITÀ.

► ESEMPIO DI REPLAY ATTACK (CAUSATO DALL'USO NON CORRETTO DEL NONCE DA PARTE DI B).

(SUCCÈ SLIDE)

TIMESTAMP = TEMPO DI VALIDITÀ DELL'INFO TRASMESSA
 (es DELLA CHIAVE DI SESSIONE TRASMESSA)

BUONE REGOLE PER FAR CRIPTOGRAFIA

- IL MESSAGGIO DEVE CHIARAMENTE INDICARE TUTTO (Se ad esempio trasferisco la chiave tra A e B, il messaggio deve contenere chiaramente A e B)
- L'IDENTIFICAZIONE DUNQUE I PARTECIPANTI È SPICITAMENTE NEL MESSAGGIO
- CHE SIA CHIARO PERCHÉ LA CRIPTOGRAFIA VIENE FATTA
- CHE SIA CHIARO PERCHÉ (PER VERIFICARE QUALI PROPRIETÀ) VENGONO USATI I NONCES.
- DEVE ESSERE CHIARO QUALI SONO LE RELAZIONI DI TRUST (DI FIDUCIA), CHI PUÒ FIDARSI DI CHI,

AGGIUNGiamo QUESTA NOTAZIONE:

$\{M\}_K \rightarrow M \text{ è CRIPTATO (CON } K\text{) ED È SEGRETO}$

$[M]_K \rightarrow M \text{ è CRIPTATO MA NON È SEGRETO (SERVE PER AUTENTICAZIONE O PER VERIFICARE L'INTEGRITÀ DEI DATI)}$

NSPK CON IL FIX DI LOWE (ANCHE CON IL FIX DI LOWE C'È CHI IL TYPE FLOW)

$$\begin{aligned} A \rightarrow B : & \left\{ \left[N_A, A \right]_{K_A^{-1}} \right\}_{K_B} \\ B \rightarrow A : & \left\{ N_A, \left[N_B \right]_{K_B^{-1}} \right\}_{K_A} \\ A \rightarrow B : & \left\{ \left[N_B \right]_{K_A^{-1}} \right\}_{K_B} \end{aligned} \quad \left. \right\} \text{ CON QUESTO NON FUNZIONA PIÙ MENO IL TYPE FLOW ATTACK}$$

PER DIMOSTRARE CHE UN PROTOCOLLO È SICURO NON DEVO SOLO DIMOSTRARE CHE È SICURO A SE STANTE MA SE "COLLABORA" CON L'ALTRO ALGORITMO DEVO FAR VEDERE CHE ENTRAMBI GLI ALGORITMI QUANDO COLLABORANO NON SONO ATTACCABILI.

• Protocolli di sicurezza

LECT - Ø7

ALL'INTRO AUTHENTICATION, ACCOUNTING E DODGE

Kerberos = PROTOCOLLO X L'AUTENTICAZIONE IN AMBIENTI APERTI e DISTRIBUITI.

L'obiettivo è l'autenticazione in ambienti aperti e distribuiti.

Ha le seguenti caratteristiche:

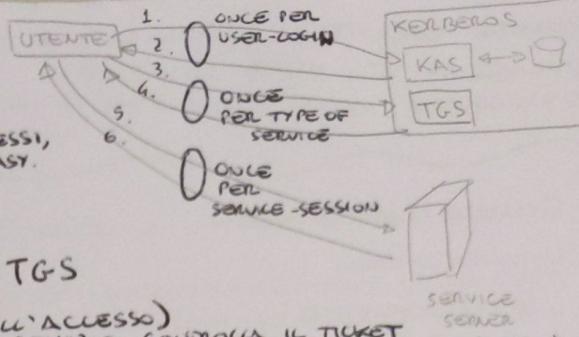
- sicuro: un attaccante in ascolto non può impersonare un utente del sistema
- affidabile: tutti i servizi dipendono dal controllo effettuato da Kerberos per l'accesso, deve essere molto affidabile ed essere in grado di supportare architetture distribuite
- trasparente: ogni utente dovrebbe inserire una singola password per accedere ai vari servizi ma questa cosa viene nascosta dal sistema (come SSO) → SINGLE SIGN-ON
L'UTENTE NON DEVE RENDERSI CONTO DEI PROTOCOLLI CHE STANNO SOTTO
- scalabile: supportare numerosi utenti, essere modulare, distribuito
SCALABILE NEL NUMERO DEGLI UTENTI CHE K. DEVE GESTIRE!

L'architettura di Kerberos si compone di due server: KAS, l'authentication server, e TGS.
il ticket granting server. Inoltre è presente un access control che controlla il ticket emesso dal TGS.

Le connessioni si distinguono in 3 fasi tutte basate su Needham and Schroeder:

ED IN KERBEROS USIAMO I ~~NONCES~~⁶ TIMESTAMP INVECE DEI NONCES.

- LIMITAZIONI DI KERBEROS IV
 - ENCRYPTION NON SERVE MA UN ATTACCANTE PUÒ VENDARE IL KAS: A → KAS: A, TGS
 - LA DOPPIA CRIPTAZIONE È RIDONDANTE, ELIMINATA IN KERBEROS 5
 - SI BASE SU HOST (E DUNQUE CLOCK) NON COMPROMESSI, SE I CLOCK SONO COMPROMESSI FAKE REPLAY È EASY.



- Autenticazione → USING KAS
- Autorizzazione ad un certo servizio → TGS
- Connessione al servizio (CONTROLO DELL'ACCESSO)
IL SERVER DEL SERVIZIO CONTROLLA IL TICKET

La fase di autenticazione viene svolta una volta ed ha una durata di svariate ore. La durata è stabilita in base a quando consideriamo scaduto il timestamp incluso nel pacchetto. Supponiamo A voglia connettersi per utilizzare un servizio.

1. $A \rightarrow KAS : A, TGS$
 2. $KAS \rightarrow A : \{K_{A,TGS}, TGS, T_1, \{A, TGS, K_{A,TGS}, T_1\}_{KKAS,TGS}\}_{KAKAS}$
- KAS → DERIVA DALLA PASSAROLA DELL'UTENTE

Ottenuta la chiave per comunicare con TGS potrà richiedere il servizio S (di cui il TGS dispone). Dovrà compiere questa fase tutte le volte che vorrà richiedere un servizio, per evitare di appesantire il sistema e per garantire la trasparenza del sistema il timestamp ha una scadenza di pochi secondi.

3. $A \rightarrow TGS : \{A, TGS, K_{A,TGS}, T_1\}_{KKAS,TGS}, \{A, T_2\}_{K_{A,TGS}}, S$
4. $TGS \rightarrow A : \{K_{A,S}, S, T_3, \{A, S, K_{A,S}, T_3\}_{KS,TGS}\}_{K_{A,TGS}}$

1. LA BREVE VALITÀ DEI MESSAGGI (SECONDI) EVITA I REPLAY ATTACK
 2. PER EVITARE I REPLAY ATTACK PIÙ IMMEDIATI, I SERVER SALVANO GLI AUTENTICATOR PIÙ RECENTI

Finalmente ha la chiave per il server cui intendeva accedere e ci si conterà nel modo seguente:

5. $A \rightarrow S : \{A, S, K_{A,S}, T_3\}_{KS,TGS}, \{A, T_4\}_{K_{A,S}}$
6. $S \rightarrow A : \{T_4 + 1\}_{K_{A,S}}$

La scalabilità di Kerberos è dovuta al fatto che, qualora avessi bisogno di distribuire i servizi, avrà differenti TGS e differenti Access Control. La divisione viene detta a reami ovvero ogni TGS conosce i servizi e a che reame appartengono e chi è il TGS in quel reame. Qualora un utente volesse richiedere un servizio ad un TGS che non è nel suo reame, si vedrebbe rispondere una chiave col TGS corretto pertanto l'utente dovrebbe rifare la fase 2 con il TGS corretto che stavolta darà la chiave per il servizio.

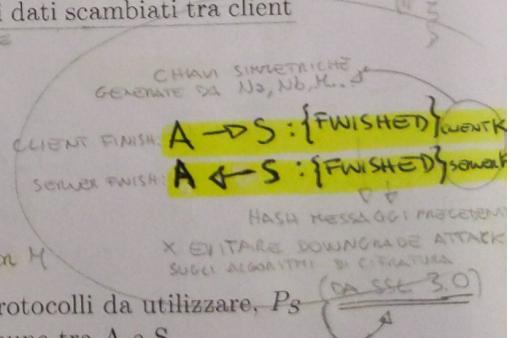
TLS/SSL (TRANSPORT LAYER SECURITY PROTOCOLS AND SECURE SOCKET LAYER)

SSL e TLS si trovano a livello di trasporto, non vengono gestiti dal sistema operativo ma dall'applicazione. Il livello di trasporto tradizionalmente garantiscono solo l'affidabilità ma i messaggi sono in chiaro e possono essere modificati. L'obiettivo è quello di fornire privacy ed integrità dei dati tra due applicazioni che comunicano. Il primo è stato SSL che poi si è evoluto in TLS che è diventato uno standard di comunicazione.

Consiste in due protocolli uno di handshake e uno di record. Quello di handshake utilizza un meccanismo di crittografia a chiave asimmetrica, quello di record usa le chiavi segrete stabilite nel handshake per proteggere confidenzialità, integrità e autenticità dei dati scambiati tra client e server.

Protocollo di handshake:

- * FIX A SSL 2.0
- NEL SECRET A INTEGRA L'INFO SULLA VERSIONE DI SSL CHE VOLVERE USARE, COSÌ IN CASO DI DOWNGRADE ATTACK IL SERVER SE NE ACCORGE.
- CHE APPLICAZIONI SONO I PROTOCOLLI E LE VERSIONI PIÙ ALTE DEI PROTOCOLLI E ALGORITMI IN COMUNE TRA A E S.
- CLIENT HELLO :** $A \rightarrow S : A, N_A, S_{id}, P_A$
- SERVER HELLO :** $A \leftarrow S : N_S, S_{id}, P_S$
- NON PREFERITI**
SONO PIÙ ANTI DI QUELLI DI A FIRMATI DAL CA.
- SERVER CERTIFICATE :** $A \leftarrow S : certificate(S, K_S)$
- CERTIFICATE (OPTIONAL) :** $A \rightarrow S : certificate(A, K_A)$
- CLIENT KEY EXCHANGE :** $A \rightarrow S : PMS_{K_S}$ → PRE MASTER SECRET PER H
- CERTIFICATE VERIFY (OPT.) :** $A \rightarrow S : hash(\dots), K_A \rightarrow$ ALL PREV. MESSAGES
- DOVE P_A È UNA LISTA DI PREFERENZE DI A RIGUARDO GLI ALGORITMI E I PROTOCOLLI DA UTILIZZARE, P_S SONO I PROTOCOLLI E LE VERSIONI PIÙ ALTE DEI PROTOCOLLI E ALGORITMI IN COMUNE TRA A E S.



Dove implementiamo la sicurezza, in quale layer?

• SSL (o TLS/SSH) : SO NON CAMBIA, LE APPLICAZIONI SI (SSL API)

• IPSEC : L'OS CAMBIA, LE APPLICAZIONI NO

• LO SPazio deve essere implementato nel supportare IPSEC

→ SSL PUÒ RIFIUTARE CHE TCP ACCETTA ⇒ EASY DOS ATTAK

- IMPLEMENTAZIONE DA SICUREZZA COME VEDUTO SU QUALSIASI PROTOCOLLO UTILE
- DOPO IMPLEMENTAZIONE LO

- NO MODIFICHE AL SO.
- MINIME MODIFICHE ALLE APP.
- - EASY DOS ATTAK

- NON CAMBIA APPLICAZIONI
- AUTENTICO SOLO IP, NON UTENTI
- BISOGNA COMBINARE API E OS

APPLICAZIONI	OS	APPLICAZIONI
SSL	TCP	
TCP	IP SEC	
IP	IP	
:	:	

- HTTPS PREVEDE ENCRYPTION, AUTHENTICATION E INTEGRITY CHECKING. (SOLITAMENTE PER IL SERVER)
 - ↳ PROBLEMA DEL MIXED CONTENT
 - ↳ DOWNGRADE DELLA CONNESSIONE A HTTP

Gestione chiavi private

Le possibilità sono:

- hardware apposta
- crittografata ma ogni reboot è necessario inserire la password
- in chiaro sul server

La metodologia ad oggi più utilizzata è quella di tenerle in chiaro sul server.

IPsec (IMPLEMENTA LA SICUREZZA SU IP) → CANALE SICURO X TUTTE LE APPLICAZIONI
DOVE ESSERE INSTALLATO SU SO E GATEWAYS.
USATO PER IMPLEMENTARE LE VPN.

I pacchetti di livello IP non sono cifrati quindi sarebbe possibile fare analisi di traffico, non c'è integrità e confidenzialità. IPsec garantisce la confidenzialità cifrando i dati, l'integrità con dei checksum e autenticazione con firma e certificati.

Si compone di 3 moduli:

- Authentication header - AH: garantisce l'autenticazione, successivamente è stato inglobato in ESP
- Encapsulating Security Payload - ESP: garantisce la confidenzialità e l'integrità
- Key Management - IKE: si occupa dello scambio delle chiavi

IPsec è utilizzabile in due modalità ovvero la Tunnel mode o la Transport mode. Nella tunnel mode il pacchetto viene criptato e diventa una componente di un pacchetto nuovo che lo contiene. Questo meccanismo è molto utilizzato nelle VPN, è caratterizzato pertanto da 2 header IP intermezzati da un header IPsec, nel header IP più interno troviamo le informazioni relative alla rete della VPN mentre nel header IP esterno quelle reali che consentono di raggiungere l'entry point della VPN. Nella transport mode l'header IPsec viene messo prima dell'header IP e viene utilizzato principalmente per criptare la comunicazione.

L'Authentication Header si compone di: authentication data ovvero un MAC delle parti immutabili del messaggio, sequence number field per evitare replay attack ovvero è un contatore di pacchetti e l'ultimo campo è security parameters index che contiene la SA (Security Association) ovvero l'insieme di protocolli, algoritmi e versioni da utilizzare nella cifratura.

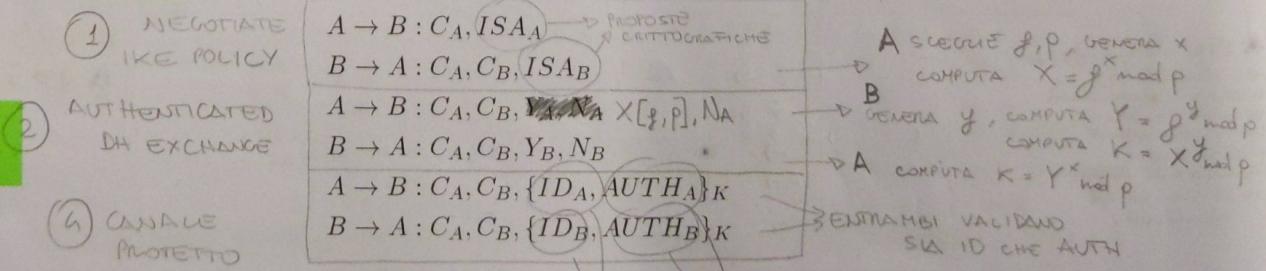
ESP cifra il payload, aggiunge un MAC e lo mette in coda al pacchetto.

NB

IKE

IKE stabilisce non solo le chiavi ma anche l'SA. È molto flessibile ma molto complesso. IKE è nata come evoluzione di Diffie-Hellman aggiungendo l'autenticazione e mantenendo la Perfect Forward Secrecy. Si divide in due fasi in base alla pregressa conoscenza o meno dei principal che vogliono scambiare chiavi.

La **Main mode** viene utilizzata quando due principal non si conoscono.



NELLA QUICK MODE:

- $A \rightarrow B : C_A, C_B, HASH(1), SA_A, Na, X^{[f, p]}, \{ID_A, ID_B\}$
 - $B \rightarrow A : C_A, C_B, HASH(2), SA_B, Nr, Y^{[f, p]}, \{ID_B, ID_A\}$ (ID È UN INSERIMENTO DI INFO UTILIZZATO A VALIDARE L'IDENTITÀ DELLE DUE PARTI).
 - $A \rightarrow B : C_A, C_B, HASH(3)$
- $HASH(1) = h(SKEY, ID_A, \{SA_A, Ni, X, ID_A, ID_B\})$
- $HASH(2) = h(SKEY, ID_B, \{SA_B, Ni, Nr, Y, ID_B, ID_A\})$
- $HASH(3) = h(SKEY, ID_A, \{O, Ni, Nr\})$

dove ISA_i è la proposta crittografica, ID_i è l'identificazione user/host, $AUTH_i$ è un MAC di tutta la conversazione.

La Quick mode viene utilizzata tra principal che si conoscono e condividono già una chiave. Per aggiornare le chiavi ho bisogno di mandare le mezze chiavi, per farlo i due principal si inviano: i cookie, le mezze chiavi nuove, le SA e gli ID il tutto corredato dall'hash di tutto il messaggio cifrato con la chiave vecchia. Successivamente viene fatto un ack con la chiave vecchia e solo quando ottengono la risposta attesa calcolano la nuova chiave, non lo fanno prima per assicurarsi che la comunicazione sia fresh.

IPSEC (cont).

↳ BENEFICI:

↳ CONFIDENTIALITÀ

- CRIPTA I DATI

INTEGRITÀ

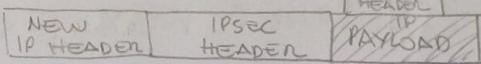
- I ROUTER E TUTTI QUELLI ALLA FINE DI UN CANALE CALCOLANO IL CHECKSUM

AUTENTICATION

- FIRME E CERTIFICATI
- MANTENENDO LA CAPACITÀ DI FAR ROUTE DI PACCHETTI IP

↳ IDEA: NELLE RETI LOCALI MANTENIAMO IL CLASSICO

MENTRE QUANDO USCIMO



↳ USA 3 PROTOCOLLI

→ AH (AUTHENTICATION HEADER)

PROTEGGE AUTENTICAZIONE E INTEGRITÀ DEI DATAGRAMMI IP
(MA NON FORNISCE CONFIDENTIALITÀ).

→ ESP (ENCAPSULATING SECURITY PAYLOAD)

FORNISCE CONFIDENTIALITÀ TRAMITE ENCRYPTION E
OPZIONALMENTE AUTENTICATION.

→ IKE (INTERNET KEY EXCHANGE)

KEY MANAGEMENT

↳ PER ISTITUIRE UN CANALE TRA DUE PRINCIPLE CHE NON SI CONOSCONO, OPPURE CHE SI CONOSCONO E VOGLIONO UN NUOVO CANALE.

IPSEC : SA (SECURITY ASSOCIATIONS)

↳ È UNA RELAZIONE ONE-WAY TRA UN SENDER E UN RECEIVER
CHE DEFINISCE I SERVIZI DI SICUREZZA

↳ SPECIFICA GLI AUTHENTICATION ALGORITHM (AH)

• ENCRYPTION ALGORITHM (ESP)

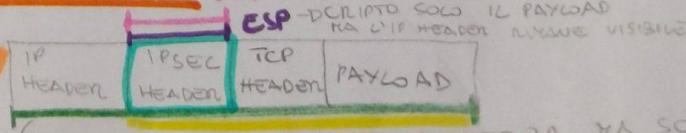
• KEYS, KEY LIFETIME

• PROTOCOL MODE (TUNNEL OR TRANSPORT)

↳ È UN CAMPO VERSO E PROPRIO IN AH/ESP HEADER

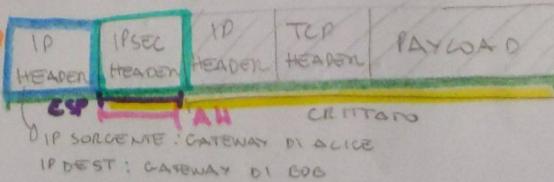
↳ L'SA È STABILITO USANDO IKE

2 MODI DI UTILIZZO → TRANSPORT MODE



(NON MI INTERESSA SEGRETERIA MA SOLO
AUTENTICATION, SI VIDE CHE IL PACCHETTO VA DA A A B)

→ TUNNEL MODE
(MOLTO USATO NELLE VPN)

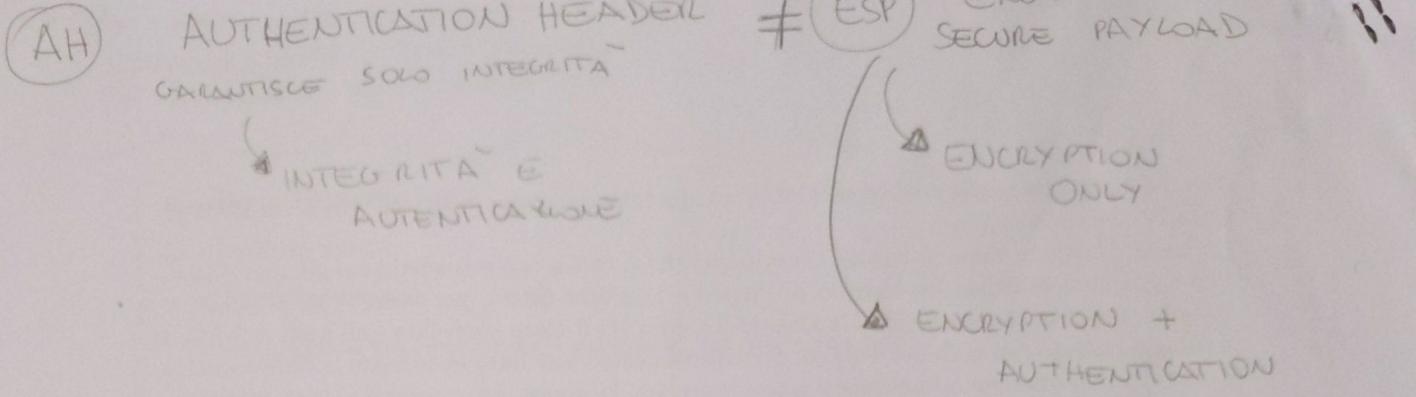


9

DIP SORGENTE: GATEWAY DI ALCHE
DIP DEST: GATEWAY DI BOB

● AUTENTICATA PER AH

● AUTENTICATA PER ESP



- IN ENTRAMBI GLI HEADER C'È UN CAMPO SPI (CHE IDENTIFICA (SA) PER IL DATAGRAMMA)
- NELL'HEADER AH C'È UN CAMPO CONTENENTE IL MAC DEL PACCHETTO (AD ECCEZIONE DEI CAMPI MUTABILI)

IKE (INTERNET KEY EXCHANGE)

- PROTOCOL/S FORMAT USED
- THE CRYPTOGRAPHIC AND HASHING ALGORITHM
- KEYS

FLEXIBLE → SUPPORTA VARI TIPI DI AUTENTICAZIONE (CON DIVERSI MASTER SECRETS)
BUT
COMPLEX

↪ E' NATO COME EVOLUZIONE DA DUE ALGORITMI GIÀ ESISTENTI
 ↪ E' BASATO (PILOTRIO COME I PREDECESSORI) SU DIFFIE HELLMAN

USO DIFFIE HELLMAN PER LA PERFECT FORWARD SECRECY (HO LA GARANZIA CHE SE UNA CHIAVE VIENE COMPROMESSA, NON PUOLO A RISALIRE A TUTTE LE ALTRE)

SSL E' UN ESEMPIO SENZA PFS (SE LA CHIAVE SEGRETA DEL SERVER VIENE SCOPERTA, TUTTE LE SESSIONI PASSATE POSSONO ESSERE DECRIPTATE).

IKE HA DUE FASI:

- PHASE 1 = LE DUE PARTI NEGOZIANO LA SA (DA USARE NELLA FASE 2)
- PHASE 2 = LA SA SCELTA NELLA FASE 1 E' ~~USATA~~ PER CREARE DELLA SAS FIGLIE PER FARLE ENCRYPTION E AUTENTICATION DELLE FUTURE COMUNICAZIONI

↪ FASE 1 PUO' ESSERE OFFERTA IN 2 MODI:

• MAIN MODE = 6 MESSAGGI SCAMBIAI TRA A & B] QUANDO A e B NON SI CONOSCONO ANCORA
 • OFFER PIROTEZIONE DI IDENTITA'

• AGGRESSIVE/QUICK MODE = 3 MESSAGGI] QUANDO A e B SI CONOSCONO GIÀ
 • SENZA PIROTEZIONE DI IDENTITA'

N.B! GUARDARE BENE A
 QUICK MODE

ESEMPI / TIPI DI ATTACCO

- MAN IN THE MIDDLE ATTACK: $A \xrightarrow{Z} B$
- REPLAY (o FRESHNESS) ATTACK: RIUSO PARTI DI MESSAGGI PRECEDENTI FAENDOLI PASSARLE PER FRESH
- MASQUERADEING ATTACK: UN ATTACCANTE Finge di essere un altro principale
- REFLECTION ATTACK: RITRASMETTE INDIETRO AL MITTENTE DELLE INFORMAZIONI
- ORACLE ATTACK: ISTANTIANDO UN PROTOCOLLO CON TERZI PRENDE VANTAGGIO DALLE RISPOSTE NORMALI A UN PROTOCOLLO PER TALE ENCRYPTION O DECRYPTION DI PARTI DI MESSAGGIO.
- TYPE FLAW ATTACK:
SOSTITUISCE DEI CAMPI DI UN MESSAGGIO CON ALTRI: SOLITAMENTE TENENDO CONTO DEI BYTES OCCUPATI DAL CAMPO.

WHY ANONYMITY IS DIFFICULT?

LEC-10

- TRAFFIC ANALYSIS
- PACKET HEADER IDENTIFICA I DESTINATARI

> UN ATTACCANTE PUÒ OSSERVARE DOVE VANNO I PACCHETTI OSSERVANDO IL TRAFFICO SUI NODI INTERMEDIAI!

↳ ANONYMITY → ANONYMITY È MIGLIOR QUANDO HO MOLTI UTENTI (TOR)
SET

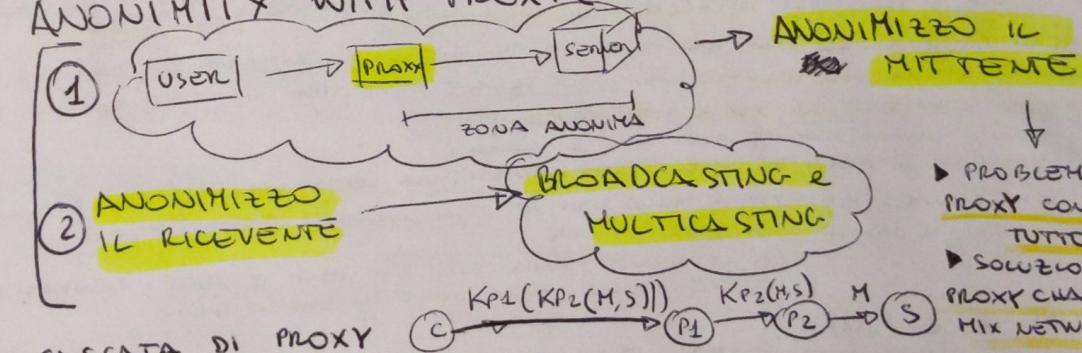


CLOUD = BETTER

> VOGLIAMO ANONIMIZZARE IL MITTENTE E IL DESTINATARIO

> USO DI PSEUDONIMI

ANONYMITY WITH PROXY (FOR SENDER)



CASCATA DI PROXY

- OGNI PROXY CONOSCE SOLO IL PREVIOUS/NEXT HOP
- LA TRAFFIC ANALYSIS È ANCORA POSSIBILE

- PROBLEMA: IL PROXY CONOSCE TUTTO
- SOLUZIONE: PROXY CHIAMS E MIX NETWORKS

- 1 - PUÒ CAPIRE ORIGINE O DEST. DI UN MESSAGGIO
- 2 - INJECT/MODIFY/REMOVE MESSAGGI
- 3 - NON PUÒ CREARE CORRISPONDENZA TRA MESSAGGI CAVIATI E PLANO

MIX NETWORKS

- COSTRUIRE UN CANALE ANONIMO IN PRESenza DI UN ATTACCANTE ATTIVO.
- ORIGINARIAMENTE PROPOSTO PER LE MAIL ANONIME

→ END-ON SERVER CHE PROCESSA MAIL

$$\langle R_1, \langle R_0, M \rangle K_B, B \rangle_{K_1} \rightarrow \langle \langle R_0, M \rangle K_B, B \rangle$$

► DURANTE DEL ~~MESSAGGIO~~ SINCE PROXY CON PADDING AND ENCRYPTION + SISTEMI PER IMPEDIRE LA TRAFFIC ANALYSIS:

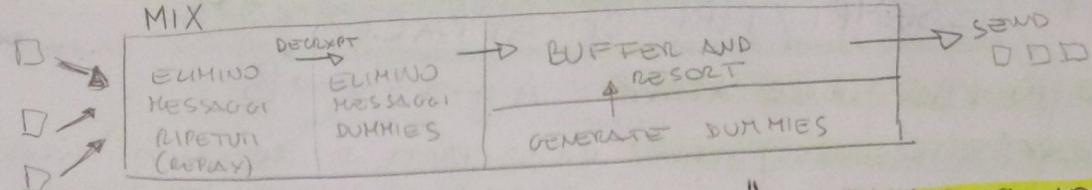
① OGGETTI DI MISURA DIVERSA → FIXED SIZE

③ PACCHETTI RIPETUTI BLOCCATI

→ ATTACCO REPLAY, INVIO IL PACCHETTO QUANDO C'È - TRAPOLLO PER SCOPPIARE INTO

② ORDINE DI ARRIVO NASCOSTO

④ È RICHIESTO UN TRAFFICO SUFFICIENTE



- CAPACITÀ DI GENERARE RICEVUTE DEL "MESSAGGIO RICONTO" → EVITARE LA NON REPUTATION
- STESSO PROBLEMA DEL PROXY: SE LO COMPROMETTO SO TUTTO
- DEBOLEZZA DIMINUITA SE METTIAMO MIX A CASCATA (IN UNA MIXNET)

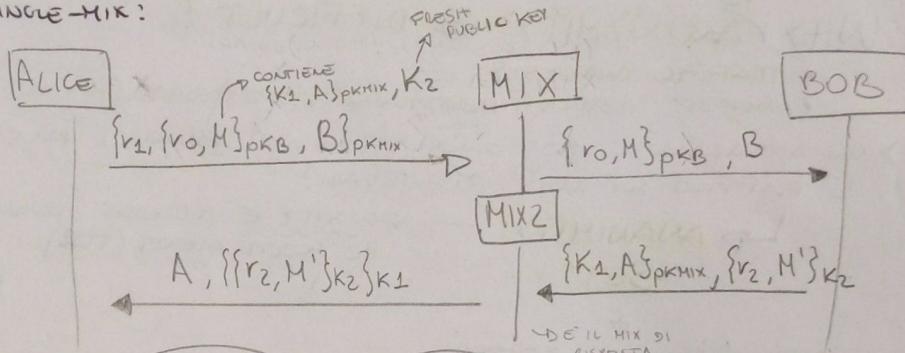
L'IDEA: ALICE SCEGLIE IL MIX-PATH PER INVIARE UN MESSAGGIO A BOB (NEL FARE QUESTO CIFRA I MESSAGGI NEL CORRETTO ORDINE PER OGNI MIX DEL PERCORSO).

✓ OGNI MIX TOGLIERÀ UN LIVELLO DI CRITTATORE, L'ULTIMO MIX PROCESSENZA IL PACCHETTO COME SE FOSSE UN SINGLE-MIX (CASO DI PRIMA)

- COME FA IL RICEVENTE A RISPONDERE?

✓ IL SENDER INSERISCE UN "RETURN ADDRESS" (IL SUO)

ESEMPIO SINGLE-MIX:



- COS'È L'ANONIMATO? È LO STATO IN CUI NON SI È INDIVIDUABILE IN UN INSIEME DI SOGGETTI.

LEC-10-BIS

↳ NASCONDERE LA PROPIA ATTIVITÀ TRA ALTRE ATTIVITÀ SIMILI (NON È COLLEGABILE UNA AZIONE A UNA IDENTITÀ)

1. (UN A SORVEGLIARE NON PUÒ NEVICOLARE DIRIGE SE UNA CERTA ATTIVITÀ HA AVUTO LUOGO O NUETO)
2. (NON È POSSIBILE CONOSCERE SE UNA CERTA ATTIVITÀ HA AVUTO LUOGO O NUETO)

↳ NON SI PUÒ ESSERE ANONIMO "DA SOLI".

- QUALI SONO I POSSIBILI ATTACCHI ALL'ANONIMATO?

► ANALISI PASSIVA DEL TRAFFICO → INFILZARE DAL TRAFFICO

► ANALISI ATTIVA DEL TRAFFICO → INJECT PACKET (USARE CHI STA PARLANDO CON CHI)

► COMPROMISSONE DEI NODI DELLA RETE (ROUTERS)

↳ NON È BANALE SCOPRIRE QUALI SONO COMPROMESSI

↳ MEGLIO NON FIDARSI DEL SINGOLO NODO (IPOTESI: UNA VERTICE PARTE DEI NODI È BUONA).

↳ IDEA: CHAUM'S MIX → PRIMA DELLO SPAM, SE ERALD ANCHE SEMBRANO FICHE

↳ GLI ATTACANTI SANNO TUTTI I MITTENTI E TUTTI I DESTINATARI MA NESSUNO RIESCE A COLLEGARE UN SINGOLO FUSSO.

- SVANTAGGI DELLE MIXNET?

LE MIXNET HANNO ALTA LATENZA → DOVUTA ALLA CRIFTOGRAFIA A CHIAVE PUBBLICA (OK PER EMAIL MA NON PER WEB BROWSING.)

↳ IDEA: FACCIO COME PER IL RESTO,

USO LA CHIAVE DI SESSIONE...

- RENDOMIZZARE IL ROUTING (CAMBIANDO IL CIRCUITO + VOLTE È + DIFFICILE EFFUGGIARE UN ATTACCO)

ONION ROUTING

• IL MITTENTE DECIDE IL PATH (E QUINDI CREA I LAYER DI CRITTATORE)

• ALCUNI NODI SONO CONTROLLATI DA ATTACANTI

↳ CI POLLA

TOR
↳ ONION ROUTING DI 2ndo GEN.
↳ DISSEGNATO PER LA BASSA LATENZA (WEB BROWSING).
TOR Router TANTI NODI, TANTI UTENTI
(INCORAGGIA PERSONA ALL'USO)

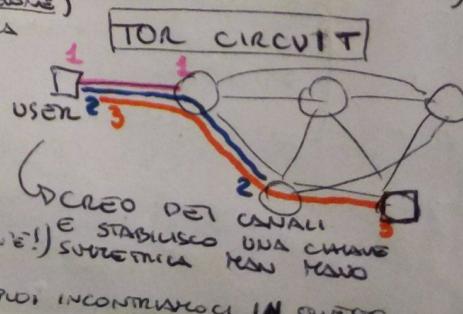
↳ BISOGNA TENERE NASTO LA COORDINAZIONE DEI SERVER

• GOAL: SERVER A CUI TUTTI POSSONO CONNETTERSI, SENZA SAPERE CHI È

• RESISTENZA ALLA CENSURA, AD ATTACCHI DOS, PHYSICAL ATTACK (NON SI SA DOVE È)

• INTRO ed EXIT POINTS

• PUNTO DI RANDOMEY (DEL TIPO: HO BISOGNO DI TE SERVER, SE PUOI INCONTRARMI CI IN QUESTO POSTO)



INTRUDERS (INTRUSI)

REC-52

(MASCHERATO)

(SIA HACKER CHE CRACKER)

- MASQUERADE: UN INDIVIDUO CHE NON È AUTORIZZATO AD USARE IL COMPUTER (OUTSIDER)

- MISFEASOR: UN UTENTE LEGITTIMO CHE HA ACCESSO NON AUTORIZZATO AI DATI O USA IN MANIERA MALEVOLA I SUOI PRIVILEGI

- CLANDESTINE USER: INDIVIDUO CHE PRENDE IL CONTROLLO DEL SISTEMA E UTILIZZA IL CONTROLLO PER EVADERE L'AUDITING

SOLITA DIFFERENZA ~~TRA~~ TRA CRACKERS E HACKERS

TECNICHE DI INTRUSIONE

- USARE ~~VULNERABILITÀ~~ VULNERABILITÀ DEL SISTEMA O DEL SOFTWARE
- DOPO AVER OTTENUTO L'ACCESSO AL SISTEMA, STRALCIARE I PRIVILEGI

LD INDOVINARE LA PASSWORD SPESO È LA TECNICA MIGLIORE
(DIFENDERSI DA QUESTO ATTACCO È COMPIUTO DELL'UTENTE)

LD CATTURARE LA PASSWORD

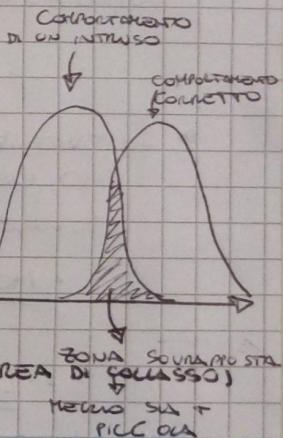
- GUARDARE QUANDO LA PASSWORD VIENE DIGITATA
- USARE UN TROJAN PER CHIEDERE ALL'UTENTE
- MONITORARE UNA RETE NON SICURA

INTRUSION DETECTION

- INEVITABILMENTE AVREMO SEMPRE DELLE FALSI
- SE ABBIANO DELLE INTRUSIONI BISOGNA
 - IDENTIFICARLE VELOCEMENTE
 - COLLEZIONARE INFO PER MIGLIORARE LA SICUREZZA FUTURA

① APPROCCI ALL'INTRUSION DETECTION:

STATISTICALLY ANOMALY DETECTION ①
RULE-BASED DETECTION ②



► STATISTICAL ANOMALY DETECTION:

- CERCA DI DEFINIRE IL "COMPORTAMENTO NORMALE"

A ► THRESHOLD DETECTION

- CONTA LE OCCORRENZE DI UN EVENTO NEL TEMPO
- SE VIENE SUPERATO IL LIMITE IN MODO RAZIONEVALE → ^{CONSIDERATO} UN ^{ATTACCO}
- DA SOLO È UN METODO GREZZO E POCO EFFICACE

B ► PROFILE BASED

- CARATTERIZZA IL COMPORTAMENTO PASSATO DEGLI USATORI
- RILEVA VARIAZIONI SIGNIFICATIVE DA QUEST'ULTIMO
- PROFILO MULTIPARAMETRICO

AUDIT RECORD

- FONDAMENTALI X I.D.
- FILE IN CUI SALVO TUTTO QUELLO CHE AVviENE NEI SISTEMI E CHI LO COMPIE

LD NATIVI

- GLI APPENDISI, PIOMBO, ACCIAIO, AD ESEMPIO IN UN S.O.

LD DETECTION-SPECIFIC

- COLLEZIONARE INFO SPECIFICHE
- HANNO UN PESO AGG. SUL SISTEMA

LD IN QUESTA TIPOLOGIA È OUVIA L'IMPORTANZA
DELL'ANALISI DEI FILE DI AUDIT

LD CONTROLLO LE METRICHE NEL TEMPO

② RULE-BASED DETECTION

- CERCA DI CAPIRE QUALI SONO I PATTERN DI ATTACCO
- APPLICO DELLE REGOLE CHE DECIDONO SE IL COMPORTAMENTO E' O MENO MALEVOLO
- DI DUE TIPI:

(A) RULE-BASED ANOMALY DETECTION

1. ANALIZZA LA STORIA PER IDENTIFICARE UN MODELLO DI COMPORTAMENTO AUTOGESTITO DA QUESTO SET DI REGOLE
2. OSSERVA IL COMPORTAMENTO ATTUALE X VEDERE SE CONFORME
NOTA! COME IL RILEVAMENTO STATISTICO NON PREvede CONOSCENZA PRELIMINARE DEI DIFETTI DI SICUREZZA (FALSI)

(B) RULE-BASED PENETRATION IDENTIFICATION

- UTILIZZA TECNOLOGIE DIAT FORNITE (CREATE DA ESPERTI)
- CON REGOLE CHE IDENTIFICANO PENETRAZIONI NOME E COMPORTAMENTI SOSPETTI
- SPECIFICI PER MACCHINA/S.O.
- LE REGOLE VENGONO CHI CONFRONTATE CON GLI AUDIT ATTUALI!

UN BUON SISTEMA DI
INTRUSION DETECTION

DEVE RILEVARE
QUASI TUTTI/TUTTI
LE INTRUSIONI
REALI

MENO FAISI
ALLARMI
POSSIBILE

► SOLITAMENTE DOBBIAMO FARLE INTRUSION DETECTION SU **SISTEMI DISTRIBUITI**
(AD ESEMPIO RETE LOCALE CON TUTTI I PC ALL'INTERNO)
DUNQUE AVRO:

- HOST AGENT MODULE = INSTALLATO SUI SINGOLI PC CHE RACCOLGONO AUDIT RECORD
- LAN MONITOR AGENT MODULE = LA STESSA COSE MA ANALIZZA IL TRAFFICO LAN
- CENTRAL MANAGER MODULE = RICEVE REPORT OPPURE DIRETTAMENTE GLI AUDIT / LI FILTRA E CORTE ANOMALIE, SEGNI DI INTRUSIONE IN CORSO.

HONEY POT

→ ALTAMENTE INVITANTI X ATTACCANTE
→ ATTUALMENTE VANNO DI MODA NEGLI IoT
↳ DEVONO COMUNQUE ESSERE BEN DIFESI E SEMBRIARE VERI (L'HACKER DEVE PENSARE CHE STA COMBATTEndo CON IL SISTEMA)

↳ DIVERSI POSTI NELLA DMZ → FUORI

DENTRO DMZ
RETE LOCALE

I SISTEMISTI SANNO CHE SOLO L'ATTACCANTE CI METTE LE MANI

USO L'HONEY POT COME LEARNING (VEDO DOWNE METTE LE MANI L'ATTACCANTE E COSA FA)

APPRENDIMENTO DI PASSWORD → INSTRUIRE USRNT

SWAPAGE HASH SUL SERVER E' RECISO

FIREWALL

- SISTEMI ← PREVENZIONE (IPS) → FIREWALL
INDIVIDUAZIONE ATTACCO (IDS)

- UN FIREWALL È UN PUNTO DI CONTROLLO E MONITORAGGIO DEL TRAFFICO
- UN FIREWALL È PIÙ VENTO TANTO PIÙ SI TROVA IN ALTO RISPETTO ALLA PILA ISO/OSI.
- FA MOLTA PROTEZIONE SUI PACCHETTI CHE ARRIVANO DALL'ESTERNO E POCO (O COMUNQUE SOLITAMENTE MENO)
- NON PROTEGGE DA MALWARE OVVIAMENTE (E DA QUALSIASI COSA SCALVAGHI IL FIREWALL)

○ FIREWALL - 1° TIPO - PACKET FILTERING

- I PIÙ SEMPLICI E I PIÙ VELOCI
 - GUARDA IP SORGENTE, IP DESTINATARIO E PORTA USCITA / DESTINAZIONE A VOLTE SI GUARDANO I FLAG AD ESEMPIO IL FLAG DI FRAGMENTAZIONE (SE NON VOGLIO PERMETTERE CHE I PACCHETTI CHE SONO FRAGMENTATI PASSINO)
 - POSSIBILI POLICY DI DEFAULT
 - SE NON È ESPRESSAMENTE CONSENTITO, TUTTO IL RESTO È PROIBITO
 - " " PROIBITO , " " CONSENTITO
 - QUESTI FIREWALL SONO MESSI SULLE INTERFAZIE D'INGRESSO E DI USCITA DEL GATEWAY.
 - ATTACCHI POSSIBILI:
 - IP ADDRESS SPOOFING: CERCA DI METTERE UN IP INTESTATO AL PACCHETTO CHE SPEDISCE DA L'ESTERNO (ORMAI NON FUNZIONA PIÙ PERCHE' I FIREWALL DISTINGUONO TRA INTERFAZIE INTERNE DA ESTERNE)
- ...

○ FIREWALL - STATEFUL PACKET FILTERS

- LA TIPOLOGIA PRECEDENTE NON ESAMINA I LAYER PIÙ ALTI (NEMMENO QUESTO TIPO MA FA UN MINIMO DI RAZIONAMENTO)
- CON QUESTO FIREWALL POSSO DIRE: OK VA BENE UN PACCHETTO VERSO QUESTO IP MA POI I SOCKET NON POSSONO ESSERE APERTI SE NON SONO SU QUESTA PORTA.

○ FIREWALL - APPLICATION LEVEL GATEWAY (OR PROXY)

- IL PROXY PUÒ CONTROLLARE SE L'UTENTE È AUTENTICATO A FAR E UNA DETERMINATA RICHIESTA.

- TIPICAMENTE DEVONO ESSERE IL PIÙ PICCOLI POSSIBILE

↳ LA TENTAZIONE È SCRIVERE UN UNICO PROXY CHE GESTISCE SO SERVIZI: QUESTO È SBACCATO POICHÉ:

INVECE DI SO PROXY PER SO SERVIZI

- SE VIENE BUCATO IL PROXY ESPONGO TUTTI I SERVIZI
- QUANDO VI È UN PROBLEMA (es. UN INTRUSION DETECTION MI HA INDIVIDUATO UN PROBLEMA) DEVO BUTTARE GIÙ TUTTI I SERVIZI SE BUTTO GIÙ IL PROXY.
- L'ANALISI DEL PROBLEMA SUL PROXY È MOLTO PIÙ DIFFICILE SE HA 5000 RICHESENZE DI 500 (E DUNQUE PIÙ DIFFICILE INDIVIDUARE IL BACO).

○ FIREWALL - CIRCUIT LEVEL GATEWAY (SI MOVA A LIVELLO TCP)

↳ POSSO SPECIFICARE L'ANALISI DEL TRAFFICO PER UN DETERMINATO "CIRCUITO"
(VEDI SLIDE)