IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Security protocols: an overview

Gabriele Costa
System Modelling and Analysis (SysMA)
gabriele.costa@imtlucca.it

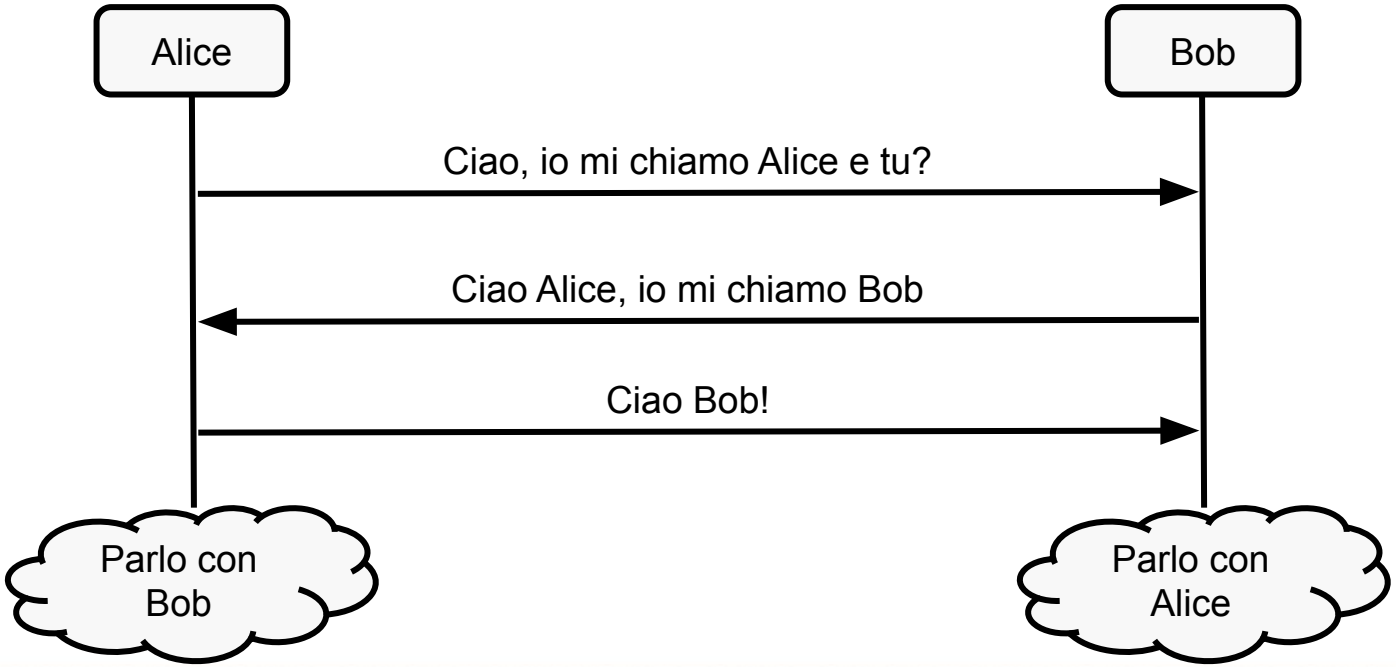
IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Agenda

- Security protocols
- Attacker models
- Security goals
- Needham-Schroeder: un classico
- Multi-factor (strong?) authentication

Protocollo

- Algoritmo distribuito, eseguito da più **agenti** che comunicano tramite **messaggi** inviati su **canali** di comunicazione



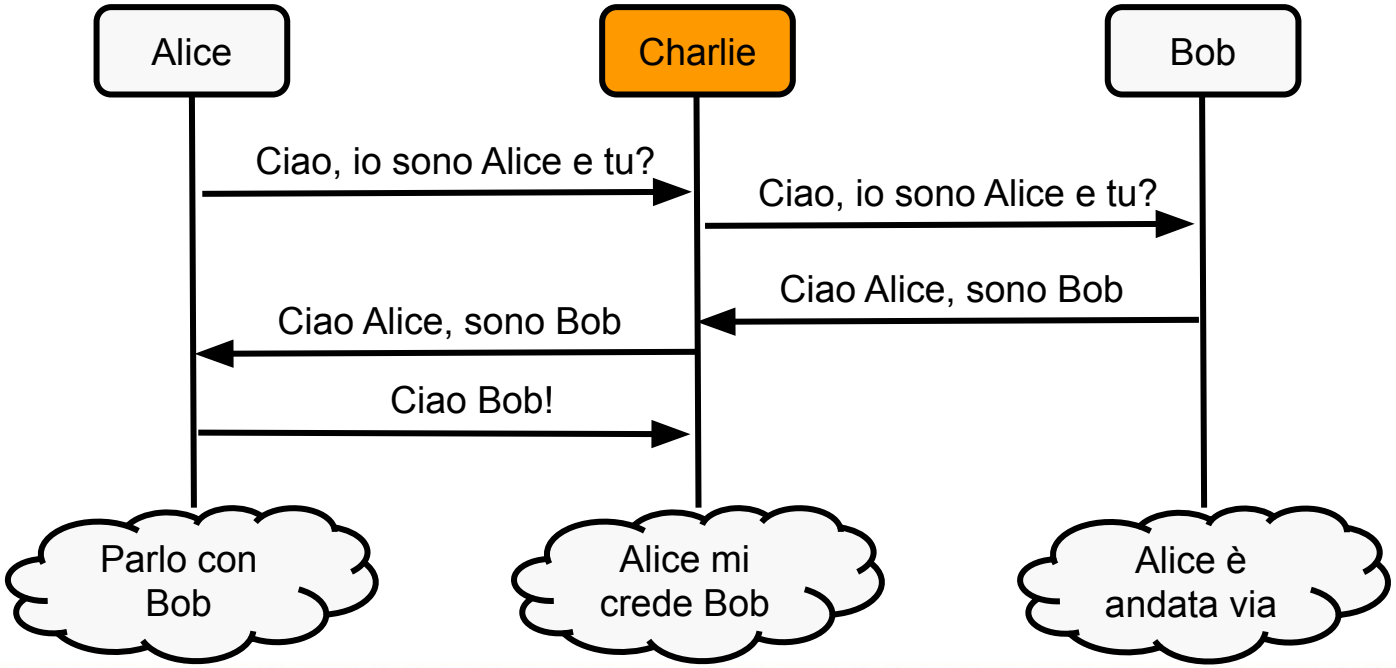
IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Security protocol

- Protocollo, con **agenti potenzialmente disonesti** e **canali non affidabili**

Security protocol

- Protocollo, con **agenti potenzialmente disonesti** e **canali non affidabili**



Notazione Alice-Bob

A invia un messaggio m a B

A -> B: m

Messaggi

- Cleartext: “hi”
- Sequenze: m1, m2
- Hash: h(m)
- Nonce: N
- Crittogrammi: {m}K
- Firme: [m]K

Notazione Alice-Bob: convenzioni

N_A: nonce generato da A

$K_{A,B}$: chiave simmetrica condivisa tra A e B

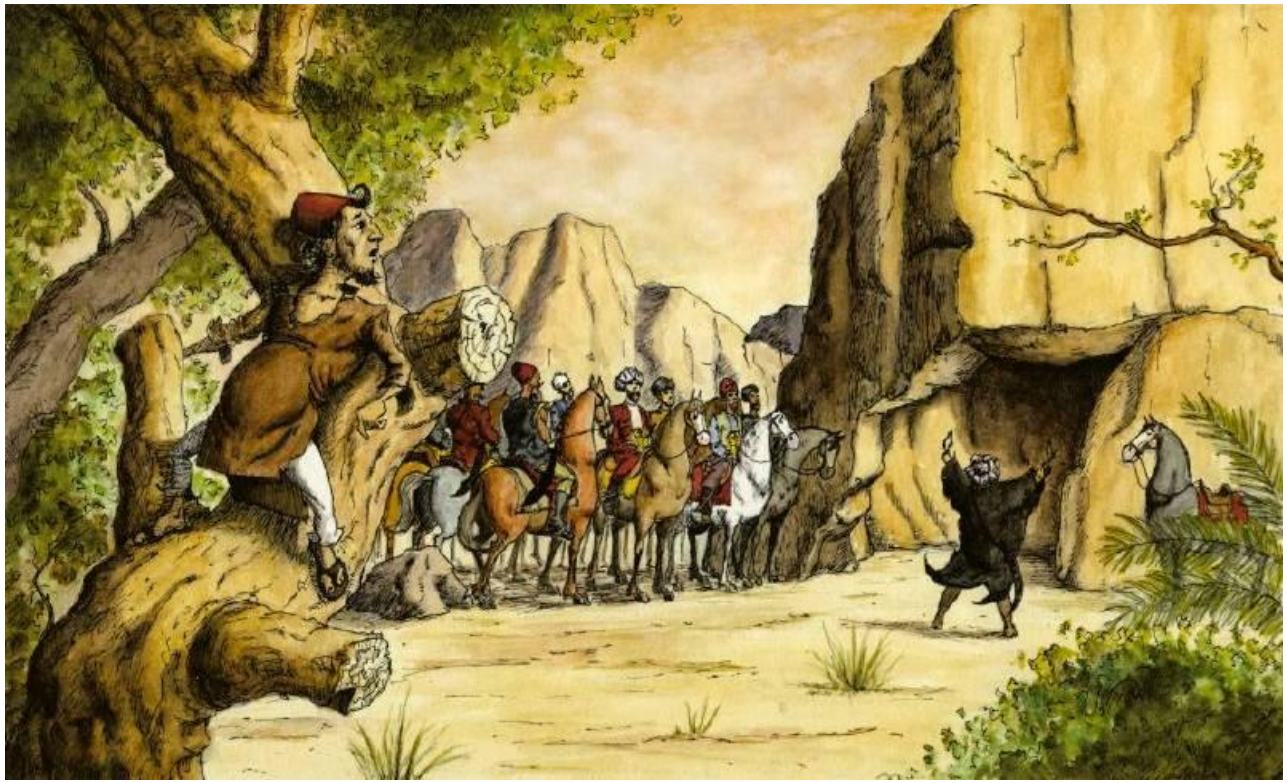
K_A: chiave pubblica di A

K^{-1}_A: chiave privata (inversa) di A

Assunzione: tutti gli agenti conoscono le proprie chiavi private e le chiavi pubbliche di tutti

Notazione Alice-Bob: esempio

L -> C: “apriti sesamo”



Attacker model (Dolev-Yao)

Assunzioni comuni sull'attaccante

- Non può infrangere la crittografia
  - Perfect crypto
- Non può invertire le trap functions
  - Perfect hashing
- Non può indovinare le nonce
  - Perfect RNG, no brute force
- Controlla i canali
  - Legge, scrive e blocca i messaggi
- Conosce le primitive crittografiche
  - Decifra/firma se conosce le chiavi

Security Goals

"Un software senza specifiche non può essere difettoso, può solo essere sorprendente"

- Senza goal un security protocol non si può dire "attaccabile"...
  - … ma neanche chiamare security protocol!
- Goal tipici
  - Authenticity (il messaggio m arriva da X)
  - Confidentiality (solo X e Y conoscono m)
  - Integrity (m non è stato alterato)

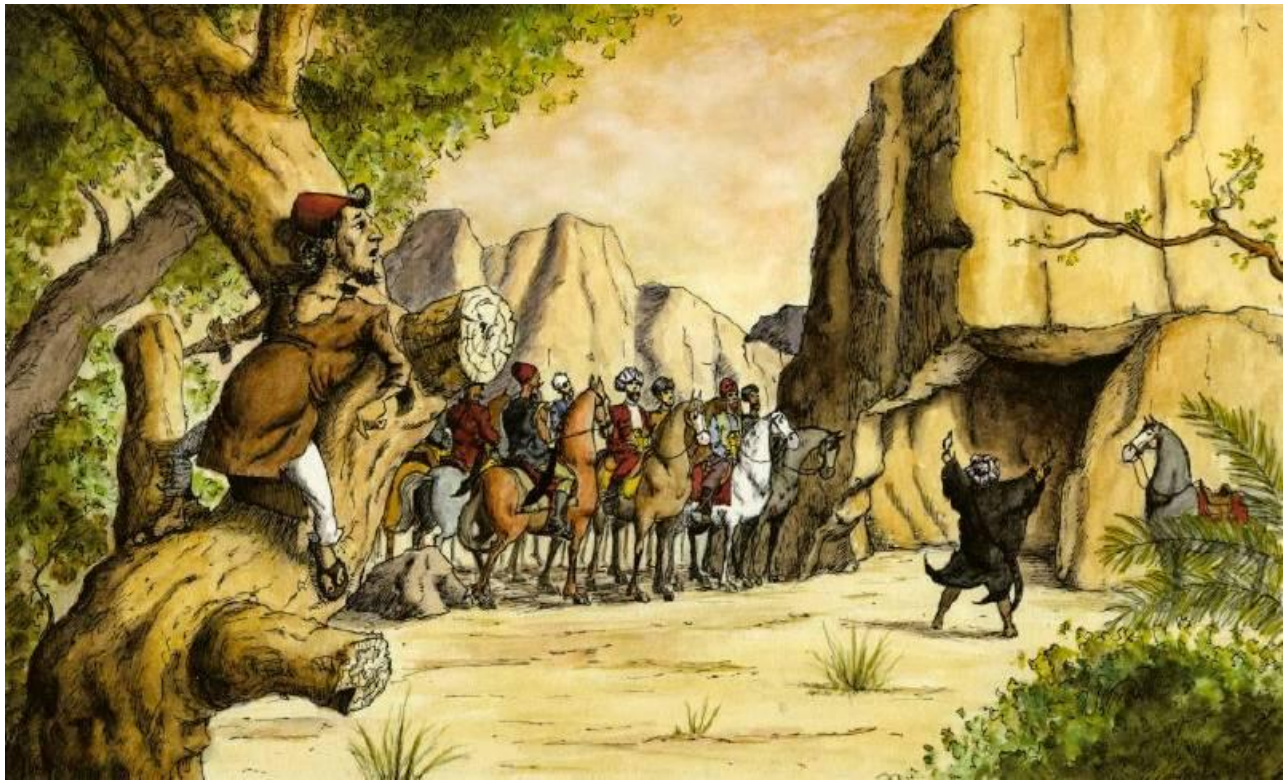
Tipi di attacco

- Replay attack: riusare (parte di) vecchi messaggi
- Man-in-the-middle: controllare interamente il canale tra due agenti
- Reflection attack: restituire messaggi al mittente
- ...

Reply attack: esempio

L -> C: “apriti sesamo”

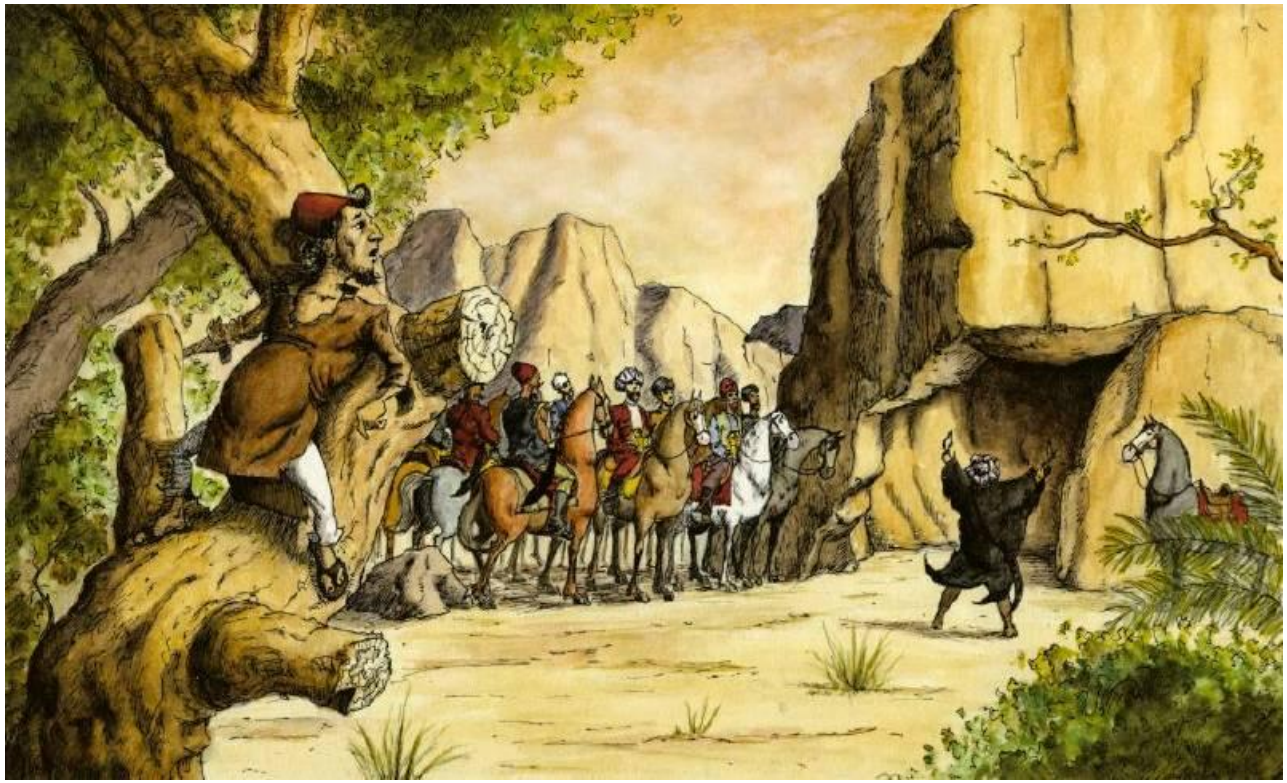
A -> C: “apriti sesamo”



Reply attack: esempio 2

L -> C: {“apriti sesamo”} $K_{L,C}$

A -> C: {“apriti sesamo”} $K_{L,C}$



Protocollo NSPK

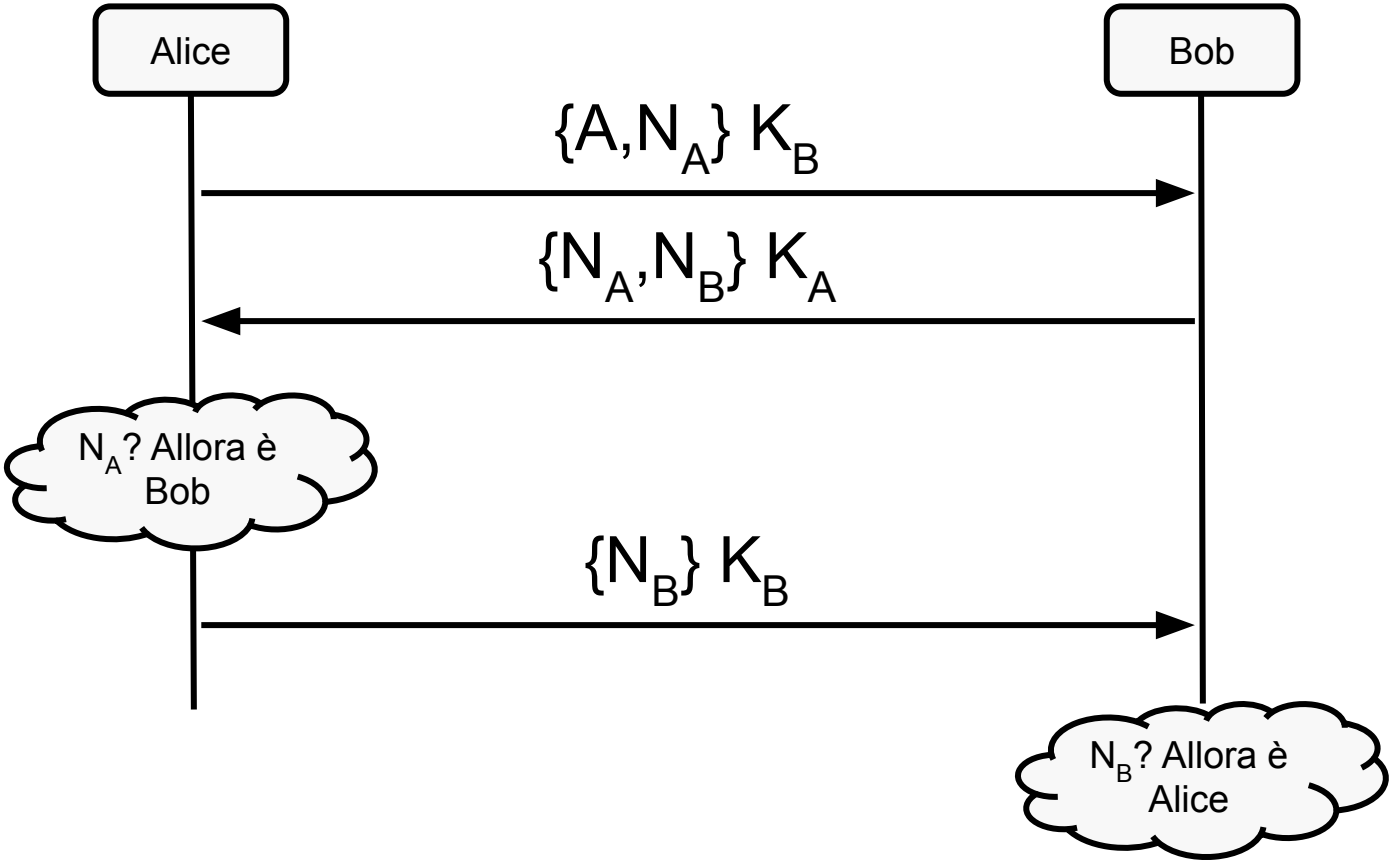
- Proposto da Needham e Schroeder nel 1978
- Revisionato nel 1987

1. A -> B: $\{A,N_A\}\ K_B$
2. B -> A: $\{N_A,N_B\}\ K_A$
3. A -> B: $\{N_B\}\ K_B$

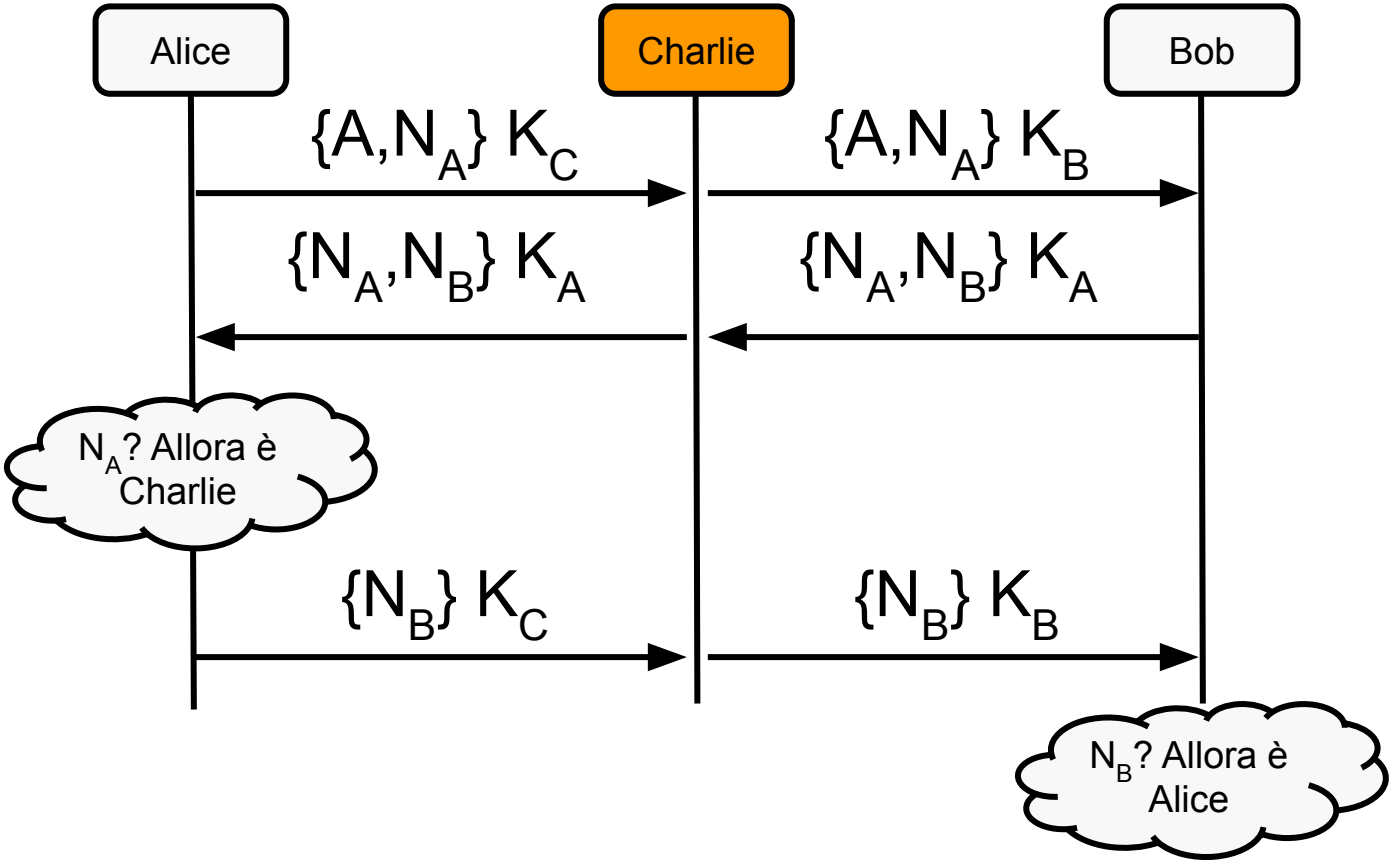
Goal: autenticazione reciproca di A e B

- Basata sulla segretezza condivisa di N_A e N_B

Protocollo NSPK: attacco (G. Lowe '95)

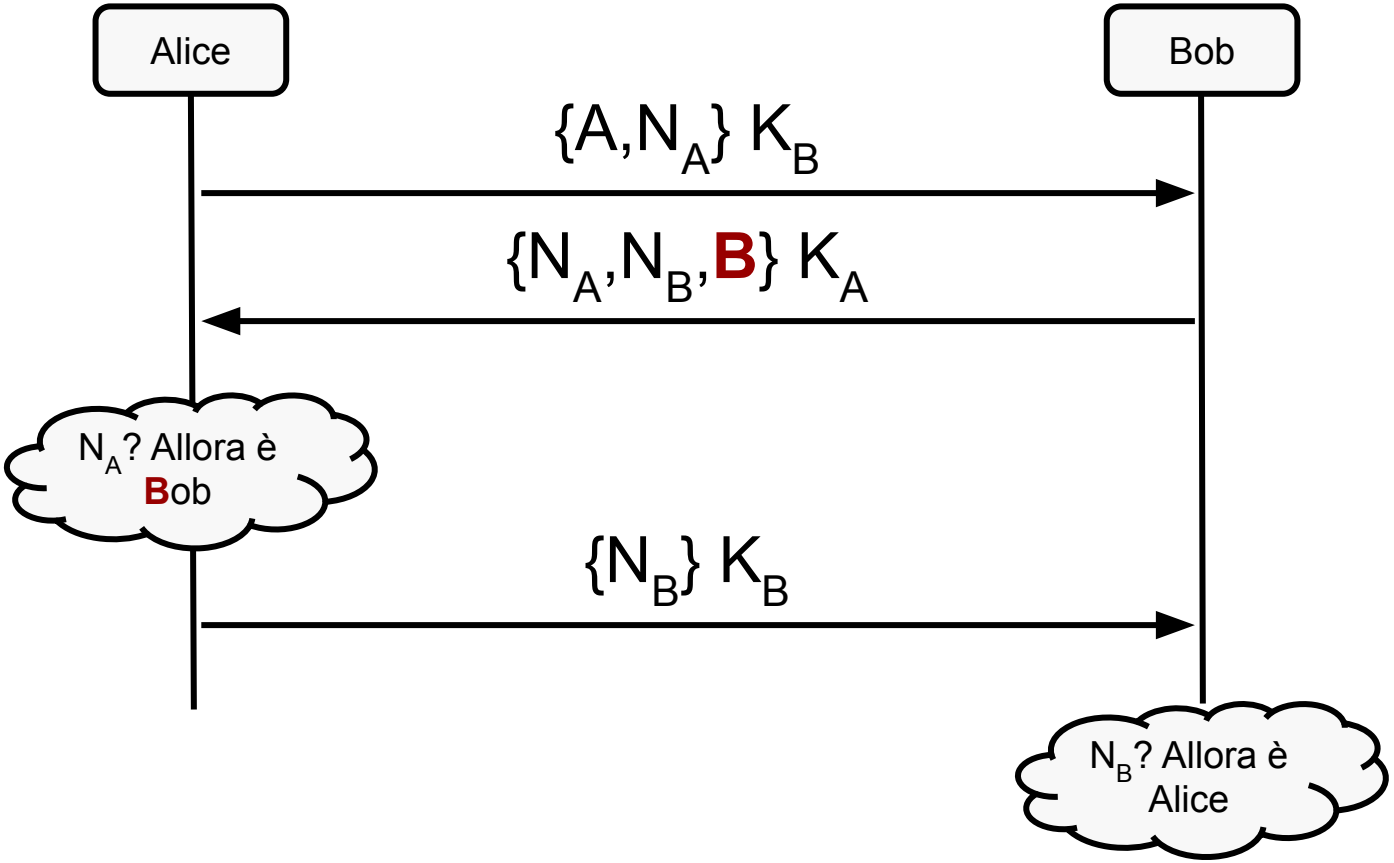


Protocollo NSPK: attacco (G. Lowe '95)



IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Protocollo NSPK: Lowe's fix



Considerazioni sull'attaccante DY

- Ha il controllo totale su tutti i canali
  - Non sempre realistico
  - Attaccare un canale ha un **costo** (economico o in termini di rischio)
- Non può compromettere gli endpoint
  - Browser, PC, smartphones
  - Ma questo può capitare (e.g. malware)

Multi-factor authentication

IDEA: aumentare il costo (oltre i possibili benefici per l'attaccante) incrementando i fattori che è necessario compromettere per portare a compimento l'attacco

Contesto tipico: **e-banking** (un utente deve essere autenticato dalla propria banca)

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Multi-factor authentication: definizioni

- Authentication Factor: qualcosa che identifica chi ne detiene il controllo
- Tre tipi di AF
  - Knowledge (K). Pin o password
  - Ownership (O). Chiave o smartcard
  - Inherence (I). Retina o impronte digitali

Multi-factor authentication: definizioni

- Authenticator: dispositivo usato per attestare il controllo su uno o più AF

Multi-factor authentication: definizioni

- Evidence: informazione da esibire per dimostrare il controllo su un AF. Viene generato da un authenticator.

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

AF fantastici e dove trovarli

Authentication Factor	Authenticator	Evidence
Memorized secret (K)	Software/Hardware keyboard	Hash
Authentication Matrix (K/O)	Software/Hardware keyboard	Content of a cell
Credit card (K/O)	Software/Hardware keyboard	ID + ccv
Smartcard (O)	Card reader	Digital signature
Phone number (O)	SIM	SMS/Call
OTP (O)	OTP Generator	Next OTP
Fingerprint (I)	Fingerprint reader	Unlock software
Face (I)	Phone camera	Unlock software
...	...	...

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Canali di comunicazione

- Protocolli di rete (es. https)
- Lettori ottici (es. fotocamera del cellulare)
- Rete telefonica mobile
- Utente umano
- IPC (segnali tra applicazioni)

Multi-factor authentication: esempio

- Autenticazione tramite card reader

Alice -> Bank: {**username**,h(**password**)} K_B

Bank -> Alice: {N_B} K_A

Alice -> CReader: N_B,**pin**

CReader -> Alice: [N_B] $\mathbf{K_{B,C}}$

Alice -> Bank: [N_B] $\mathbf{K_{B,C}}$

Multi-factor authentication: esempio

Secure element

- Alcuni authenticator sono progettati per essere resistenti contro le manomissioni
  - Per esempio, SIM, Smartcard, OTP generators
- Praticamente mai questo avviene se l'authenticator è un software
  - La piattaforma di esecuzione è untrusted

Enrollment, Binding, Exemptions

- Anche le fasi preliminari all'esecuzione di un protocollo devono essere tenute in considerazione
  - Enrollment: come l'utente viene abilitato all'uso del protocollo (prima identificazione)
  - Binding: come gli AF e gli authenticator vengono attivati e consegnati all'utente
  - Exemptions: come e quando l'utente può evitare di eseguire MFA

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Attacker model revised

- **Device Thief**: può rubare oggetti fisici al proprietario
- **Authenticator Duplicator**: può duplicare un authenticator (tranne secure elements)
- **Shoulder Surfer**: può spiare l'attività dell'utente
- **Eavesdropping software**: può spiare l'immissione di dati da parte dell'utente (es. keylogger)

Attacker model revised

- **Social Engineer**: può indurre un utente male informato a usare un authenticator in modo scorretto
- **Man in the Browser**: ha il controllo del browser dell'utente (se l'endpoint è un PC)
- **Man in the Mobile**: ha il controllo dello smartphone dell'utente

Attaccanti più forti si ottengono per composizione

Composizione di attaccanti: esempio

SS compromette i fattori di **conoscenza**

Alice -> Bank: {**username**,h(**password**)} K_B

Bank -> Alice: {N_B} K_A

Alice -> CReader: N_B,**pin**

CReader -> Alice: [N_B] $K_{B,C}$

Alice -> Bank: [N_B] $K_{B,C}$

Composizione di attaccanti: esempio

SS compromette i fattori di **conoscenza**

Alice -> Bank: {**username**,h(**password**)} K_B

Bank -> Alice: {N_B} K_A

Alice -> CReader: N_B,**pin**

CReader -> Alice: [N_B] $\mathbf{K_{B,C}}$

Alice -> Bank: [N_B] $\mathbf{K_{B,C}}$

Ma non può compromettere i fattori di **possesso**

Composizione di attaccanti: esempio

- SS ha bisogno di comporsi con un'altra tipologia di attaccante
  - Maggiori capacità => maggiori costi
- Un buon candidato per la composizione deve poter attaccare i fattori di possesso
  - DT (sottrae CReader ad Alice)
  - SE (inganna Alice che usa male CReader)
  - DD (crea una copia di CReader)
    - Ma CReader è un secure element

Composizione di attaccanti: esempio

SS+**SE** in azione

Charlie -> Bank: {**username**,h(**password**)} K_B

Bank -> Charlie: {N_B} K_A

Charlie -> Alice: {N_B} K_A

Alice -> CReader: N_B,pin

CReader -> Alice: [N_B] $K_{B,C}$

Alice -> Charlie: [N_B] $K_{B,C}$

Charlie -> Bank: [N_B] $K_{B,C}$

Composizione di attaccanti: esempio

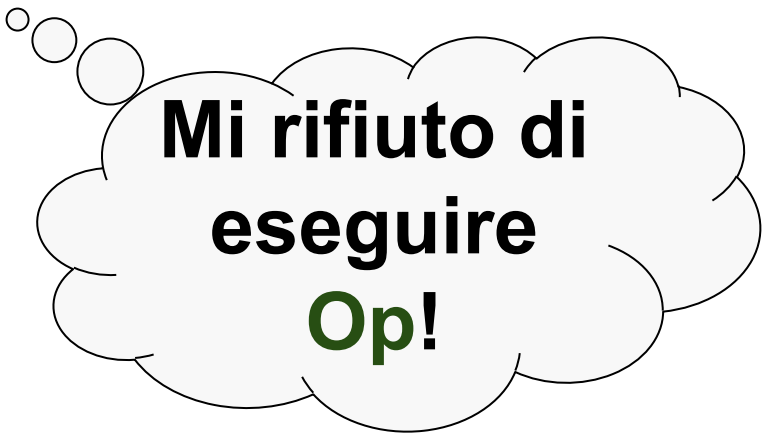
Possibile contromisura: informare Alice

Charlie -> Bank: {username,h(password)} K_B

Bank -> Charlie: {N_B,**Op**} K_A

Charlie -> Alice: {N_B,**Op**} K_A

Alice



Applicazioni recenti

Zero-Knowledge proofs

- “Dimostrare” significa convincere qualcuno della verità di una proposizione
- La dimostrazione è ZK se nel processo non vengono comunicati certi fatti “sensibili”

Blockchain

- Una struttura dati distribuita gestita interamente tramite security protocols

Conclusioni

I protocolli di sicurezza sono intorno a noi

- Centinaia di specifiche
- Migliaia di implementazioni

I goal di sicurezza sono molteplici

- Autenticazione (es. Kerberos)
- Autorizzazione (es. OAUTH)

La crittografia è necessaria ma non sufficiente

- Molti attacchi assumono perfect crypto