Reti di Calcolatori e Sicurezza

4. Crittografia per la Sicurezza

Capp. 6,7,15 Schneier

Capitoli crittografia Stalling

Capitolo sicurezza kurose

Thanks to Giampaolo Bella for slides draft!!

Crittografia

- Scienza antichissima: codificare e decodificare informazione
- Tracce risalenti all'epoca di Sparta
- Seconda guerra mondiale: **ENIGMA**
- Antica: crittografia simmetrica
- Moderna: crittografia asimmetrica (1977)

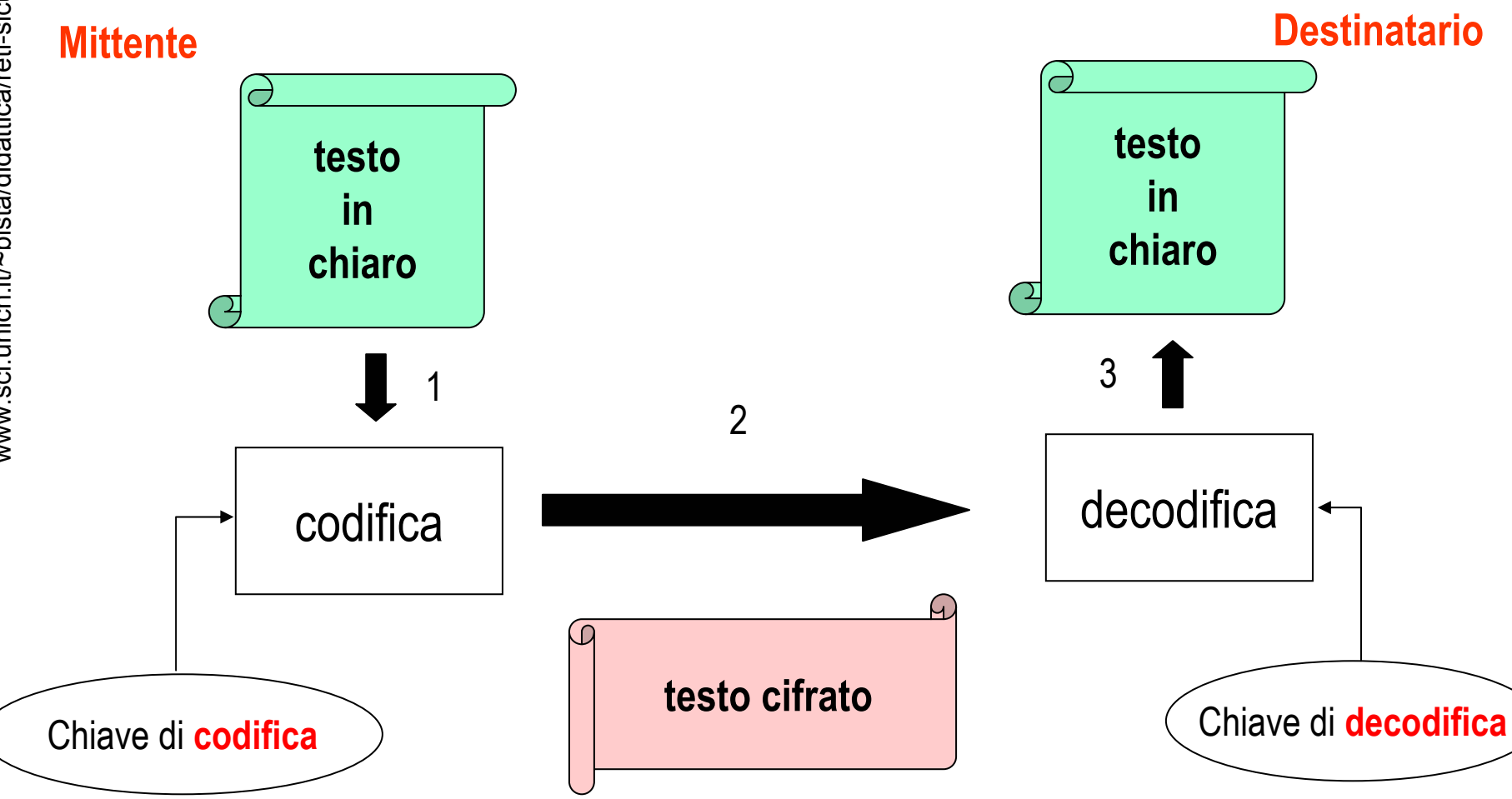
Thanks to Giampaolo Bella for slides draft!!

Crittografia

- Codificare: testo in chiaro → testo codificato
- Decodificare: testo codificato → testo in chiar
- Ambedue basate su: **algoritmo** e **chiave**
  - *Es: "Shiftare" di k una stringa*
- Algoritmo pubblico!
- Sicurezza data da:
  1. segretezza della chiave
  2. robustezza dell'algoritmo

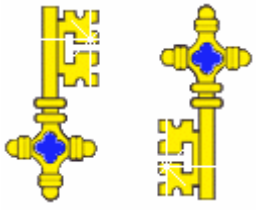
Thanks to Giampaolo Bella for slides draft!!

Codifica e decodifica



Thanks to Giampaolo Bella for slides draft!!

Crittografia simmetrica



- Medesima chiave per codifica e decodifica
- Segretezza, autenticazione, integrità dalla segretezza della chiave

\+ Di solito (*DES*) usano chiavi di 64-128 bit (17-34 cifre decimali) e sono molto veloci

\- Distribuire chiave a coppie di utenti

\- Per n utenti servono n^2 chiavi diverse

Thanks to Giampaolo Bella for slides draft!!

Crittografia asimmetrica

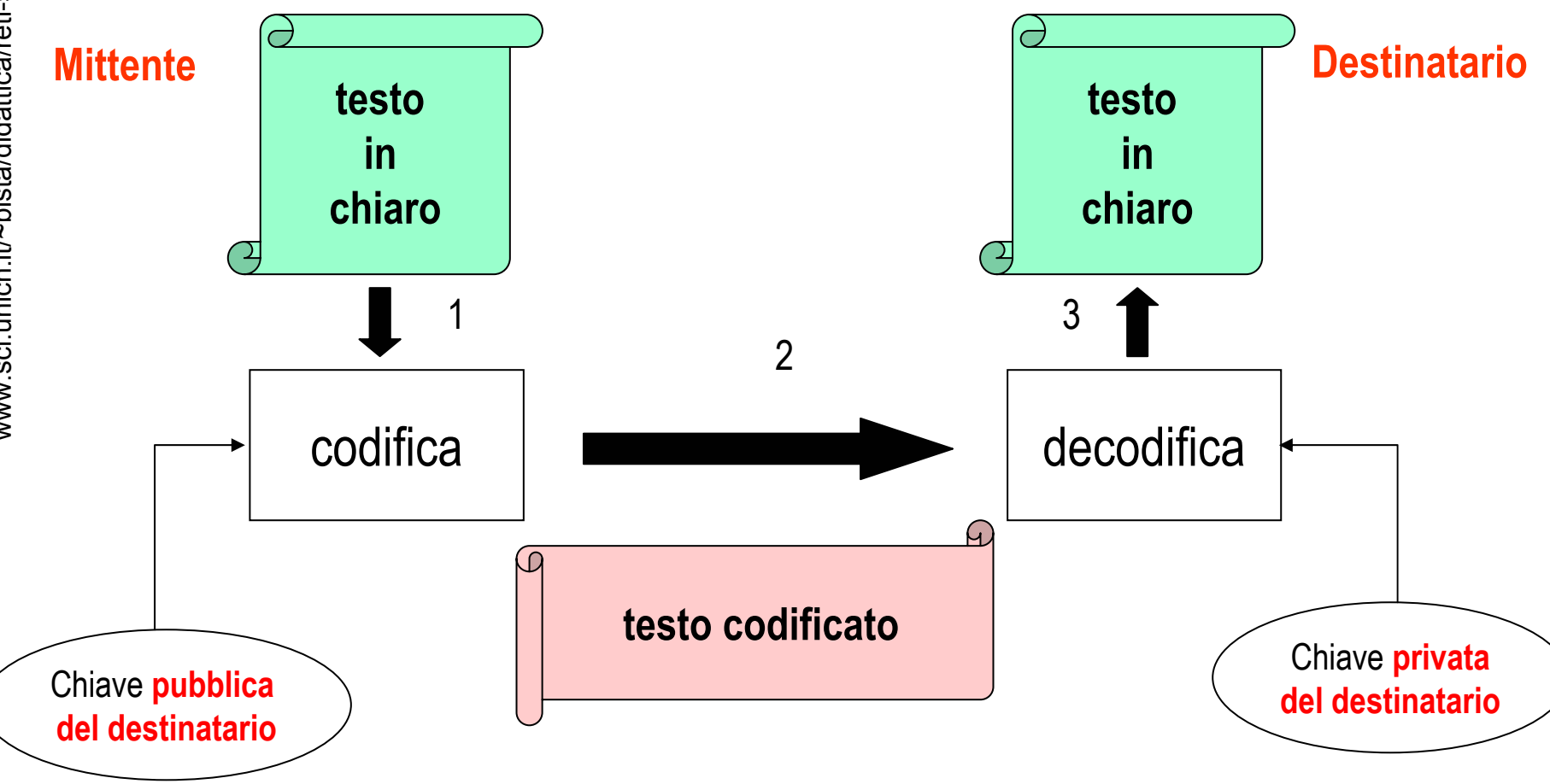
- Una chiave per codifica, un'altra per decodifica
- Ogni utente ha una coppia di chiavi
  - **chiave privata**: segreto da custodire
  - **chiave pubblica**: informazione da diffondere
- Entrambe usabili per codificare o decodificare

- Di solito (*RSA*) usano chiavi di 1024-2048 bit (circa 160-320 cifre decimali) e sono lenti

+ Segretezza, autenticazione, integrità…

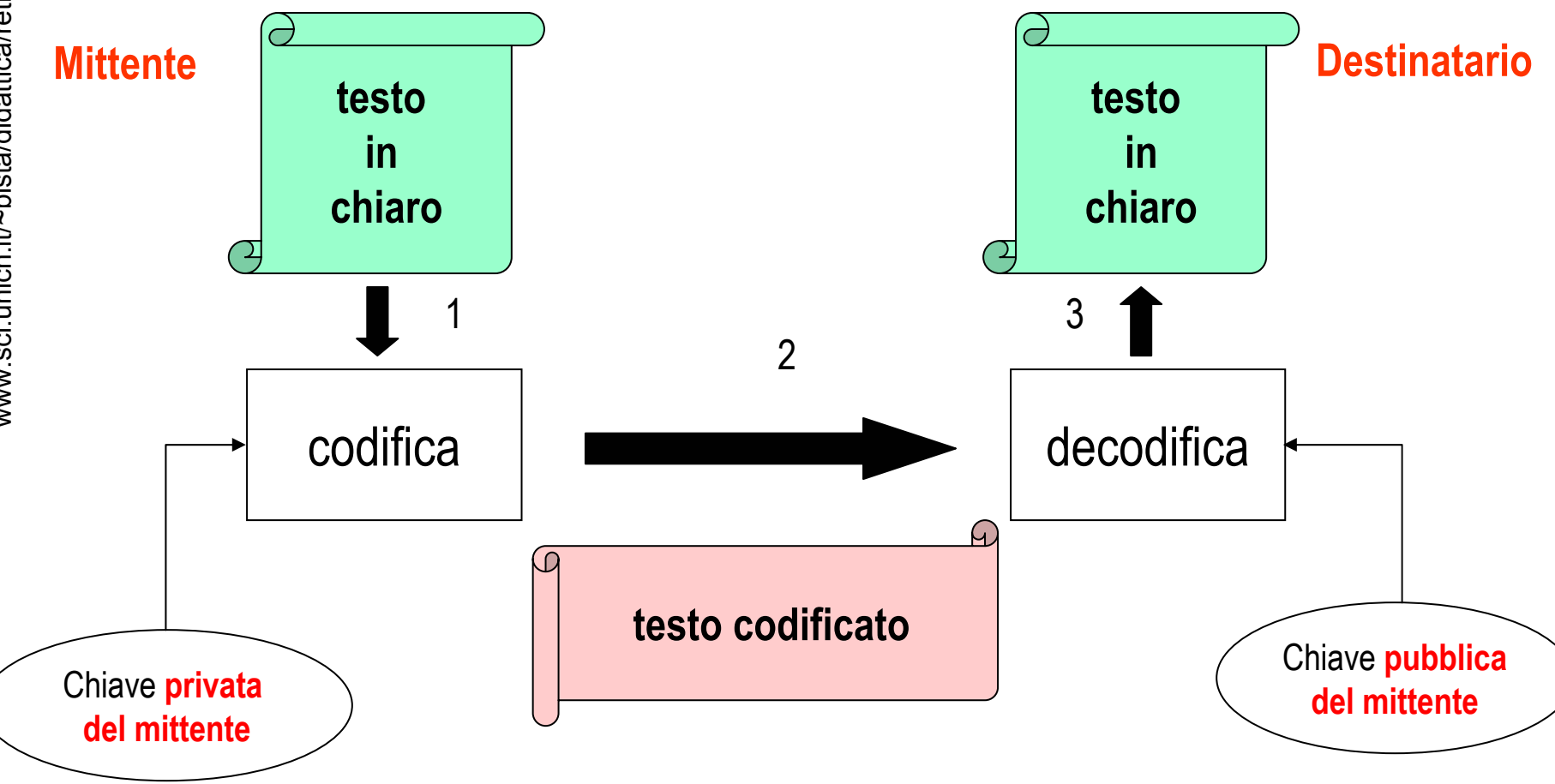
Thanks to Giampaolo Bella for slides draft!!

Segretezza



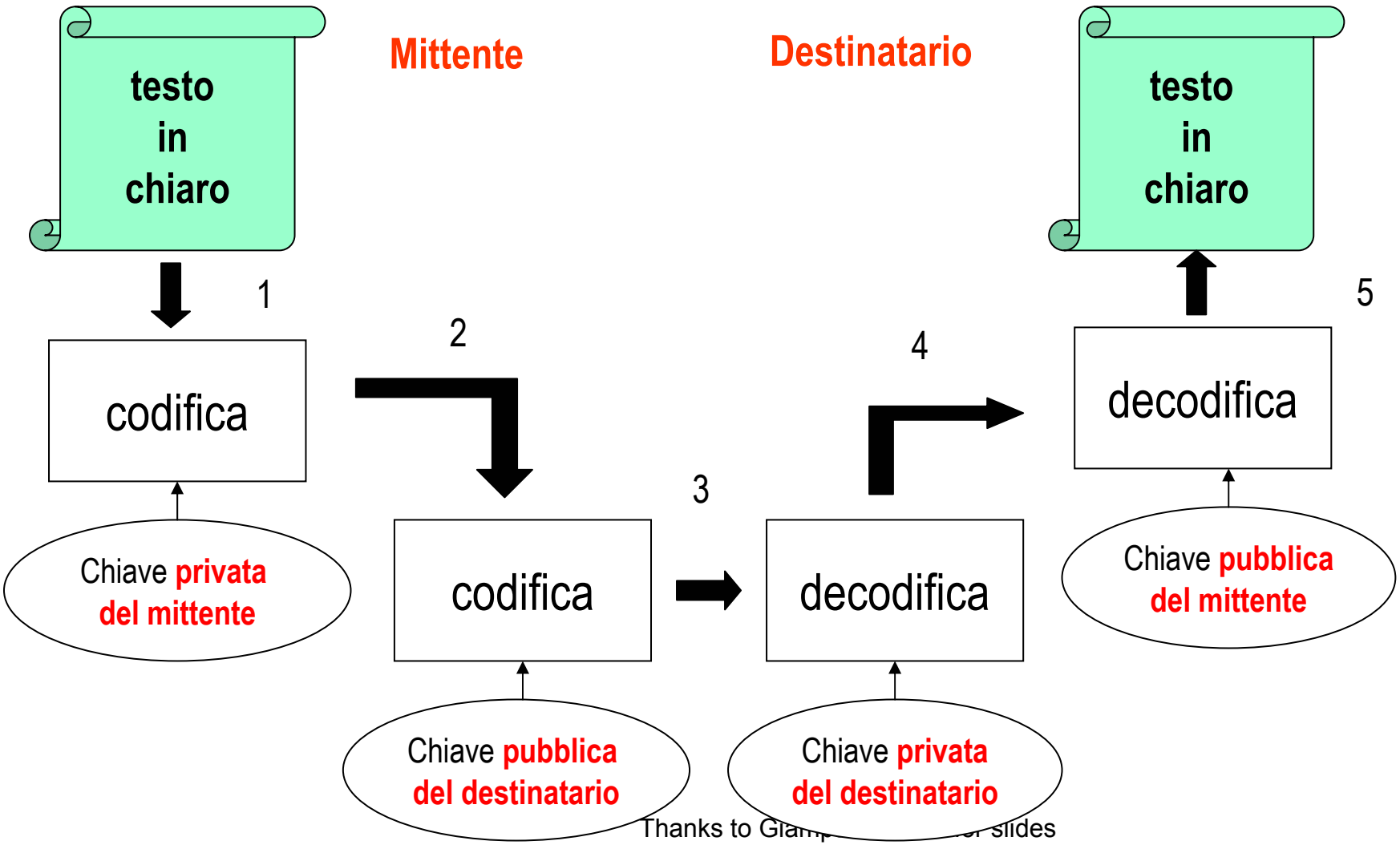
Thanks to Giampaolo Bella for slides draft!!

Autenticazione e integrità



Thanks to Giampaolo Bella for slides draft!!

Le tre insieme



Thanks to Gian... ...r slides draft!!

Attacchi crittografici (crittoanalisi)

1. **Cyphertext only**: noto solo il testo codificato
2. **Known plaintext**: testo in chiaro noto
3. **Chosen plaintext**: testo in chiaro scelto
4. **Brute-force**: attacco alla chiave

Thanks to Giampaolo Bella for slides draft!!

Crittografia perfetta

Def. Nessun testo codificato rilascia informazione alcuna né sulla chiave usata per la codifica, né sul testo in chiaro, il quale può essere recuperato se e solo se la chiave è disponibile

- Ideale: nessun tipo di crittoanalisi possibile
- Probabilità nulla di ricavare informazioni supplementari da un testo codificato
- Crittografia in pratica quasi mai perfetta

Thanks to Giampaolo Bella for slides draft!!

Funzioni hash irreversibili (digest)

- h : X → Y è **hash** se
  1. H can be applied to a block of data at any size
  2. H produces a fixed length output
  3. Dato x ∈X è computazionalmente facile (tempo polinomiale nella dim. dell'input) calcolare h(x)
- …è **irreversibile** se
  1. For any given block x, it is computationally infeasible to find x such that H(x) = h
  2. For any given block x, it is computationally infeasible to find y≠x with H(y) = H(x).
  3. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y)
- Integrità di un testo

Thanks to Giampaolo Bella for slides draft!!

Codici di autenticazione dei messaggi (MAC)

- Forma primitiva di crittografia
- Mittente e ricevente condividono una chiave per calcolare il MAC
- Mittente manda x, MAC(x)
- Ricevente prende x e ne ricalcola MAC(x)
- Si può usare una funzione hash come MAC?

Thanks to Giampaolo Bella for slides draft!!

Numeri (pseudo)casuali

- Generati mediante algoritmo (pseudo)deterministico
  - *Sul rumore elettrico prodotto da un diodo*
  - *Movimenti casuali richiesti all'utente*
- Servono ad ottenere **freshness**
  - *Genero x casuale e lo spedisco insieme a…*
  - *Qualunque cosa riceva che citi x è posteriore a…*

Thanks to Giampaolo Bella for slides draft!!

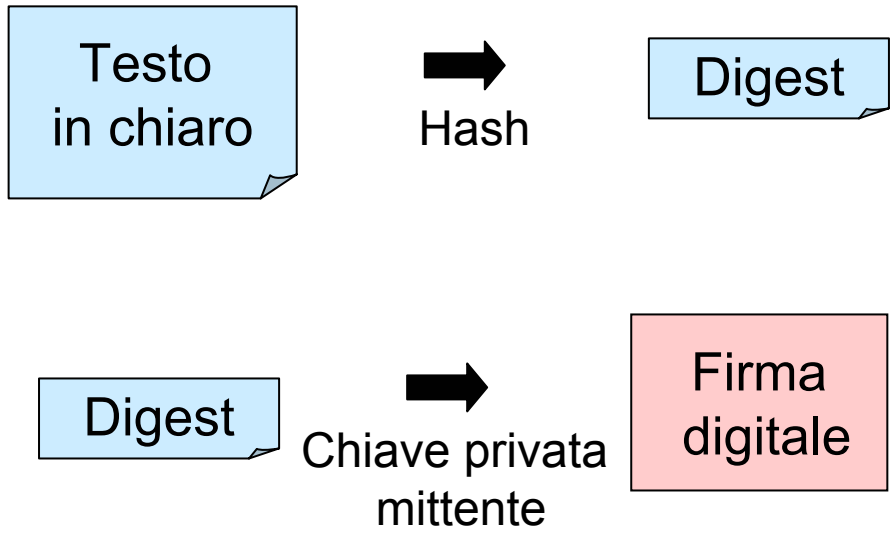
Firma digitale

- Basata su crittografia asimmetrica
- Ottiene solo autenticazione e integrità
- Firmare non è esattamente codificare
- Verificare una firma non è esattamente decodificare

Thanks to Giampaolo Bella for slides draft!!

Creazione della firma

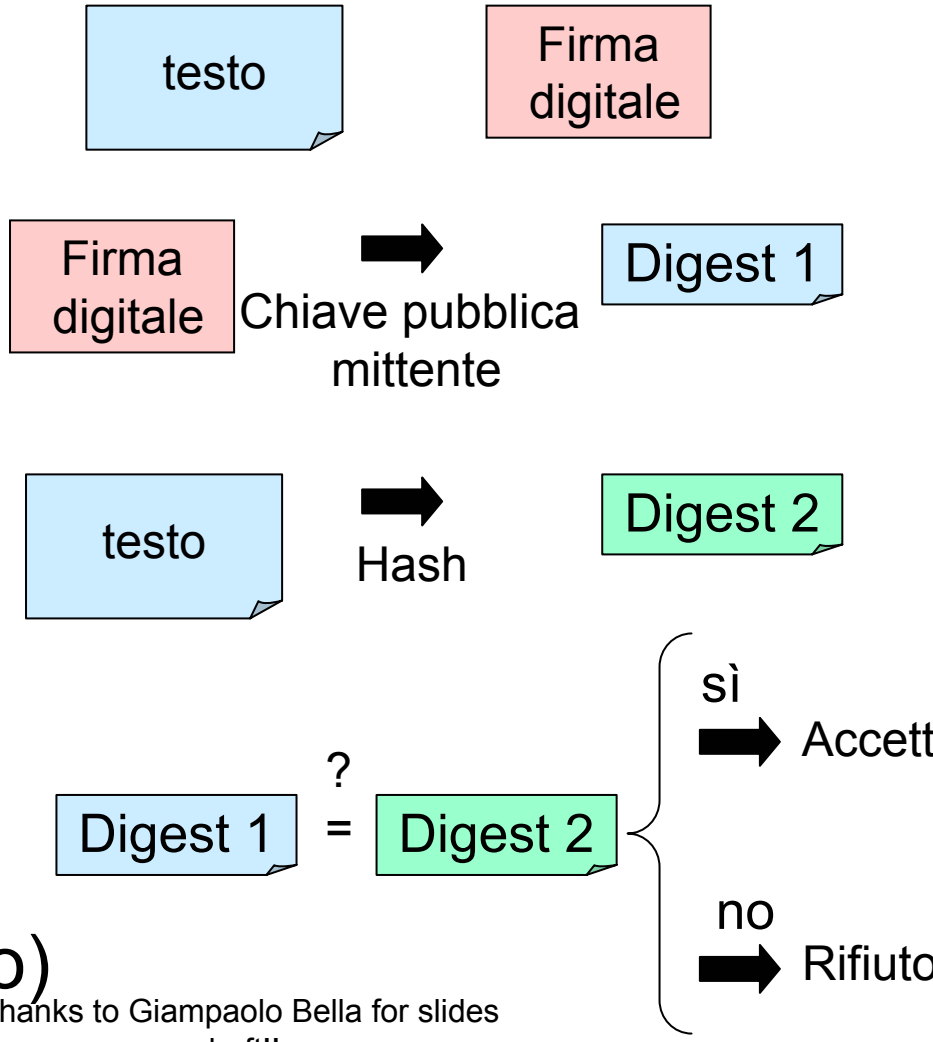
1. Calcolare il **DIGEST** del testo
2. Codificare il digest con la chiave privata del mittente (si ottiene la firma digitale vera e propria)
3. Creare coppia testo+firma e spedirla



Thanks to Giampaolo Bella for slides draft!!

Verifica della firma

1. Separare il testo dalla firma
2. Decodificare la firma con la chiave pubblica del mittente
3. Calcolare il digest del testo
4. Verificare che i due digest coincidano
   - sì: accetto (testo OK)
   - no: rifiuto (testo alterato)



Thanks to Giampaolo Bella for slides draft!!

Garanzie

La firma digitale garantisce che:

- **Autenticità: Il messaggio arrivi proprio da chi dice di essere il mittente**
- **Integrità: Il messaggio non abbia subito modifiche o manomissioni**

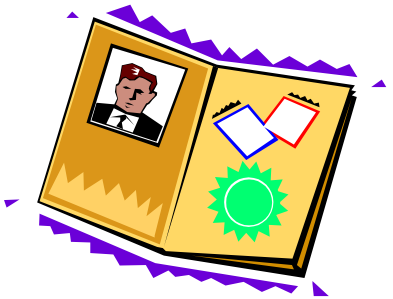
Thanks to Giampaolo Bella for slides draft!!

Autorità di certificazione

- Chi garantisce che la chiave pubblica di Bob, che otteniamo da un registro pubblico, sia stata rilasciata proprio a Bob?
- Una terza parte fidata: l'autorità di certificazione (**CA**), che certifica il legame utente/chiave pubblica mediante apposito certificato digitale

Thanks to Giampaolo Bella for slides draft!!

Certificato reale



- Cartaceo
  - *Carta d'identità, etc.*
- Emesso da un'autorità riconosciuta
- **Associa l'identità** di una persona (nome, cognome, data di nascita, …) al suo **aspetto fisico** (foto)

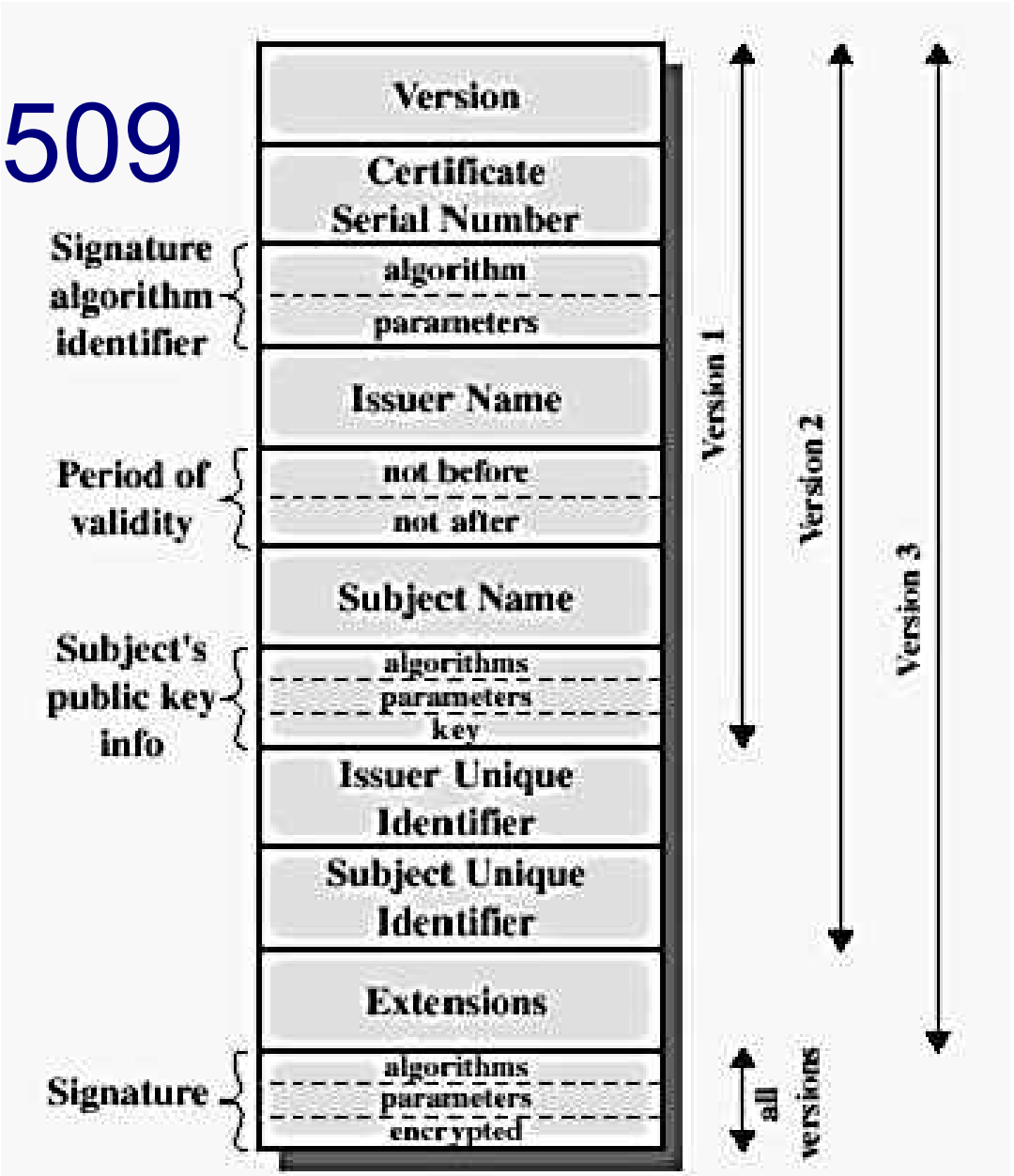
Thanks to Giampaolo Bella for slides draft!!

Certificato digitale

- Elettronico
- Associa **l'identità** di una persona ad una **chiave pubblica**
- Emesso da una CA riconosciuta
- Firmato con la chiave privata della CA
- Formato tipico: *X.509*
  - Raccomandato dall'ITU (International Telecommunication Union)

Thanks to Giampaolo Bella for slides draft!!

Certificato X.509 - struttura



I 10 compiti di una CA

1. Identificare con certezza la persona che fa richiesta della certificazione della chiave pubblica
2. Rilasciare e rendere pubblico il certificato
3. Garantire l'accesso telematico al registro delle chiavi pubbliche
4. Informare i richiedenti sulla procedura di certificazione e sulle tecniche per accedervi
5. Dichiarare la propria politica di sicurezza

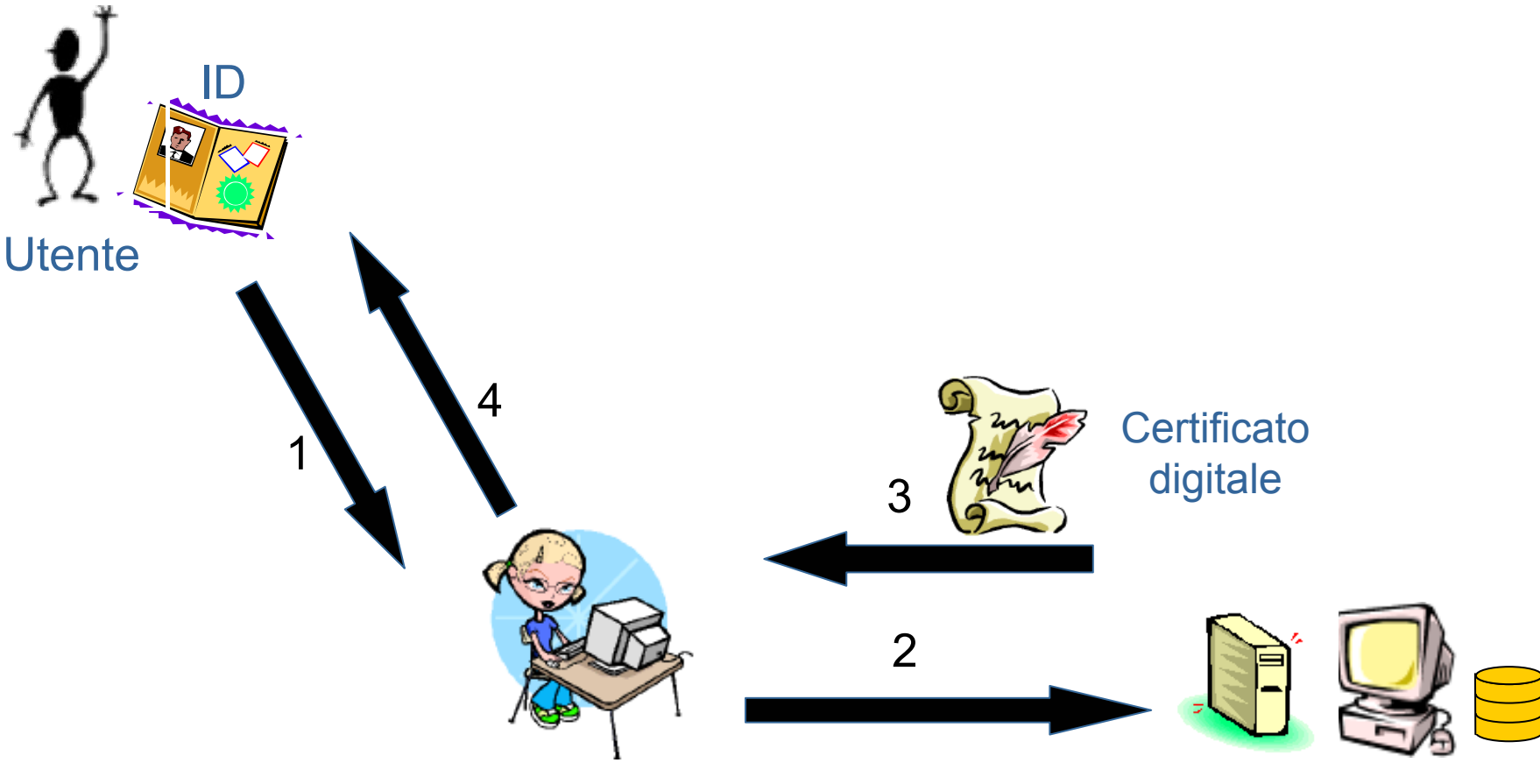
Thanks to Giampaolo Bella for slides draft!!

I 10 compiti di una CA

6. Attenersi alle norme sul trattamento di dati personali
7. Non rendersi depositario delle chiavi private
8. Procedere alla revoca o alla sospensione dei certificati in caso di richiesta dell'interessato o venendo a conoscenza di abusi o falsificazioni, ecc.
9. Rendere pubblica la revoca o la sospensione delle chiavi.
10. Assicurare la corretta manutenzione del sistema di certificazione

Thanks to Giampaolo Bella for slides draft!!

Ottenere un certificato digitale



Ottenere un certificato digitale

1. L'utente genera sul proprio PC una coppia di chiavi
   - I browser comuni offrono il servizio (*Netscape, Explorer*)
   - La chiave privata è memorizzata localmente in un file nascosto (o floppy disk)
   - Maggiore sicurezza: generare la coppia di chiavi tramite SmartCard collegata al PC - la chiave privata non esce mai dalla SmartCard (protetta da PIN)
2. L'utente invia alla CA una richiesta di certificato, insieme alla chiave pubblica generata (a meno che non sia la CA a generare la coppia di chiavi per l'utente)

Thanks to Giampaolo Bella for slides draft!!

Ottenere un certificato digitale

3. La CA autentica il richiedente, di solito chiedendogli di recarsi di persona ad uno sportello di **LVP** (Local Validation Point) collegato con la CA
4. Verificata l'identità, la CA emette il certificato, lo invia al richiedente tramite posta elettronica ed inserisce la chiave certificata nel registro delle chiavi pubbliche

L'intera procedura accade nell'ambito di una **PKI** (Public Key Infrastructure)

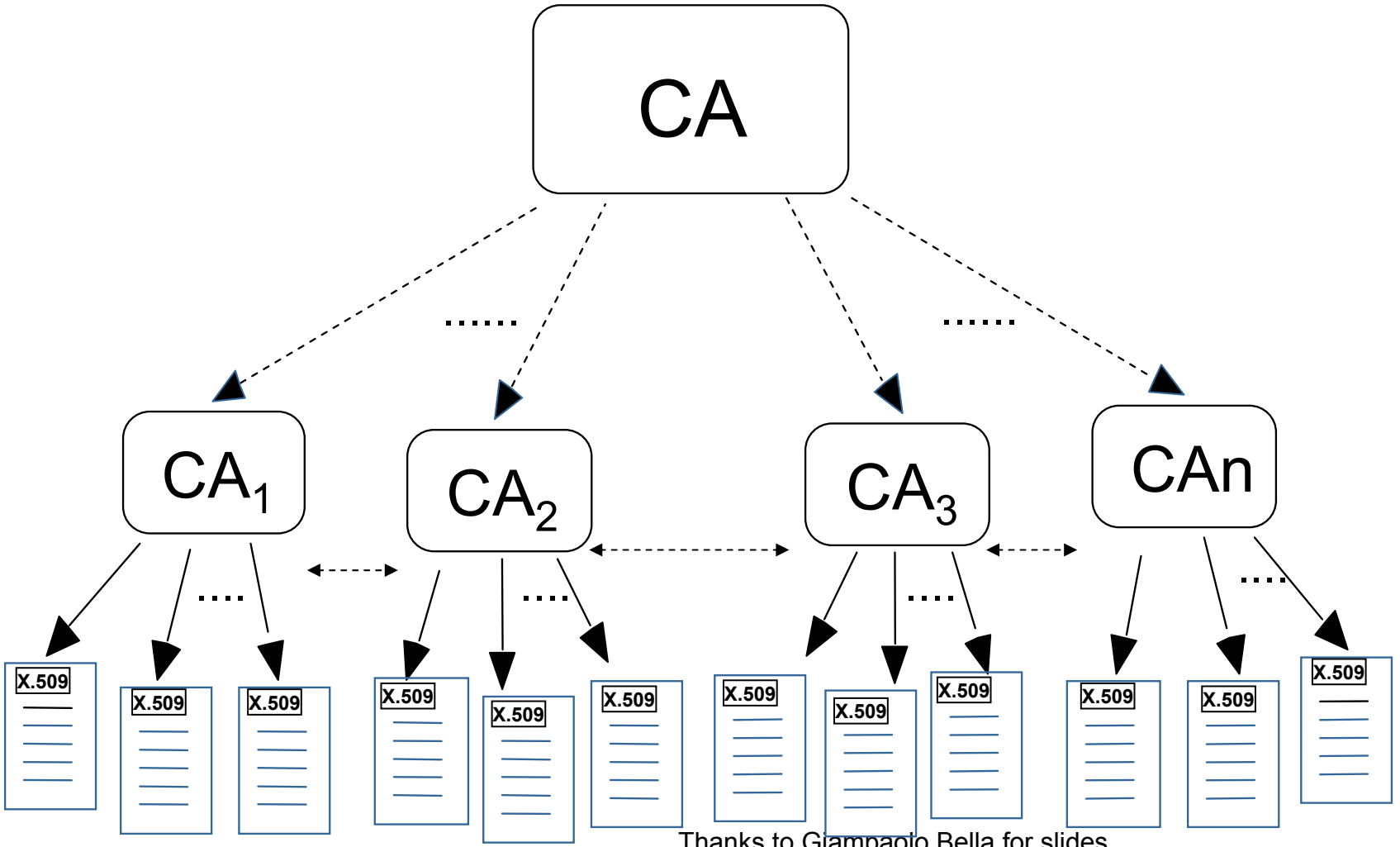
Thanks to Giampaolo Bella for slides draft!!

PKI (Public Key Infrastructure)

- Struttura minima: CA+LVP. Ammesse più LVP
  - LVP è lo sportello per l'autentica classica dell'utente; LVPO il suo operatore
- Struttura gerarchica: alcune CA certificano altre, ottenendo una "catena di fiducia"
  - Struttura ad albero
  - La **Root CA** certifica le CA di primo livello
  - Le primo livello certificano le CA di secondo livel
  - Le CA di ultimo livello certificano il singolo utente

Thanks to Giampaolo Bella for slides draft!!

PKI a struttura gerarchica



Thanks to Giampaolo Bella for slides draft!!

Revoca del certificato

- Varie ragioni
  - *Cambio dei dati personali (email, recapito, etc)*
  - *Licenziamento, dimissioni*
  - *Compromissione della chiave privata…*
  - *…*
- Richiesta di revoca (cessazione di validità)
  - Dall'utente
  - Dall'emettitore (LVPO)
- Revoca mediante **CRL** (Certificate Revocation List)

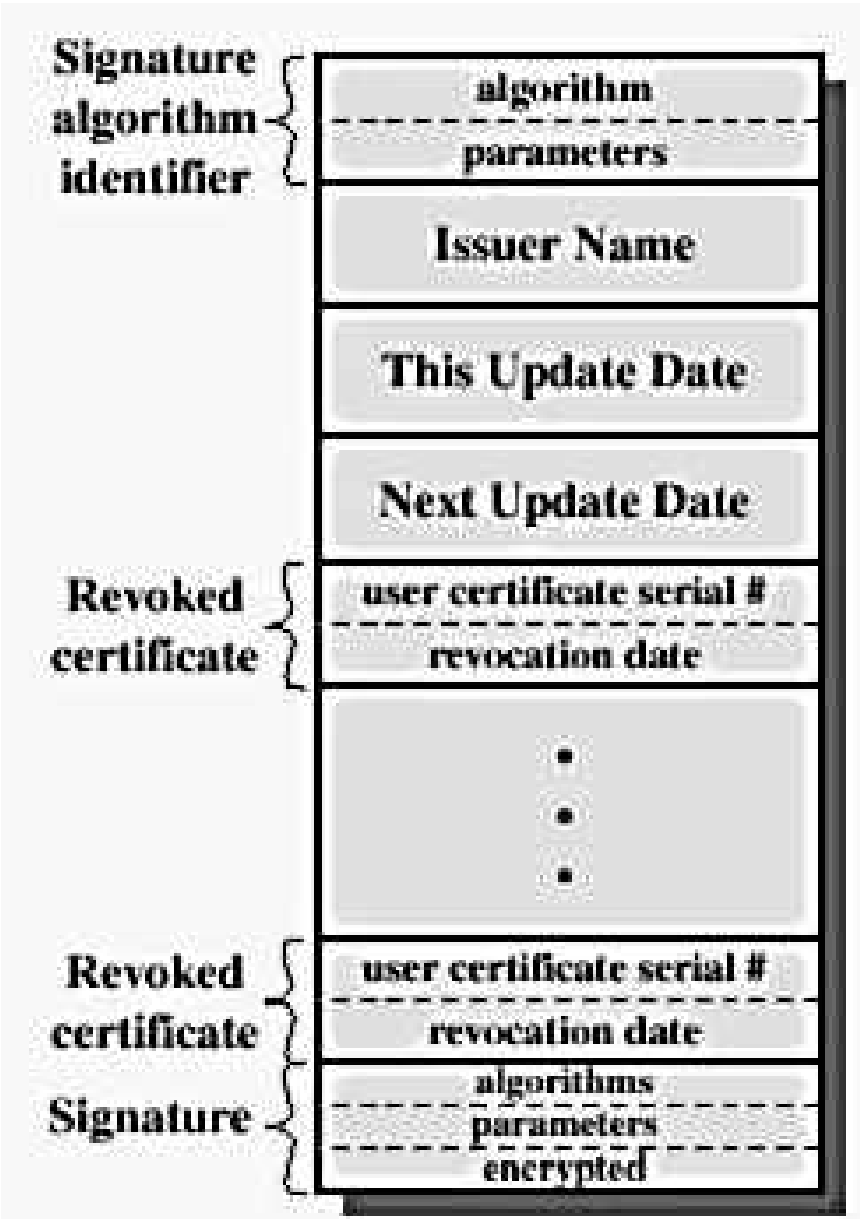
Thanks to Giampaolo Bella for slides draft!!

CRL (Certificate Revocation List)

- Lista di certificati revocati prima della loro naturale scadenza temporale
- Firmata digitalmente dalla stessa CA che ha emesso il certificato ora revocato
- Un LVPO emette una **CRR** (Certificate Revocation Request) per 1 particolare certificato
- La CA relativa emetterà la nuova CRL

Thanks to Giampaolo Bella for slides draft!!

CRL - struttura



(b) Certificate Revocation List

CRL - esempio

Certificate Revocation List (CRL):

Version 1 (0x0)

Signature Algorithm: md5WithRSAEncryption

Issuer: /Email=pki-ra-staff@iit.cnr.it/CN=IIT PKI-RA/OU=PKI- RA STAFF/O=IIT/C=IT

Last Update: Sep 2 07:25:49 2002 GMT

Next Update: Sep 9 07:25:49 2002 GMT

Revoked Certificates: Serial Number: 02 Revocation Date: Aug 27 08:26:46 2002 GMT Serial Number: 12 Revocation Date: Sep 2 07:25:18 2002 GMT Serial Number: 13 Revocation Date: Sep 2 07:25:31 2002 GMT Signature Algorithm: md5WithRSAEncryption
3f:13:45:5a:bc:fc:f4:e5:1b:e2:c1:4c:02:69:1c:43:02:e6:
11:84:68:64:6e:de:41:fa:45:58:4e:1d:44:a7:c5:91:7d:28:

.....

-----BEGIN X509 CRL-----
MIIB8zCB3DANBgkqhkiG9w0BAQQFADBvMSYwJAYJKoZIhvcNAQkBFhdwa2
ktcmEt

.....

c3RhZmZAaWl0LmNuci5pdDETMBEGA1UEAxMKSKKxV4RCKffBP9zW5t1IKx
5J7cdG

-----END X509 CRL-----

Thanks to Giampaolo Bella for slides

Dalla crittografia alla sicurezza

- Gli strumenti crittografici visti sono usati per risolvere vari problemi di sicurezza (ottenere le relative proprietà di sicurezza)
  - *Combinazioni di segretezza, autenticazione, integrità: crittografia asimmetrica o firma digitale*
  - *Sessione di comunicazione segreta: PKI + chiave di sessione*
  - *…*
- Si crea un **protocollo di sicurezza**, un preciso schema di eventi che possibilmente fanno uso di crittografia

Thanks to Giampaolo Bella for slides draft!!

Protocollo – esempio 1

- Protocollo 1 per acquistare un bene di valore
  1. Il venditore consegna la merce al cliente
  2. Il cliente compila un assegno e lo consegna al venditore
  3. Il venditore deposita l'assegno in banca
- E se l'assegno fosse scoperto?
- Il protocollo non garantisce le sperate proprietà di sicurezza

Thanks to Giampaolo Bella for slides draft!!

Protocollo – esempio 2

- Protocollo 2 per acquistare un bene di valore
  1. Il cliente si reca in banca e chiede il rilascio di un assegno circolare
  2. La banca verifica la disponibilità sul conto corrente del cliente e in caso affermativo rilascia l'assegno al cliente
  3. Il venditore consegna la merce al cliente
  4. Il cliente consegna l'assegno circolare al venditore
  5. Il venditore deposita (e incassa) l'assegno

Thanks to Giampaolo Bella for slides draft!!

Un problema di sicurezza

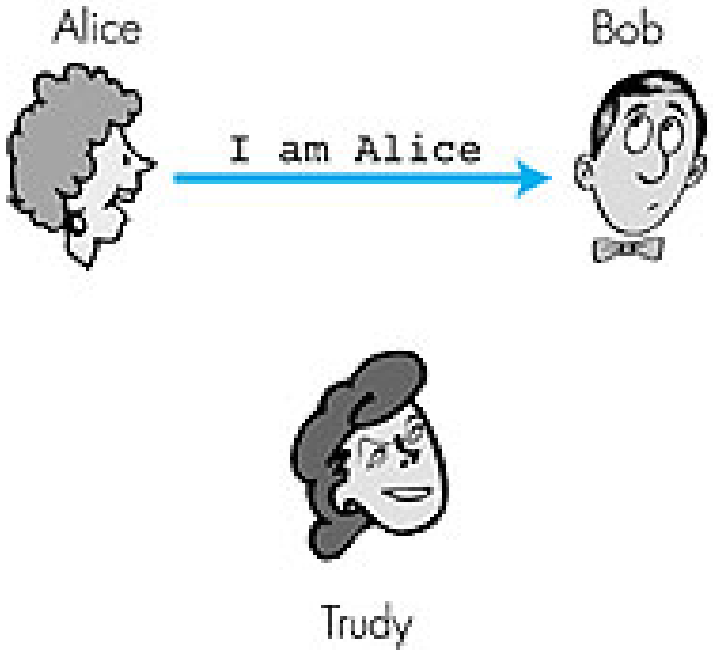
- L'Autenticazione di utenti remoti

t paolo Bella for slides draft!!

Autenticazione

Obiettivo: Bob vuole che Alice sia in grado di dimostrare la propria identità

Protocollo ap1.0: Alice says "I am Alice"

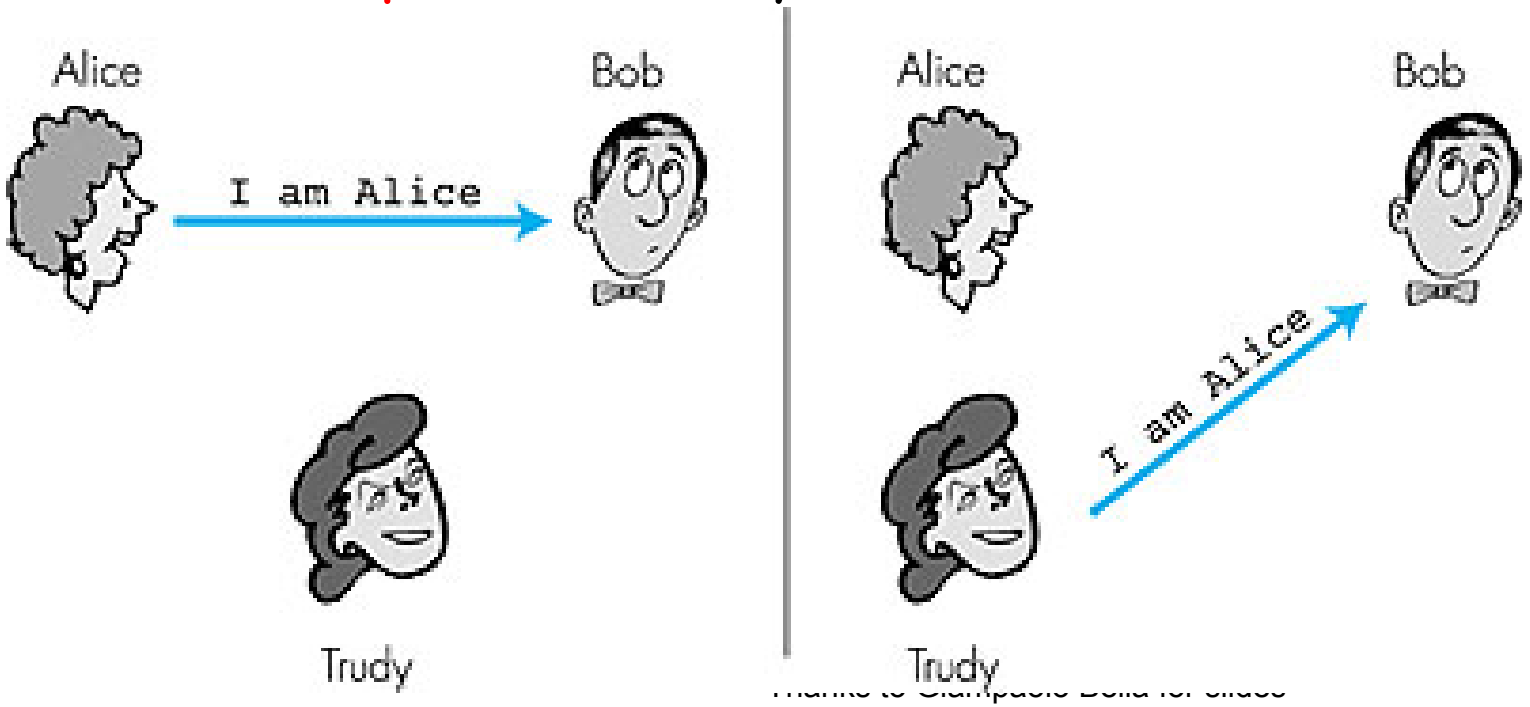


L'intruder è in grado di Inserirsi nel protocollo

Autenticazione

Obiettivo: Bob vuole che Alice sia in grado di dimostrare la propria identità

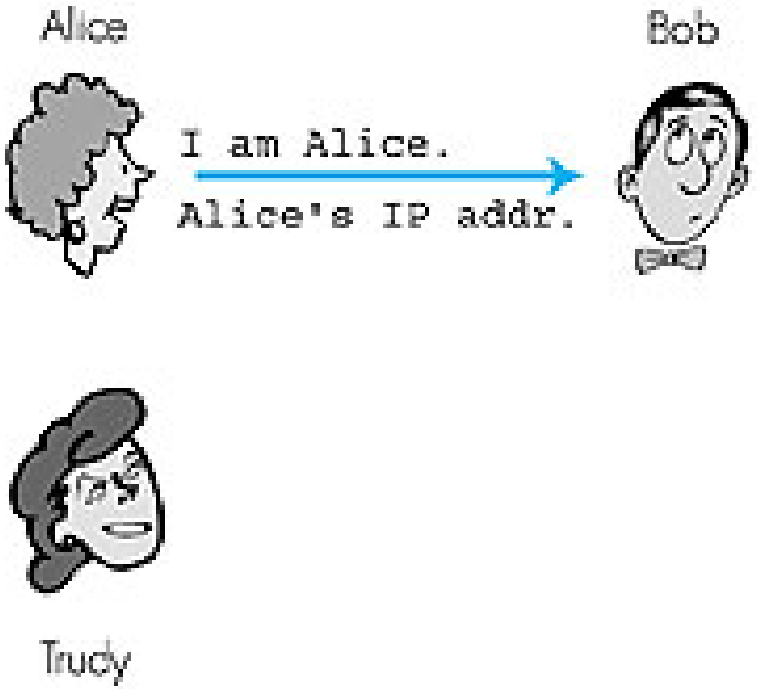
Protocollo ap1.0: Alice says "I am Alice"



draft!!

Autenticazione: proviamo nuovamente

Protocollo ap2.0: Alice says "I am Alice" and sends her I
address along to "prove" it

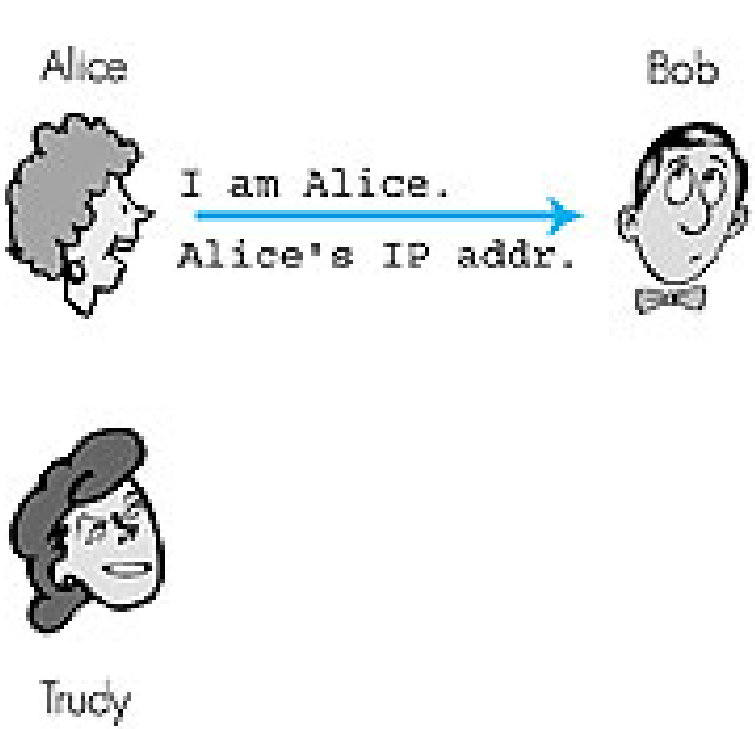


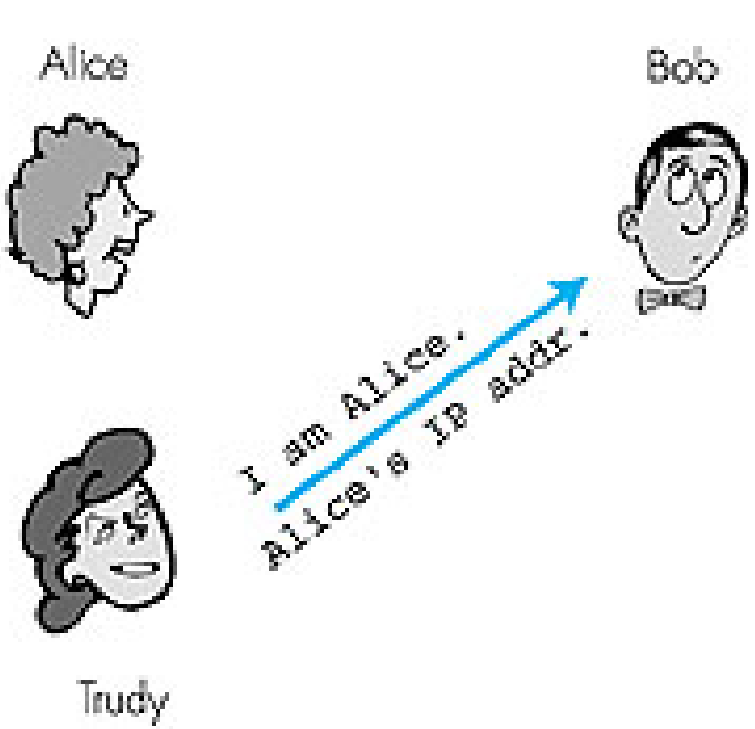
Cosa può fare l'intruder?

Thanks to Giampaolo Bella for slides draft!!

Autenticazione: proviamo nuovamente

Protocollo ap2.0: Alice says "I am Alice" and sends her I address along to "prove" i

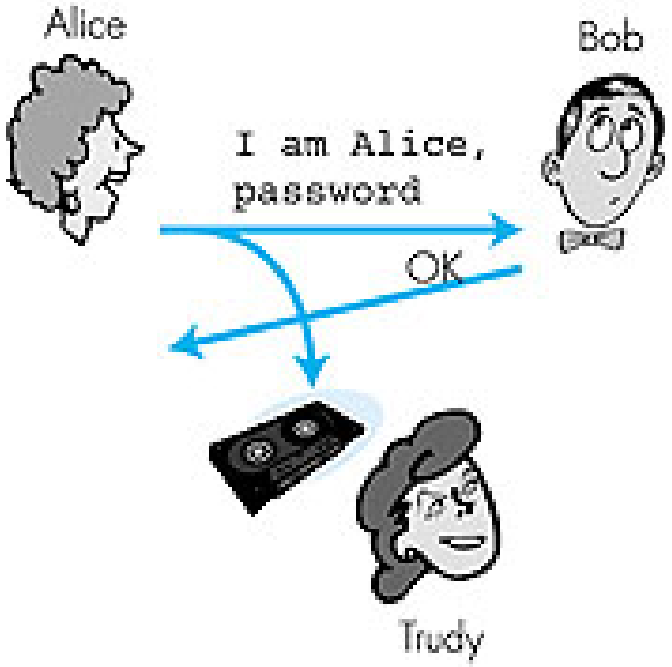




Thanks to Giampaolo Bella for slides draft!!

Autenticazione

Protocollo ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.

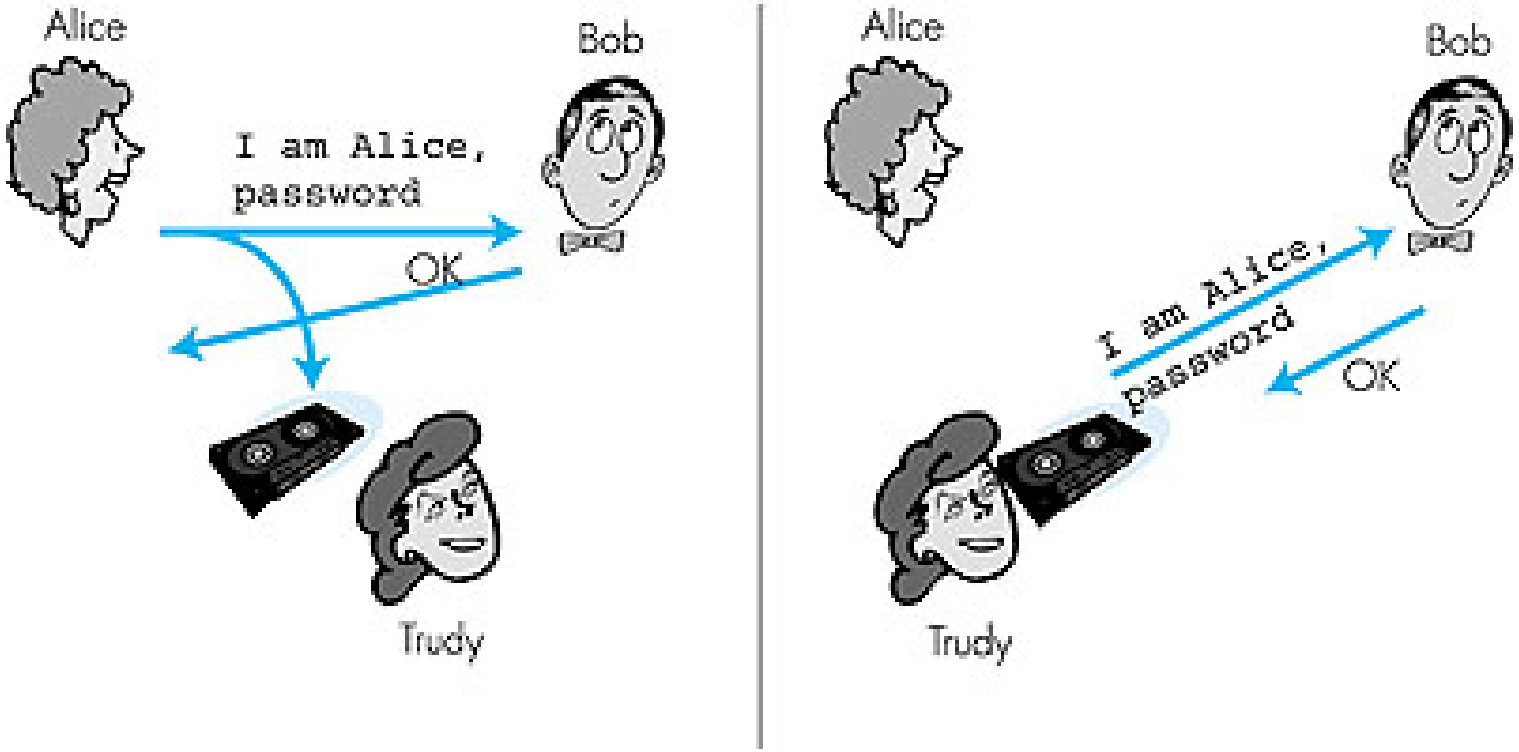


Cosa può fare l'intruder

Thanks to Giampaolo Bella for slides draft!!

Autenticazione

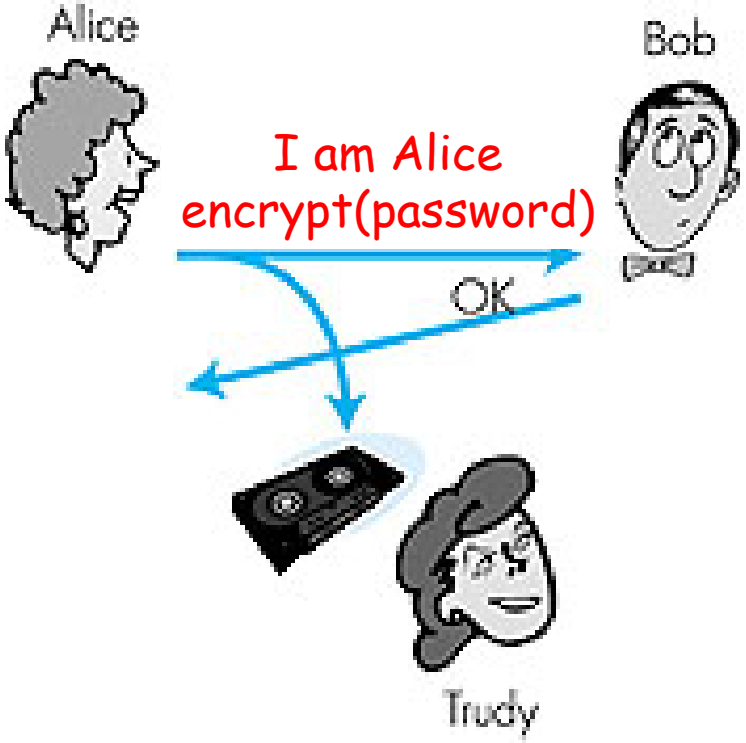
Protocollo ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.



Thanks to Giampaolo Bella for slides draft!!

Autenticazione

Protocollo ap3.1: Alice says "I am Alice" and sends her *encrypted* secret password to "prove" it.

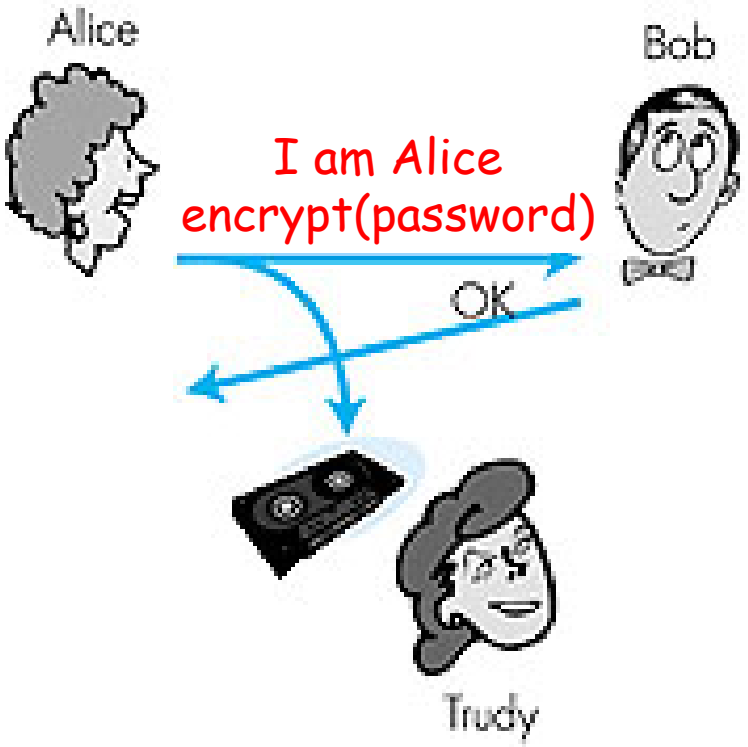


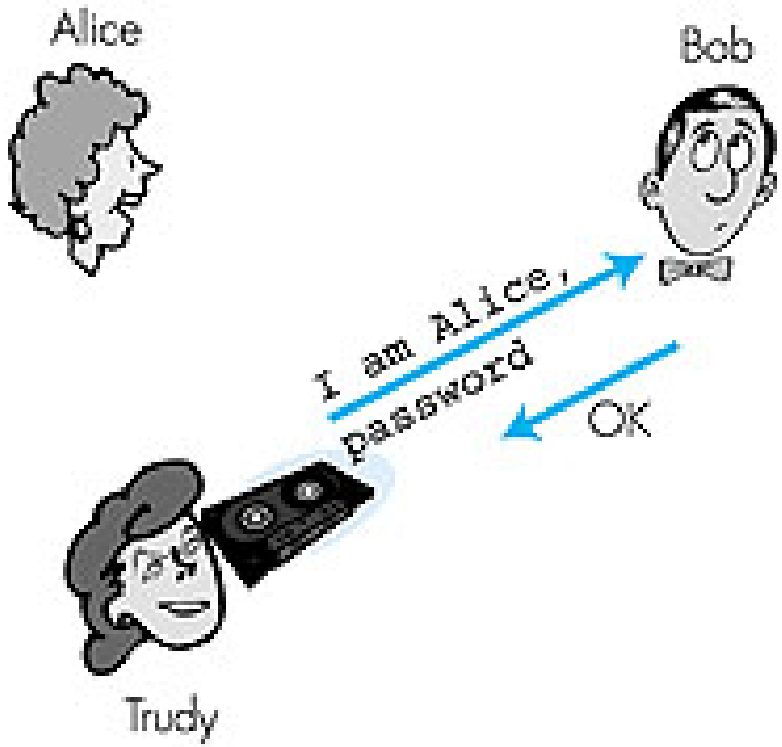
Intruder: attacco di replica

Thanks to Giampaolo Bella for slides draft!!

Autenticazione

Protocollo ap3.1: Alice says "I am Alice" and sends her *encrypted* secret password to "prove" it.



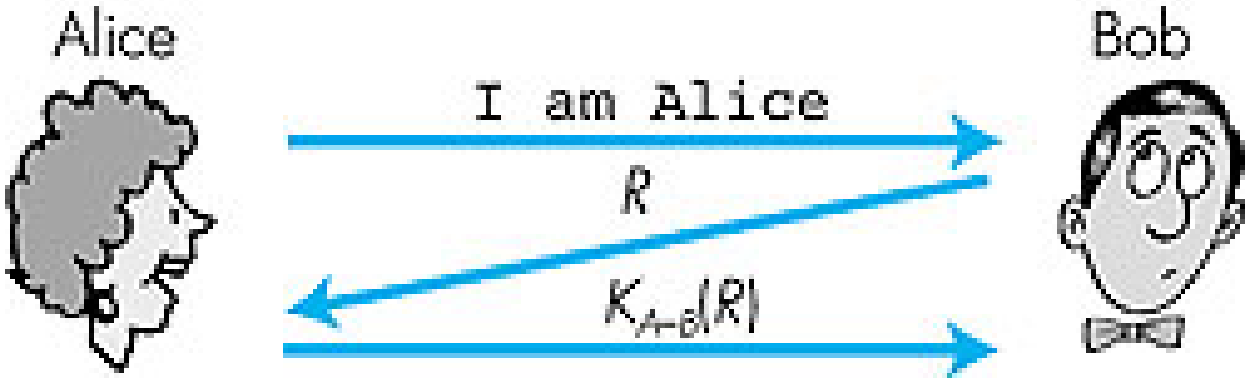


Thanks to Giampaolo Bella for slides draft!!

Autenticazione

Nonce: numero (R) usato una sola volta (onlyonce)

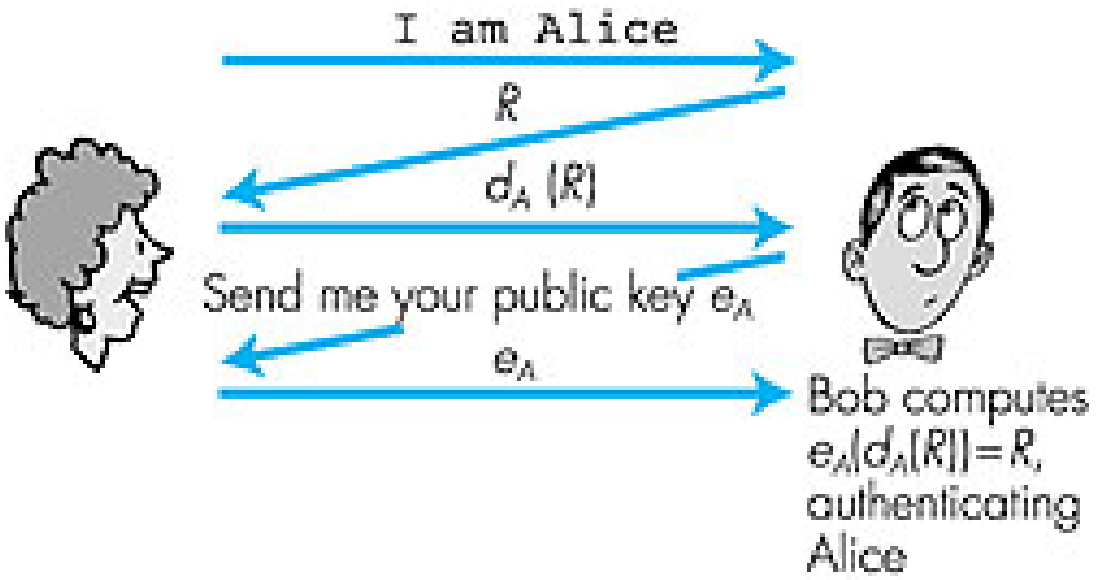
ap4.0: to prove Alice "live", Bob sends Alice nonce, R. Alice must return R, encrypted with shared secret key



Thanks to Giampaolo Bella for slides draft!!

Autenticazione: ap5.0

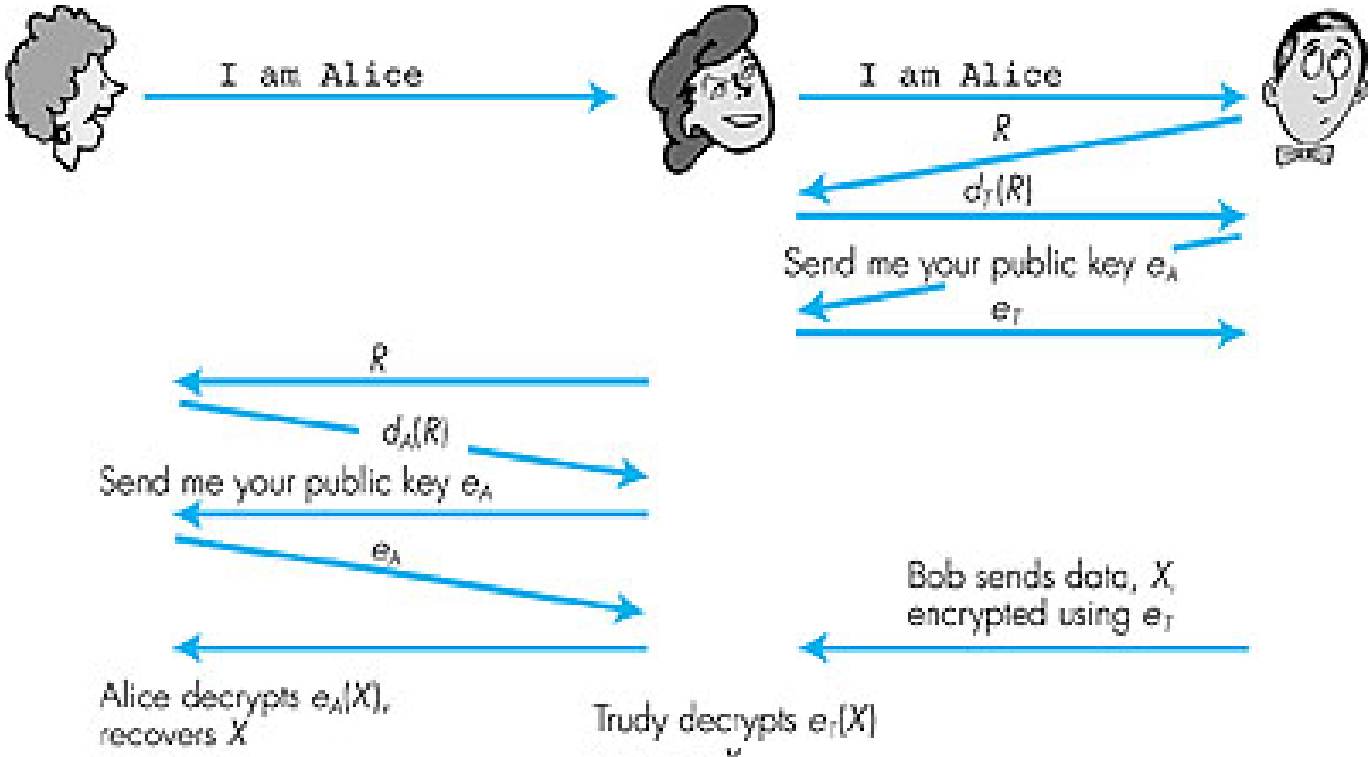
ap5.0: nonce e chiave pubblica



Thanks to Giampaolo Bella for slides draft!!

ap5.0: violazione

Man (woman) in the middle attack:



Thanks to Giampaolo Bella for slides draft!!

Un problema di sicurezza

- L'Autenticazione di utenti remoti

paolo Bella for slides
draft!!

Capacità della spia DY

1. Intercettare messaggi e prevenirne il recapito
2. Rimbalzare a piacere i messaggi intercettati
3. Imparare i testi in chiaro e i testi codificati
4. Tentare di decriptare con tutte le chiavi note
5. Utilizzare le proprie credenziali legali
6. Ottenere certe credenziali illegalm. (*pagando*)
7. Creare messaggi fasulli da componenti note

Tranne violare crittotesti!

Thanks to Giampaolo Bella for slides draft!!

Un problema di sicurezza

- L'Autenticazione di utenti remoti

- Soluzione: scambio di messaggi crittografici secondo un **preciso** protocollo di sicurezza

Thanks to Giampaolo Bella for slides draft!!

Messaggi

Atomici

1. Nomi di utenti: A, B, C, …
2. Chiavi crittografiche
   - a lungo temine: K_a, K_b, …
   - a breve termine: K_{ab},… (chiavi di sessione)
3. Nonce: N_a, N_b, …
4. Timestamp: T_a, T_b, …
5. Digest
6. Label: “trasferisci denaro”, “collegati alla porta xy”,…

Compositi

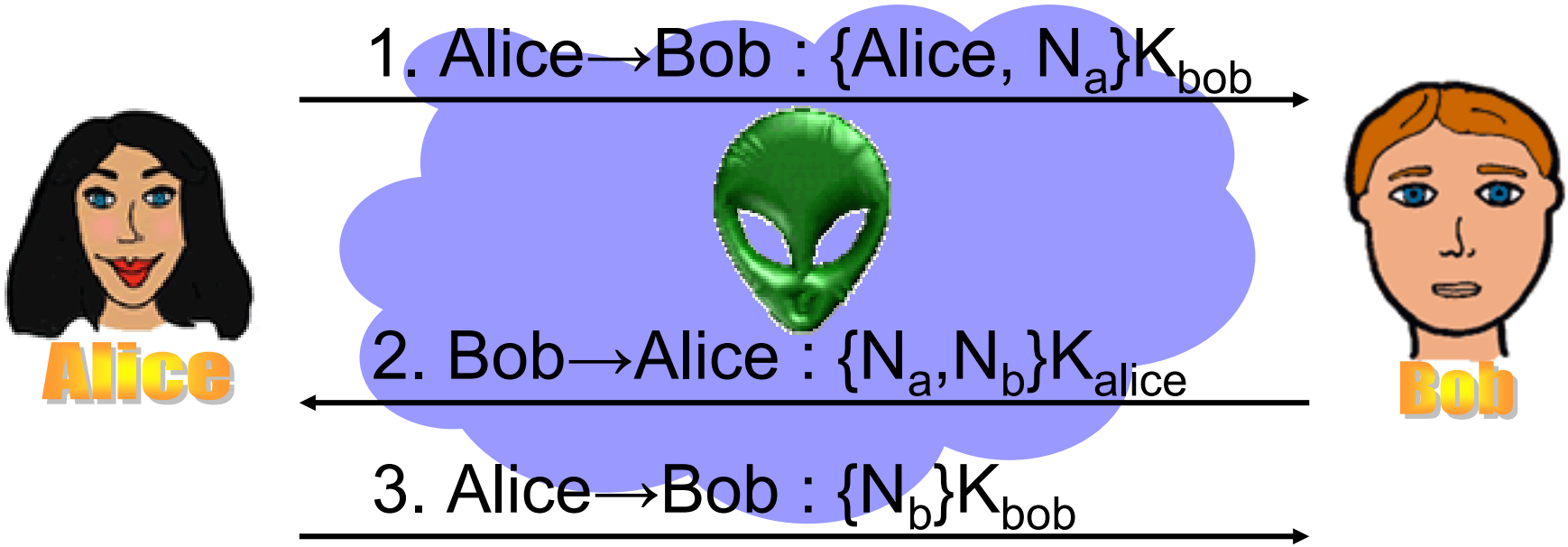
1. Concatenati: m,m’, …
2. Criptati: m_K, {m,m’}K, …

Problema: autenticazione di un messaggio concatenato?

Thanks to Giampaolo Bella for slides draft!!

Protocollo di sicurezza – esempio

- Dovuto a Needham-Schröder, 1978
- Presuppone una PKI con crittografia perfetta



Thanks to Giampaolo Bella for slides draft!!

Obiettivi di sicurezza del protocollo (goal)

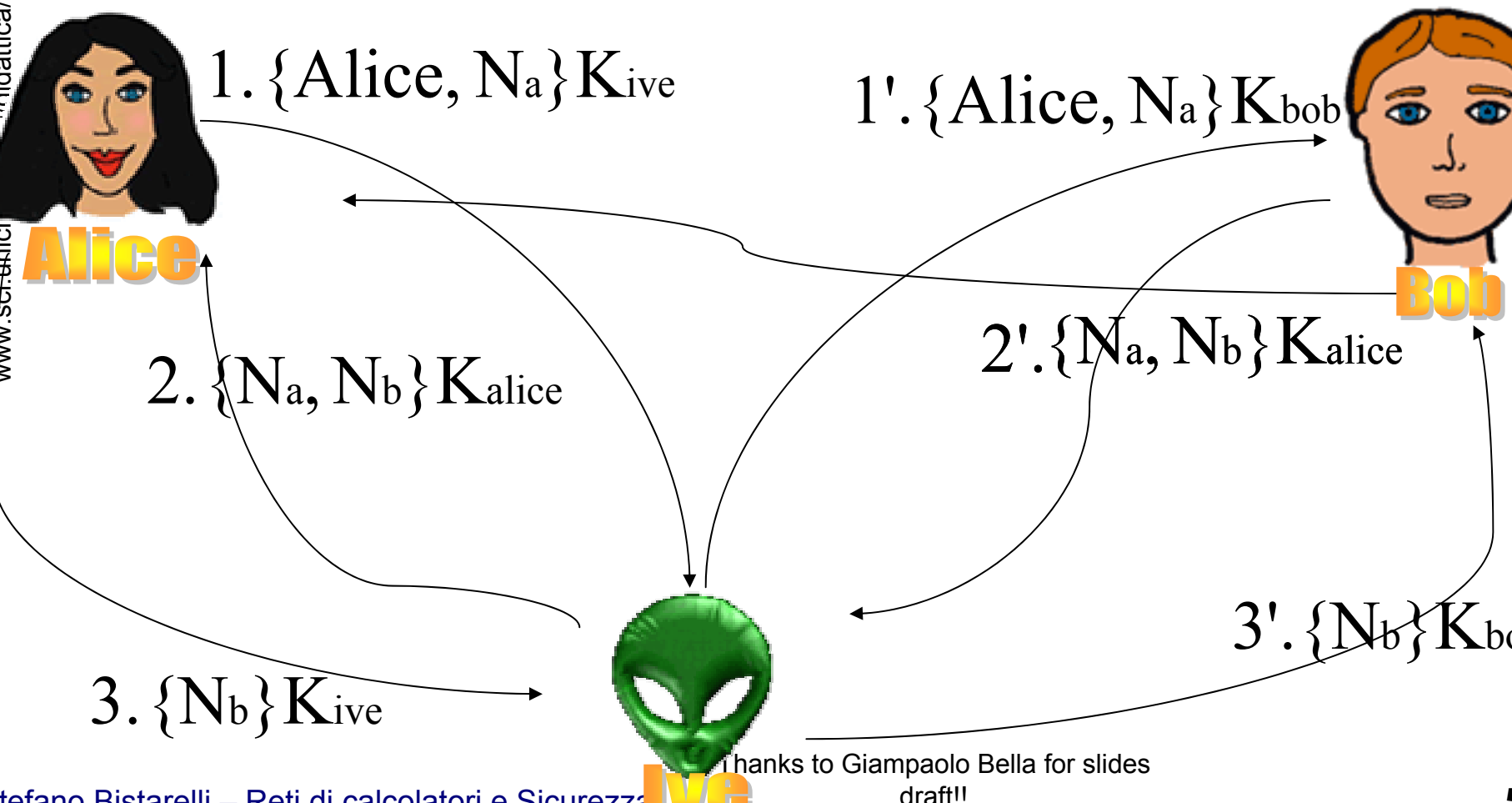
1. Alice→Bob : $\{Alice, N_a\}K_{bob}$
2. Bob→Alice : $\{N_a,N_b\}K_{alice}$
3. Alice→Bob : $\{N_b\}K_{bob}$

1. Autenticazione reciproca degli utenti
   - Etichette mittente e ricevente inaffidabili!
   - Autenticazione garantita da segretezza delle nonce
2. Segretezza delle nonce scambiate

Thanks to Giampaolo Bella for slides draft!!

Gli obiettivi falliscono!

1. Alice→Bob : $\{Alice, N_a\}K_{bob}$
2. Bob→Alice : $\{N_a,N_b\}K_{alice}$
3. Alice→Bob : $\{N_b\}K_{bob}$



Thanks to Giampaolo Bella for slides draft!!

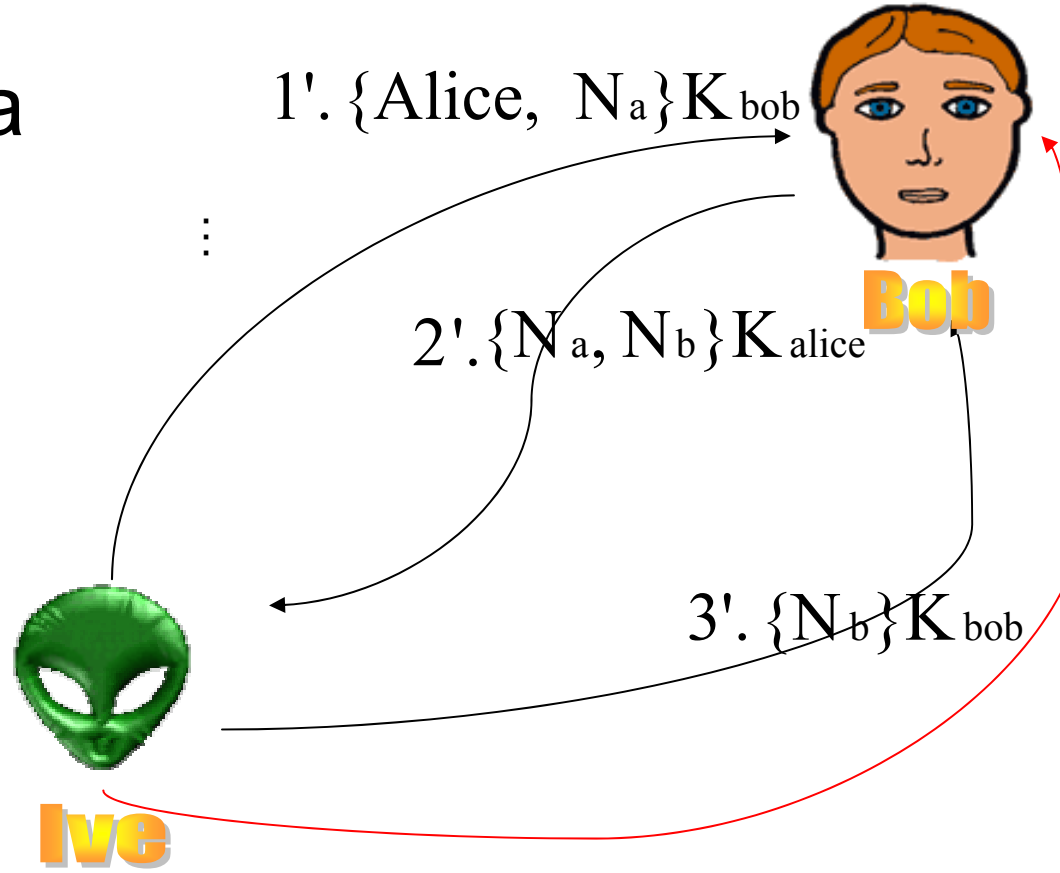
Gli attacchi visti (attacco di Lowe, 1995

- 2 sessioni **interlacciate**
- Nell'ipotesi che alice cominci con la spia
- Attivi, da **posizione intermedia**
  - Segretezza di Nb fallisce col passo 3
  - Autenticazione di Alice con Bob fallisce col passo 3'. Come??
- Sicurezza (segretezza, autenticazione) fallita anche nell'ipotesi di crittografia perfetta!!

Thanks to Giampaolo Bella for slides draft!!

Conseguenze dell'attacco

- Se Bob fosse una banca e gli altri due correntisti…
- Se Alice fosse il docente e gli altri due studenti…



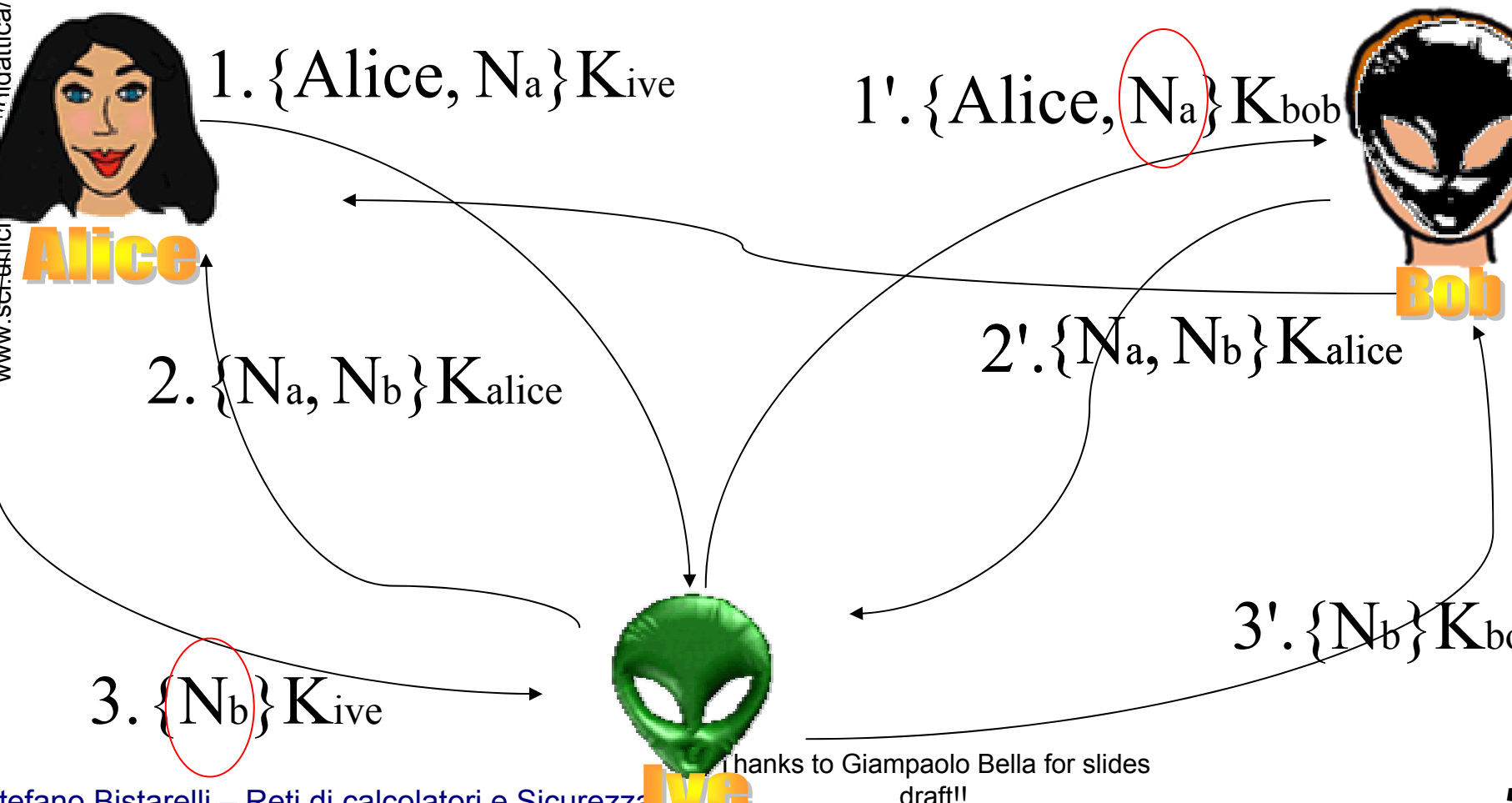
4'. {Na, Nb, "trasferisci 10000€ dal conto di Alice al conto di Ive"}Kb

4'. {Na, Nb, "l'esame di domani è cancellato"}Kbob

Thanks to Giampaolo Bella for slides

draft!!

Lo stesso attacco studiato nella tassonomia BUG: perse due nonce



Thanks to Giampaolo Bella for slides draft!!

Vendetta nella tassonomia BUG

- Ipotesi: Bob scopra l'importanza di N_a
- Se anche Alice è una banca, Bob può vendicarsi su Ive come segue

$$5'.\{N_a, N_b, \text{"trasferisci 20000 € dal conto di Ive al conto di Bob"}\}K_{alice}$$

Thanks to Giampaolo Bella for slides draft!!

Protocollo di sicurezza – esempio 2

- Dovuto a Woo-Lam, metà anni ’80
- Usa crittografia simmetrica
- Usa un **TTP** (Trusted Third Party), che possiede un database di tutte le chiavi
- Goal: autentica di Alice con Bob

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_b$
3. $A \rightarrow B : \{N_b\}K_a$
4. $B \rightarrow TTP : \{A, \{N_b\}K_a\}K_b$
5. $TTP \rightarrow B : \{Nb\}K_b$

Thanks to Giampaolo Bella for slides

draft!!

Un attacco su Woo-Lam

1. $A \rightarrow B : A$
2. $B \rightarrow A : N_b$
3. $A \rightarrow B : \{N_b\}K_a$
4. $B \rightarrow TTP : \{A, \{N_b\}K_a\}K_b$
5. $TTP \rightarrow B : \{Nb\}Kb$

- B vede indietro N_b
- Pertanto autentica l'utente cui l'ha associata, ossia A
- A potrebbe perfino essere off-line
- B non distingue la sessione!

1. $C \rightarrow B : A$

   1'. $C \rightarrow B : C$

2. $B \rightarrow A : N_b$

   2'. $B \rightarrow C : N_b'$

3. $C \rightarrow B : \{N_b\}K_c$

   3'. $C \rightarrow B : \{N_b\}K_c$

4. $B \rightarrow TTP : \{A, \{N_b\}K_c\}K_b$

   4'. $B \rightarrow TTP : \{C, \{N_b\}K_c\}K_b$

5. $TTP \rightarrow B : \{N_b''\}K_b$

   5'. $TTP \rightarrow B : \{N_b\}K_b$

Thanks to Giampaolo Bella for slides draft!!

Esempio con trusted third party (TTP)

Thanks to Giampaolo Bella for slides draft!!

Symmetric Needham-Schröder

1. $A \rightarrow TTP : A,B,N_a$
2. $TTP \rightarrow A : \{N_a,B,K_{ab},\{K_{ab},A\}K_b\}K_a$
3. $A \rightarrow B : \{K_{ab},A\}K_b$
4. $B \rightarrow A : \{N_b\}K_{ab}$
5. $A \rightarrow B : \{N_b-1\}K_{ab}$

ticket

- A che serve N_a?
- K_{ab} è **chiave di sessione**
- Mutua autentica mediante passi 4 e 5

Thanks to Giampaolo Bella for slides draft!!

Replay attack

Def. Spacciare informazione (*chiavi,…*) obsoleta, magari violata, come recente

- Supponiamo che C abbia violato una vecchia chiave di sessione K_{ab} che B condivise con A

…

3. $C \rightarrow B : \{K_{ab}, A\}K_b$ (rispedito identico)
4. $B \rightarrow A : \{N_b'\}K_{ab}$ (intercettato)
5. $C \rightarrow B : \{N_b'-1\}K_{ab}$

- B autenticherebbe A e quindi accetterebbe di usare K_{ab}

Thanks to Giampaolo Bella for slides draft!!

I rischi di attacchi aumentano

- 1978: Needham-Schröder, 6 pagine
- Metà anni ’90: **SSL**, 80 pagine
- Fine anni ’90: **SET**, 1000 pagine!

Quasi vent’anni per scoprire che un protocollo di 6 pagine celava un bug! Allora…

Thanks to Giampaolo Bella for slides draft!!

Potenziali soluzioni??

- Needham-Schröder asimmetrico:

? 1. Alice→Bob : {{Alice,Na}Kalice^{-1}}Kbob

? 1. Alice→Bob : {{Alice,Na}Kalice^{-1}}Kbob
2. Bob→Alice : {{Na,Nb}Kbob^{-1}}Kalice

? 2. Bob→Alice : {{Na,Nb}Kbob^{-1}}Kalice

? 2. Bob→Alice : {Na,Nb,Bob}Kalice

? 1. Alice→Bob : {Alice,Bob,Na}Kbob

Thanks to Giampaolo Bella for slides draft!!

Potenziali soluzioni??

- Woo-Lam:
  - ? 3. A → B : {A,Nb}Ka
  - ? 5. TTP →B : {A,Nb}Kb
  - ? 4. B → TTP : {A, {A,Nb}Ka}Kb
  - ? 2. B → A : Nb,B

Thanks to Giampaolo Bella for slides draft!!

Principi di disegno: explicitness

Def. Se le identità del mittente e del ricevente sono significative per il messaggio, allora è prudente menzionarle esplicitamente

Problema: quando sono “significative”??

Thanks to Giampaolo Bella for slides draft!!