ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Non-Ripudio nella Sicurezza Digitale

Non ripudio nella Sicurezza Digitale

- Un servizio che fornisce la prova **dell'integrità e l'origine dei dati**.
- **Un'autenticazione**, a garanzia della genuinità dei dati stessi.

Non ripudio nella Sicurezza Digitale

- L'integrità dei dati -> hash dei dati garantisce una **bassissima probabilità che i dati vengano alterati**.
- L'**integrità dei dati deve essere confermata** dal destinatario

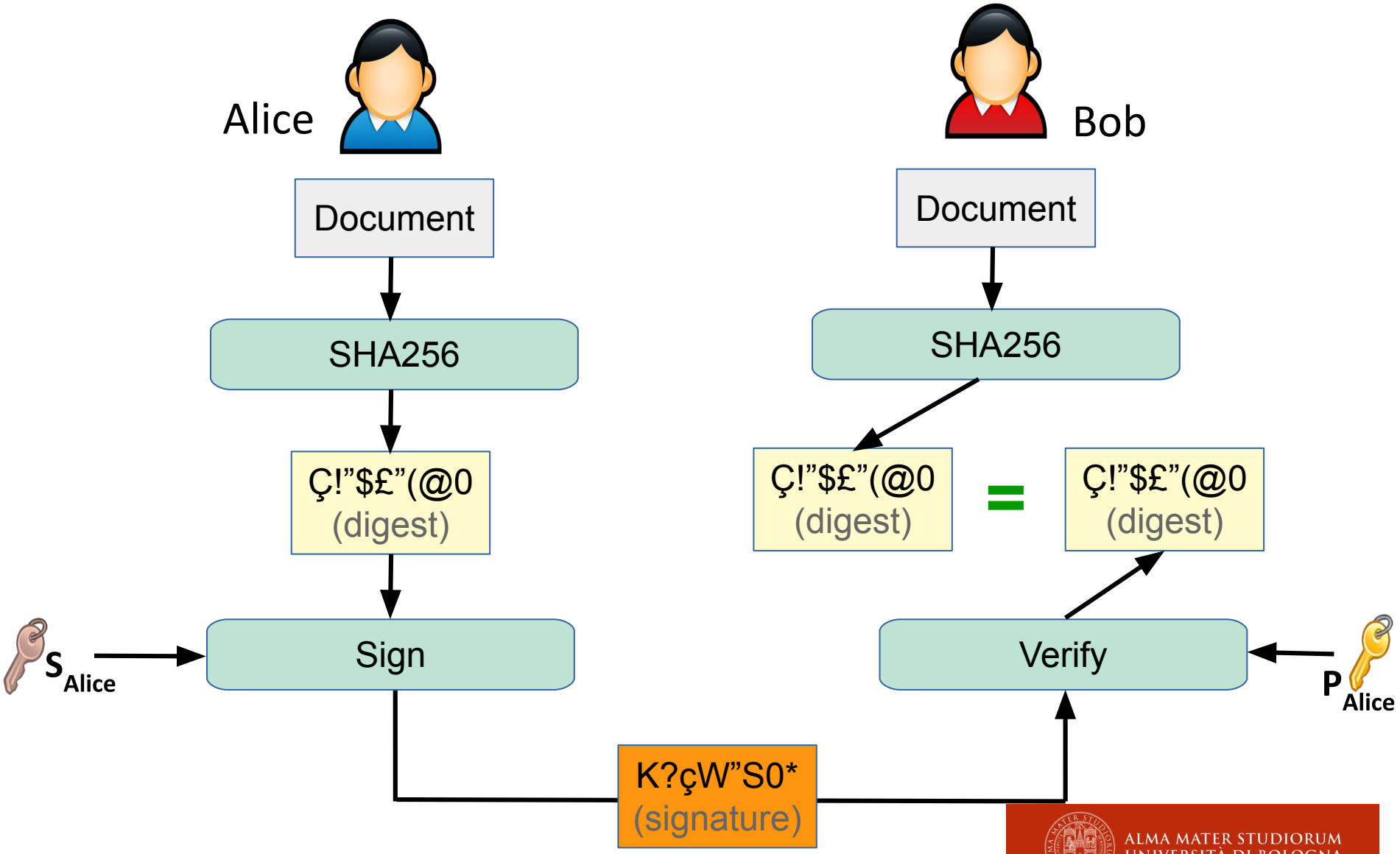
Verifica nella Sicurezza Digitale

- Il metodo più comune per la verifica dell'origine dei dati è l'utilizzo della **firma digitale** accompagnata da:
- **Certificati digitali** -> una forma di infrastruttura a chiave pubblica da cui dipende la firma digitale.

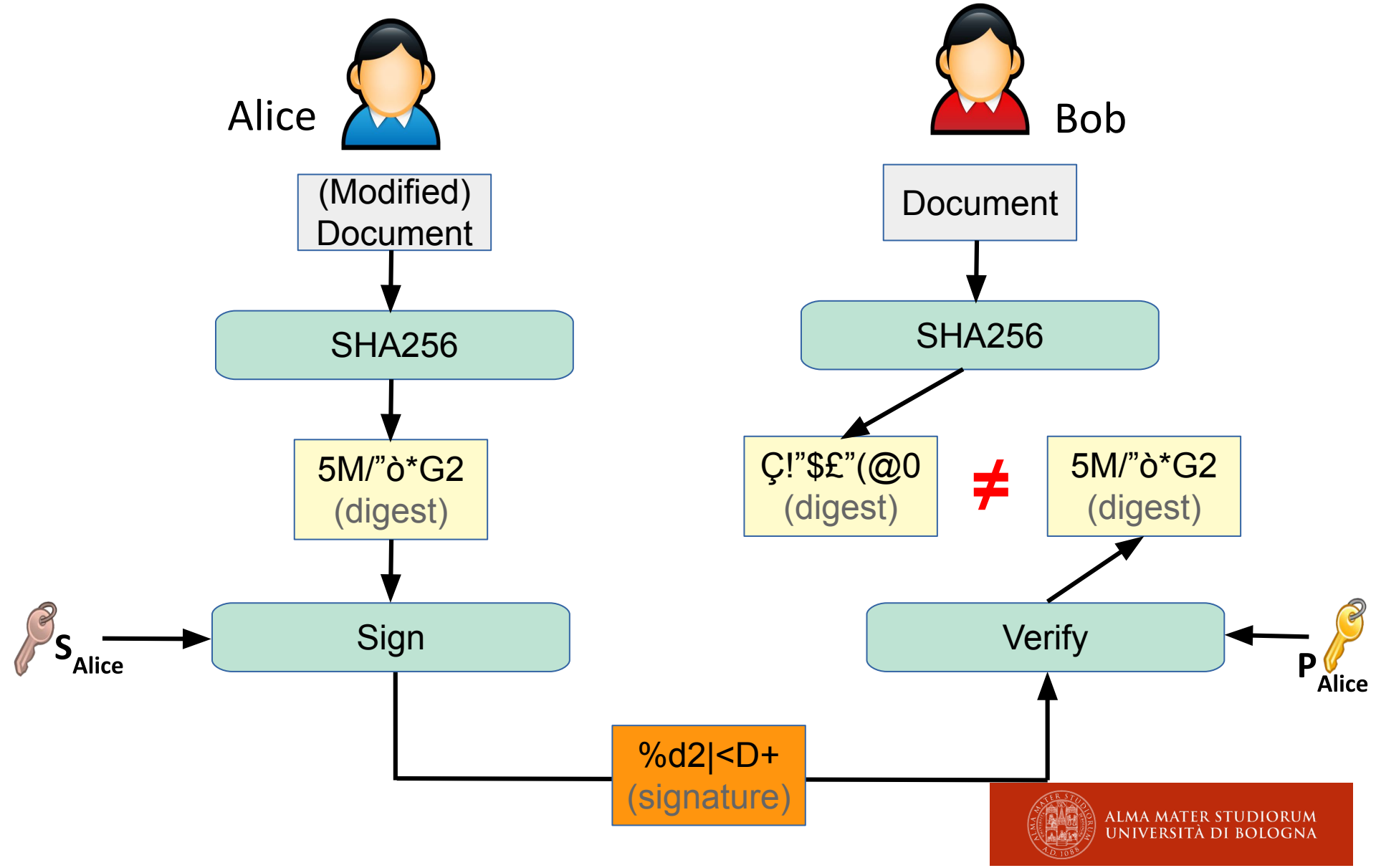
Firma Digitale

- Schema per la verifica dell'**autenticità** dei messaggi digitali (documenti).
- Impiega la crittografia **asimmetrica**
- **Integrità**: garantisce che il messaggio non sia stato alterato durante il trasporto (utilizzando il digest)
- **Autenticazione**: una firma digitale valida dà al destinatario un motivo molto forte per credere che il messaggio sia stato creato da un mittente conosciuto.

Alice firma un documento per Bob



Alice firma un documento (alterato) per Bob



Certificato Digitale

documento elettronico che attesta l'associazione univoca tra una chiave pubblica e l'identità di una persona

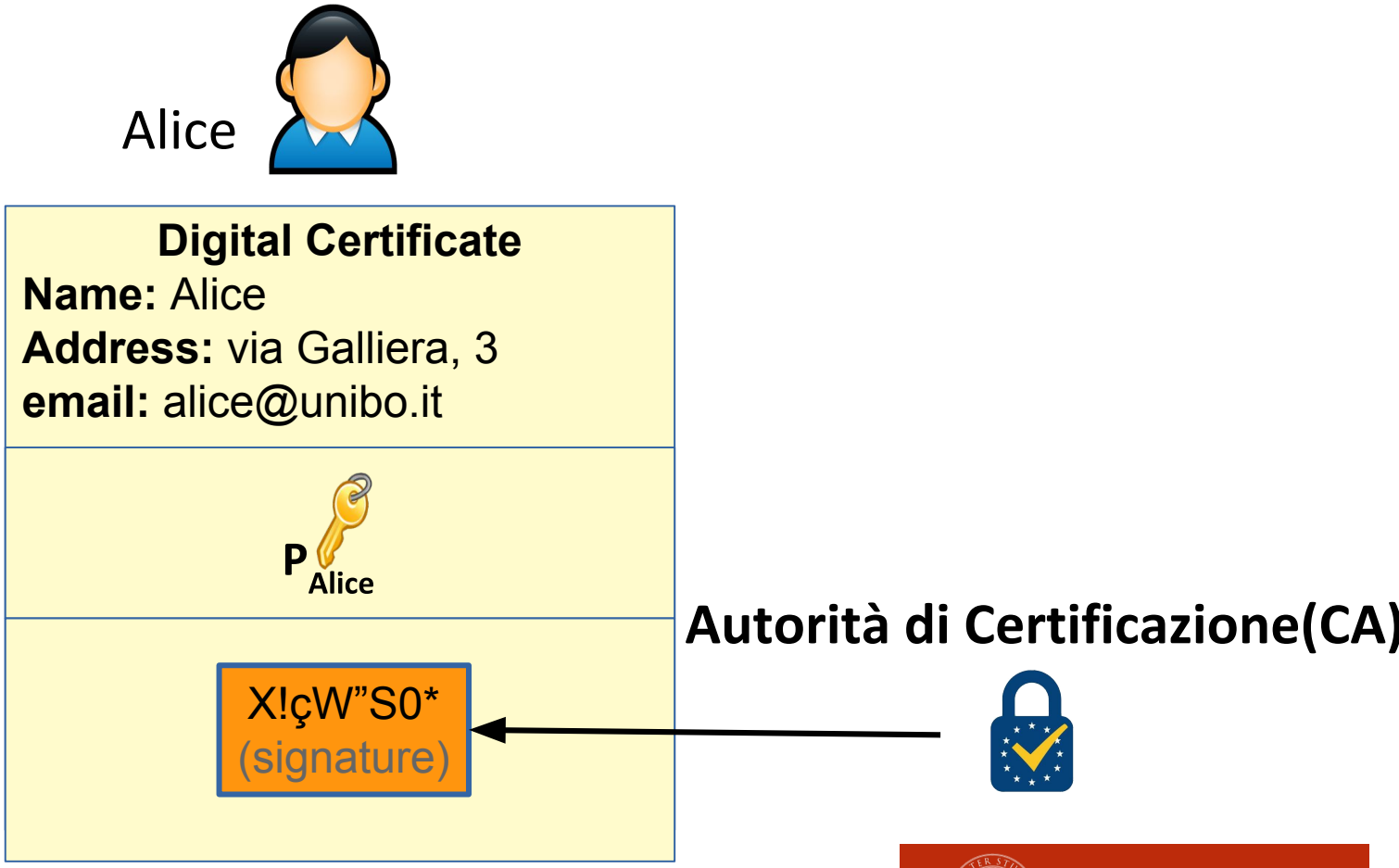
Alice 

Digital Certificate
Name: Alice
Address: via Galliera, 3
email: alice@unibo.it

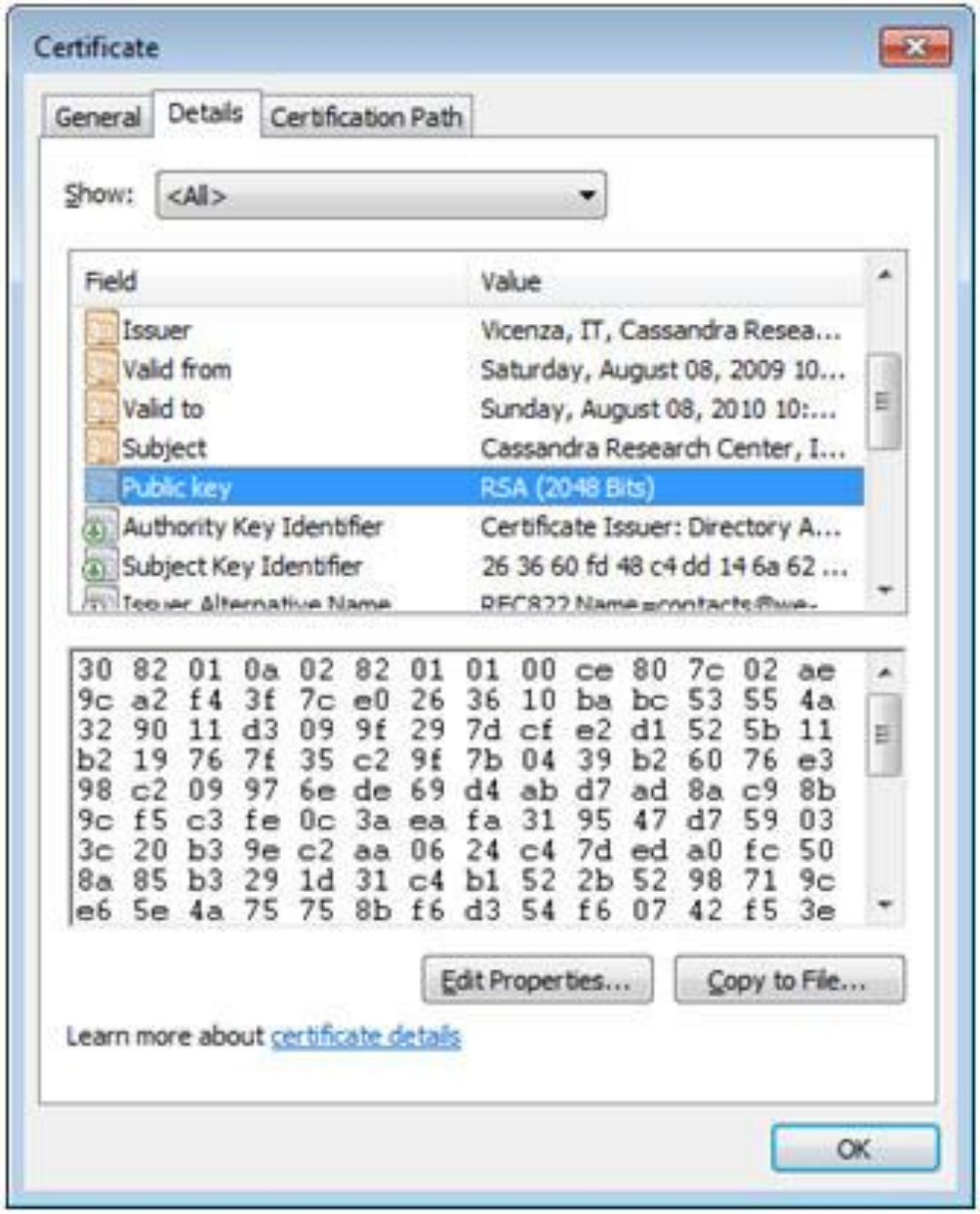
Infrastruttura a chiave pubblica (PKI) X.509

- **X.509**
  - formato più comune per i certificati digitali
- **Infrastruttura a chiave pubblica X.509** (RFC 5280) Public Key Infrastructure (PKI)
  - insieme di processi e mezzi che consentono a **terze parti fidate di verificare e/o farsi garanti** dell'identità di un utente, oltre che di associargli una chiave pubblica
- I certificati X.509 sono utilizzati in molti protocolli Internet, tra cui TLS/SSL, che è alla base di **HTTPS**, il protocollo sicuro per la navigazione in rete.

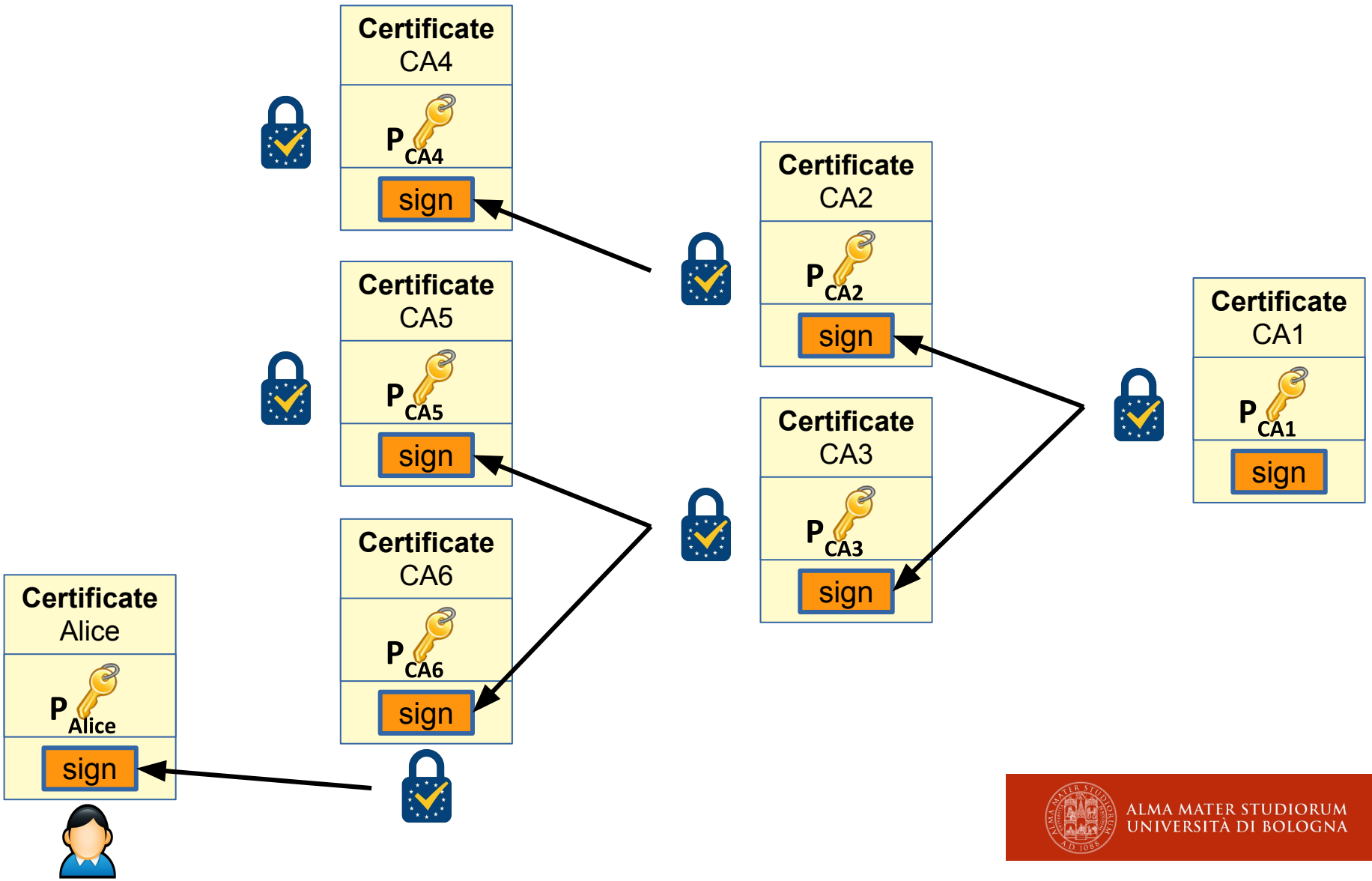
Certificato Digitale X.509



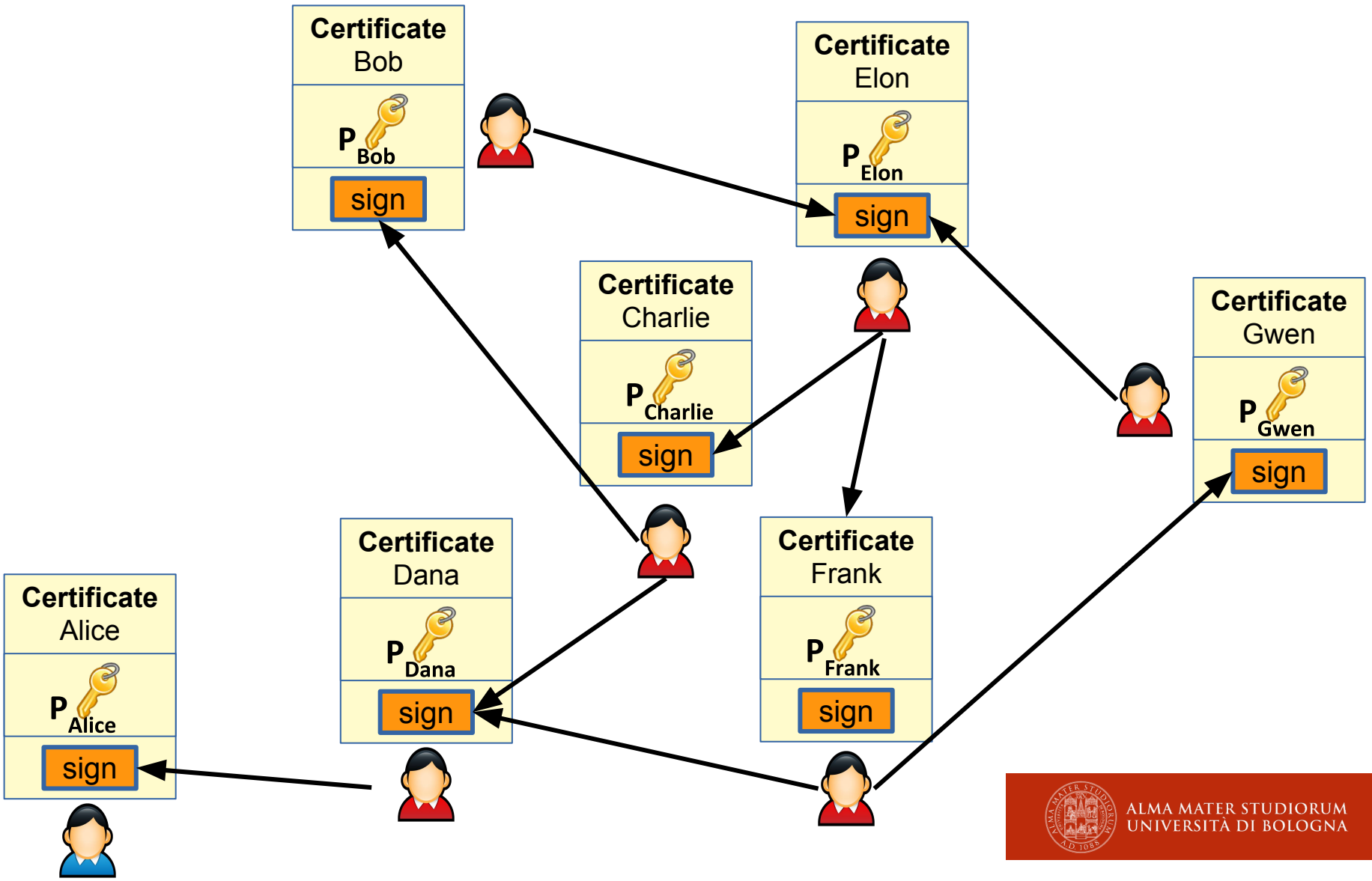
Certificato Digitale X.509



Infrastruttura a chiave pubblica X.509



Rete di fiducia (PGP)



Demo Firma Digitale

https://www.phpdocx.com/demos/digital-signature-package

electronic IDentification Authentication and Signature

- Il Regolamento UE n° 910/2014 - eIDAS
- Base normativa comune per **interazioni elettroniche sicure** fra cittadini, imprese e pubbliche amministrazioni
- **Interoperabilità a livello comunitario delle firme elettroniche e dei sistemi di validazione temporale**
  ”Una firma elettronica qualificata basata su un certificato qualificato rilasciato in uno Stato membro è riconosciuta quale firma elettronica qualificata in tutti gli altri Stati membri.” (articolo 25, comma 3)

eIDAS riconosce 3 tipi di e-signature

1. Firme Elettroniche

   Il regolamento eIDAS definisce un fondamento per tutte le firme elettroniche, affermando che **nessuna firma può essere negata legalmente soltanto per il fatto di essere in forma elettronica.**

   Esempi

   Firmare un'e-mail con il proprio nome o inserire un codice PIN

eIDAS riconosce 3 tipi di e-signature

2. Firme Elettroniche Avanzate (AdES)

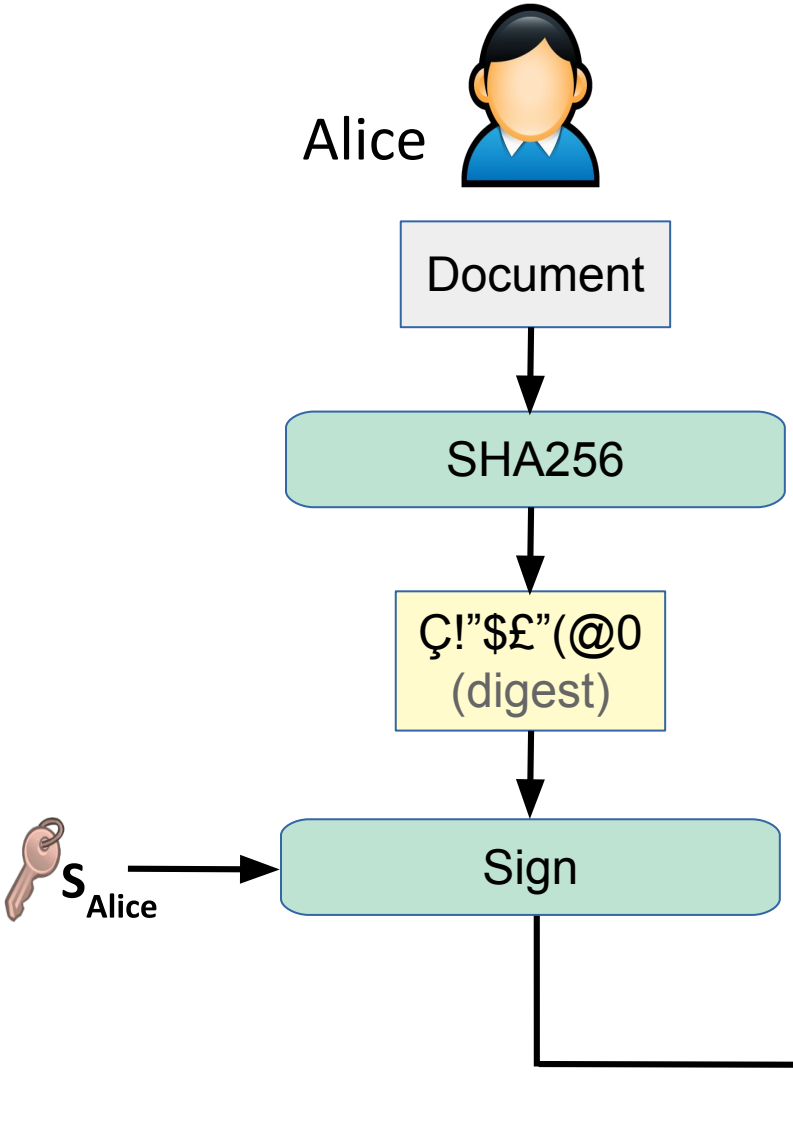
   Le firme AdES **devono corrispondere in modo univoco al firmatario e devono essere in grado di identificarlo**. I firmatari generano la loro firma esclusivamente utilizzando dati posti sotto il loro controllo, mentre il documento finale deve essere a prova di manomissione.

   Esempi

   ←**Firme Digitali**

   XAdES, PAdES, CAdES, Associated Signature Container Baseline Profile senza Certificato Qualificato, firma grafometrica, firma biometrica, ecc.

Firme Elettroniche Avanzate (AdES)



1. il documento finale deve essere a prova di manomissione
2. I firmatari generano la loro firma esclusivamente utilizzando dati posti sotto il loro controllo
3. devono corrispondere in modo univoco al firmatario e devono essere in grado di identificarlo

eIDAS AdES

- I formati che queste firme elettroniche avanzate devono possedere sono definiti nella **Decisione di esecuzione (UE) 2015/1506** (articolo 1)**:**
  - "Gli Stati membri [...] riconoscono la firma elettronica avanzata **XML, CMS, PDF"**
- *"Le firme elettroniche avanzate di cui all'articolo 1 della decisione devono rispettare una delle seguenti specifiche tecniche ETSI":*
  - **XAdES, CAdES, PAdES**

ETSI

- European Telecommunications Standards Institute
- Organizzazione non-profit responsabile della creazione e del mantenimento di questo insieme di **norme tecniche a sostegno del quadro giuridico eIDAS**.

XAdES: XML Advanced Electronic Signature

- Firme codificate in un formato testuale leggibile e conforme alle regole dell'XML (Extensible Markup Language).
- XAdES è leggibile **sia dall'uomo che dalla macchina**, il che lo rende adatto a una grande varietà di casi (immagini JPEG, file multimediali MP3, qualsiasi tipo di dati binari, documenti PDF, ecc.)
- XAdES consente **2 modalità** di firma:
  - **Detached**: produce un file XML **senza modificare il file iniziale**. I dati sono separati dalla firma, ma poi possono essere confezionati insieme.
  - **Encapsulated**: produce un file XML che include i dati. La firma poi impacchetta tutto insieme.
- **Vantaggio** -> facilita l'elaborazione automatica, supporta la firma multipla, due diversi firmatari possono firmare lo stesso documento o gruppi di documenti in parallelo o in sequenza.

CAdES: CMS Advanced Electronic Signature

- CMS -> Cryptographic Message Syntax, an IETF Standard per messaggi protetti da crittografia
- Le sue caratteristiche sono molto simili a quelle di XAdES, solo che CAdES **può essere applicato solo ai dati binari**.
- Inoltre, **manca** di alcuni concetti chiave di XAdES come la **la firma di più documenti**

PAdES: PDF Advanced Electronic Signature

- Questo formato è più limitato rispetto a XAdES -> solamente firma di file PDF
- Per impostazione predefinita, **la firma elettronica è sempre incorporata nel documento PDF firmato**, che è leggibile solo dall'uomo.
- Non è quindi adatto nel caso in cui i dati debbano essere letti anche da un computer.
- PAdES non supporta la firma parallela e **richiede un software PDF per firmare e verificare** la firma elettronica es. -> Adobe Reader.

eIDAS riconosce 3 tipi di e-signature

3. Firme Elettroniche Qualificate (QES)

   QES è una forma più rigorosa di AdES. Ha lo stesso valore legale delle firme tradizionali.

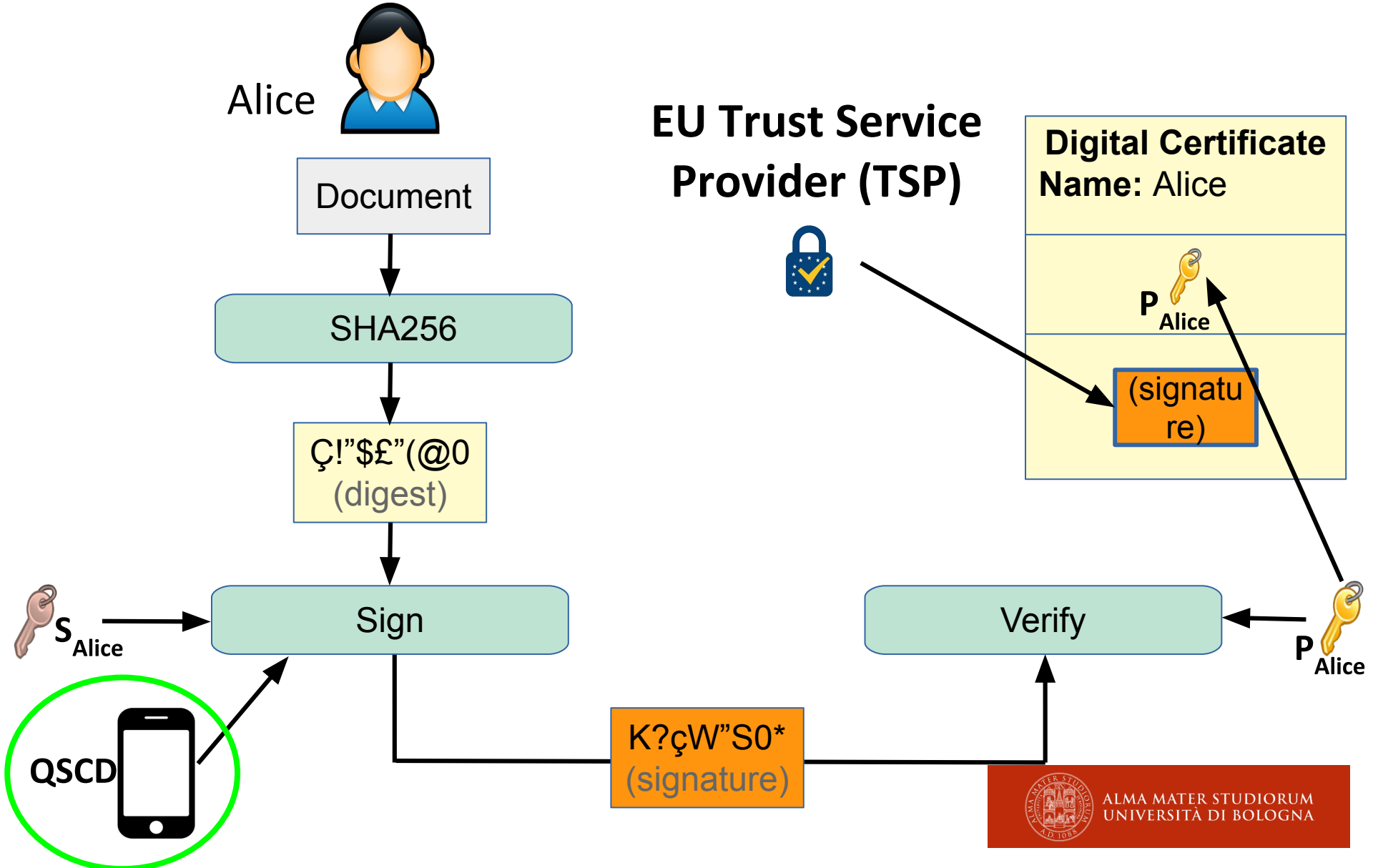
   Richiede ai firmatari di:

   a. utilizzare un **ID digitale basato su un Certificato Digitale,** rilasciato da un **EU Trust Service Provider (TSP)** qualificato
   b. utilizzare un **dispositivo per la creazione di una firma qualificata (QSCD)**.

   Esempi

   XAdES, PAdES, CAdES con Certificato Qualificato e dispositivo dicuro: smart card, USB token, o smartphone con una password one-time

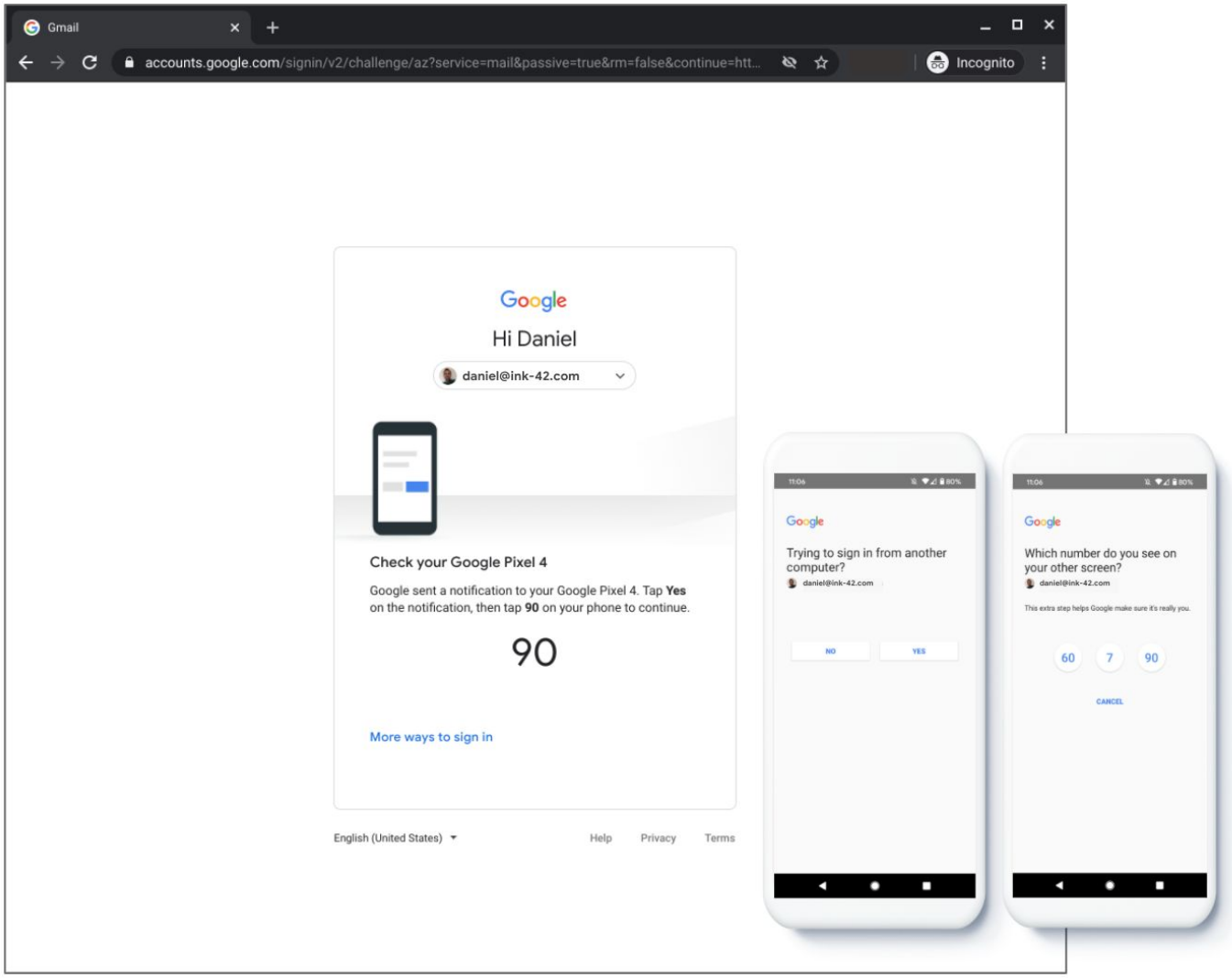
Firme Elettroniche Qualificate (QES)



Password One-time e verifica in due passaggi

- Un'autenticazione forte combina due o più di:
  - Qualcosa che **conosci**
  - Qualcosa che **hai**
  - Qualcosa che **sei** (impronta digitale)
- Una combinazione delle prime due è la più comune ed è conosciuta come **verifica in due passaggi**:
  - Conosci: Una **password personale**
  - Hai: un oggetto fisico come un "security token" della banca o **uno smartphone con una Password One-Time**

Verifica in due passaggi



Password One-time: Challenge-Response

1. Alice dichiara la sua intenzione di accedere al Servizio
2. Il Servizio seleziona una **"sfida"** e la invia ad Alice
3. Alice calcola una **"risposta"** alla sfida e la rimanda indietro
4. Il Servizio confronta la risposta ricevuta da Alice con la **risposta "attesa"** per la sfida che ha inviato
   - Se corrispondono, accesso consentito, altrimenti no

One-Time -> la **"risposta" è unica** per la sfida e può essere utilizzata una sola volta (perché la **"sfida" cambia** ogni volta)

Password One-time: Challenge-Response Crittografia Simmetrica

1. Alice dichiara la sua intenzione di accedere al Servizio con il quale condivide una chiave segreta **K**
2. La sfida del Servizio è: una **stringa random** "ciaosfida" inviata ad Alice
3. Alice calcola la risposta alla sfida cifrandola con la chiave K **risposta = C("ciaosfida", K)**
4. Il Servizio decifra la risposta, **risultato = D(risposta, K)** e confronta il risultato con **"ciaosfida"**
   - Se **risultato == "ciaosfida"**, accesso consentito, altrimenti no

Password One-time: Challenge-Response Crittografia Asimmetrica

1. Alice dichiara la sua intenzione di accedere al Servizio con il quale condivide una chiave pubblica
2. La sfida del Servizio è: una **stringa random** “ciaosfida” inviata ad Alice
3. Alice calcola la risposta alla sfida firmandola digitalmente **risposta = sign(“ciaosfida”)**
4. Il Servizio verifica la risposta, **risultato = verify(risposta)**
   - Se la firma è valida, accesso consentito, altrimenti no

Demo Firma Digitale eIDAS

https://joinup.ec.europa.eu/dss-webapp/sign-a-document

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Marcatura Temporale Fidata

Marca Temporale (timestamp)

- Sequenza di caratteri rappresentante **una data e/o un'ora** per accertare l'effettivo verificarsi di un determinato evento

2020-10-07T15:54:19+00:00

- Standard **ISO 8601** per la rappresentazione -> usato nei protocolli di rete per limitare la possibilità di errore
- Nella maggior parte dei calcolatori viene derivato tramite lo **Unix time** ->

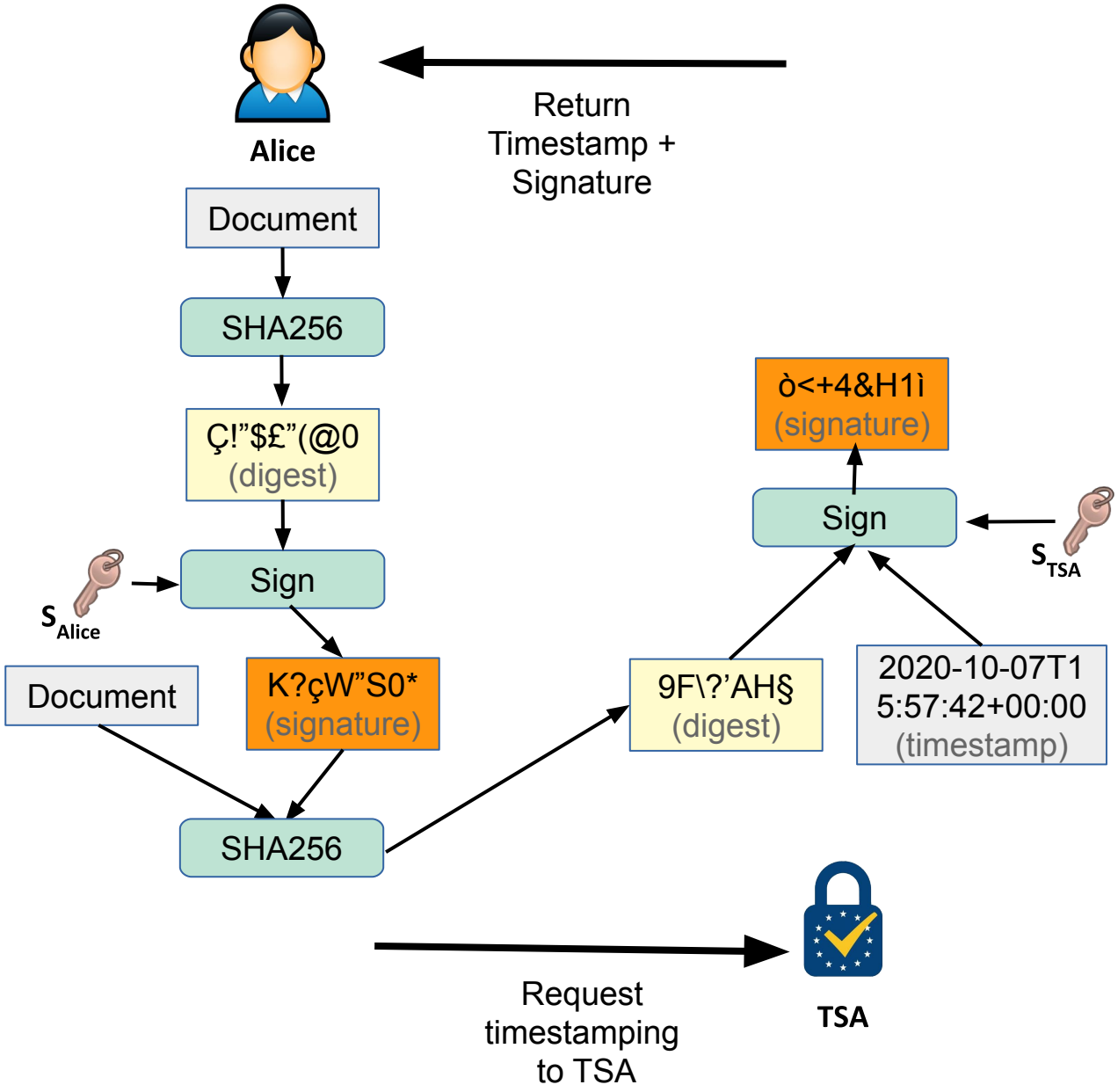
il numero di secondi passati dal 1° Gennaio 1970

1602086059

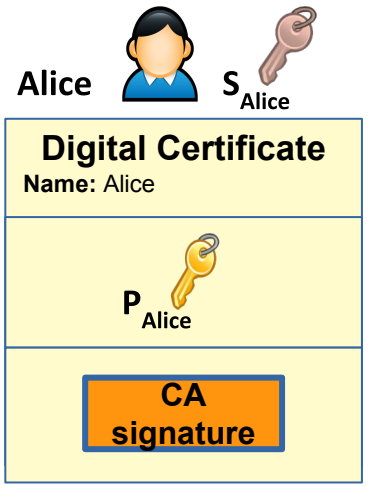
Marcatura Temporale basata su Infrastruttura a chiave pubblica

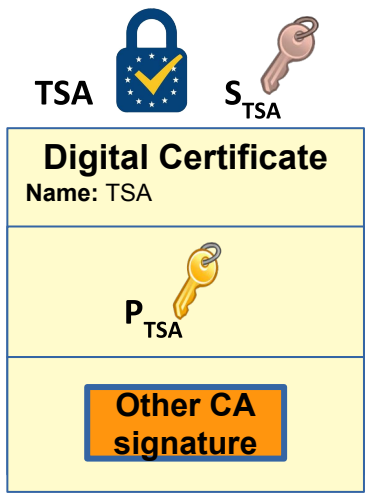
- L'apposizione della marca temporale permette di **stabilire l'esistenza ed il contenuto del documento a partire da un determinato momento**.
- Lo standard **RFC 3161** definisce il processo di marcatura temporale fidata basato su una **PKI X.509**
- Il processo di marcatura temporale consiste nell'apposizione di un timestamp su un documento digitale, da parte di un **Certificatore Accreditato** (Time Stamping Authority **TSA**), mediante firma digitale sul documento.

Marcatura Temporale

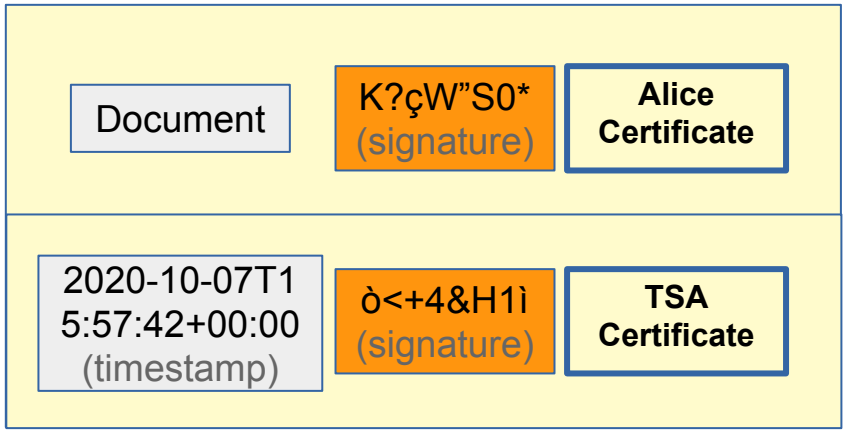


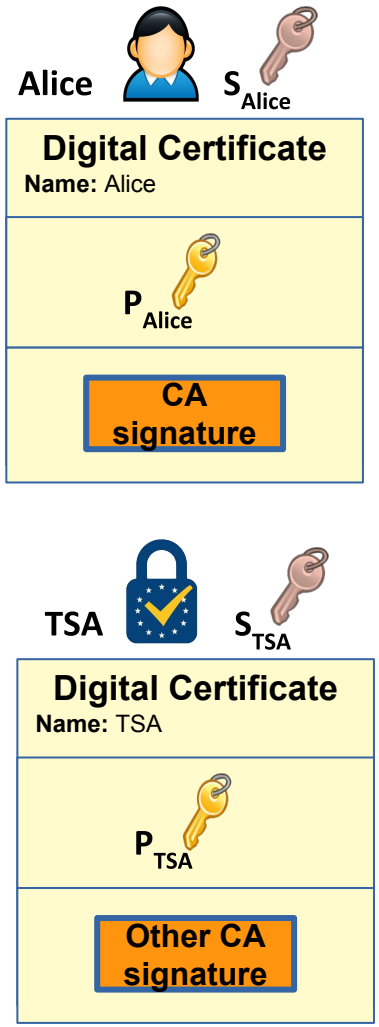
PKI





Marcatura Temporale





ANSI ASC X9.95 Standard

- L'**ANSI X9.95** è un'estensione del RFC 3161 per garantire una maggiore sicurezza sull'integrità dei dati
  - **Schemi basati sul collegamento** -> il timestamp viene generato in modo tale da essere collegato ad altri timestamp (merkle tree).
  - **Schema a chiave transitoria** -> variante PKI con chiavi di firma che hanno una "breve durata".
  - **MAC** -> schema semplice basato su una chiave segreta condivisa
  - **Database** -> gli hash dei documenti sono archiviati in un archivio fidato.
  - **Schemi ibridi**

Marcatura temporale distribuita

- Invece di un unico TSA ci si può affidare ad un **algoritmo distribuito** che guida **diverse parti** che dialogano tra di loro a raggiungere un **consenso** ->
- Con l'avvento della **blockchain** e delle tecnologie relative (**Distributed Ledger Technologies**), l'hash dei documenti digitali può essere incorporato in una transazione che viene memorizzata nella blockchain.
- In questo caso l'immutabilità della DLT prova l'ora in cui quei dati esistevano.

Marcatura temporale distribuita: Problemi

- La sicurezza di questo approccio deriva dal **meccanismo di consenso.** Es. in Bitcoin questo è il **Proof of Work** (PoW), un enorme lavoro di calcolo effettuato ogni volta che viene aggiunto un nuovo blocco alla blockchain.
- La manomissione del timestamp richiederebbe più risorse di calcolo rispetto al resto della rete combinata.
- Tuttavia, il protocollo di molte DLTs rende i suoi **timestamp vulnerabili ad un certo grado di manipolazione** -> un timestamp può essere spostato fino a due ore nel futuro e possono essere accettati prima dei dati con timestamps antecedenti.

Marcatura temporale distribuita: Problemi

- Un altro problema consiste nel caricamento dell'hash dei documenti digitali sul registro distribuito **->**

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Immutabilità delle DLTs e Dati Personali

Immutabilità

- Blockchain -> **hash pointers + PoW**
- In Bitcoin (probabilisticamente) quando un blocco viene seguito da almeno 6 blocchi può essere considerato **Valido**
- Da quel momento in poi i dati salvati su quel blocco possono essere **considerati immutabili**
- E sono **pubblici** per tutta la rete

Dati personali

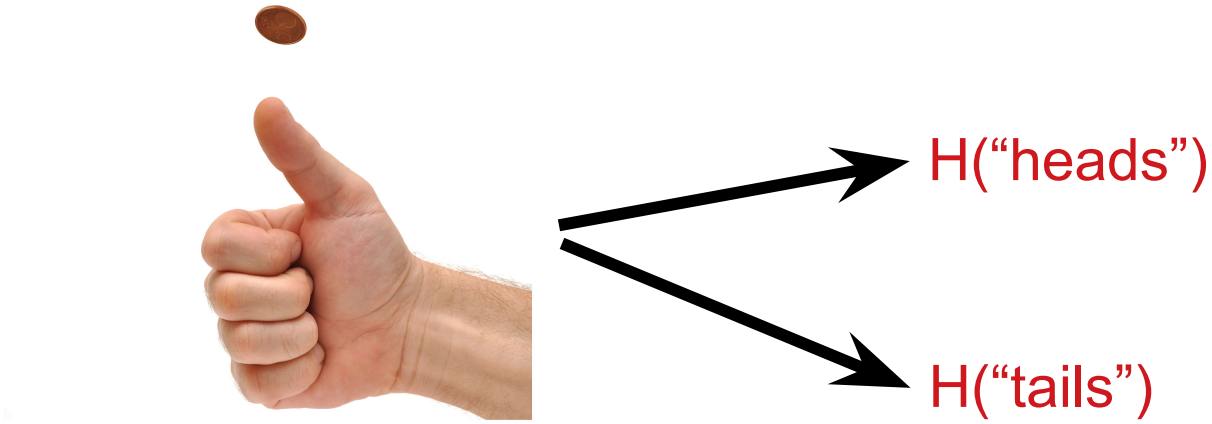
- **Compliance con GDPR** - Regulation (EU) 2016/679 (General Data Protection Regulation)
- Documenti contenenti dati personali non possono essere caricati in chiaro sulla DLT perché **pubblici**
- Inoltre gli articoli 16 e 17, richiedono la modifica o la cancellazione dei dati in determinate circostanze -> **"diritto all'oblio”**
- Soluzione **-> Caricare solamente il risultato dell’hash sulla DLT**
- Ma...

Anonimizzazione dei Dati personali

- GDPR -> dati possono essere pubblicati se **anonimizzati**
- Considerando 26 GDPR -> i dati sono anonimi se è **"ragionevolmente probabile" che non possano essere collegati a una persona fisica identificata o identificabile**

L'hash può portare ad una reidentificazione!

Example where Hiding Fails



- By looking at the hash results, it is easy to understand if x was a tails or a heads
  - Only two inputs!
  - Just hash the two inputs and looks at the hash result

Example where Hiding Fails

- Dati Personali -> **Età**

H(“1”) -> HD57cb4jiuKlo98hgf

.

.

.

H(“99”) -> i902And3òjam32sì34d

Example where Hiding Fails

- Dati Personali -> **Numero di Telefono**

  H(“+39 300 0000000”) -> 1A57cb4X2uKlo98hgf

  .

  .

  .

  H(“+39 399 9999999”) -> ò20bAnd3òjam32sazè2

1.000.000.000 di numeri possibili

SHA256 -> Windows 10 con 1 core Intel i7 2.60GHz -> **meno di 13 minuti**

Una comune GPU x60 più veloce

Example where Hiding Fails

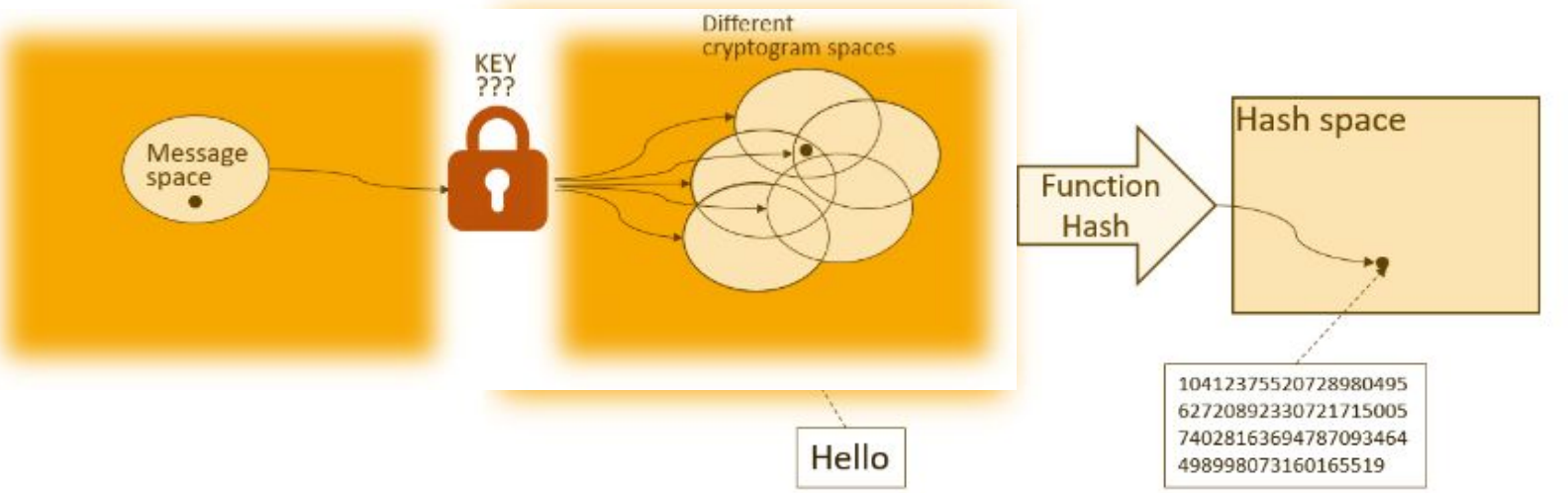
=> l'hash è ancora un dato personale

Strategie per impedire la reidentificazione

- **Agencia Española de Protección de Datos** -> Introduction to the Hash Function as a Personal Data Pseudonymisation Technique (2019)
- ‘non consentire l'identificazione del soggetto dei dati personali tramite l'impiego di "tutte" le modalità "possibili" e "ragionevoli"’
- Strategie:
  - **Pepper: Hashing con riutilizzo delle chiavi**
  - **Salt: Aggiunta di un message heading**
  - **Single-use salting**
  - **Differential Models**

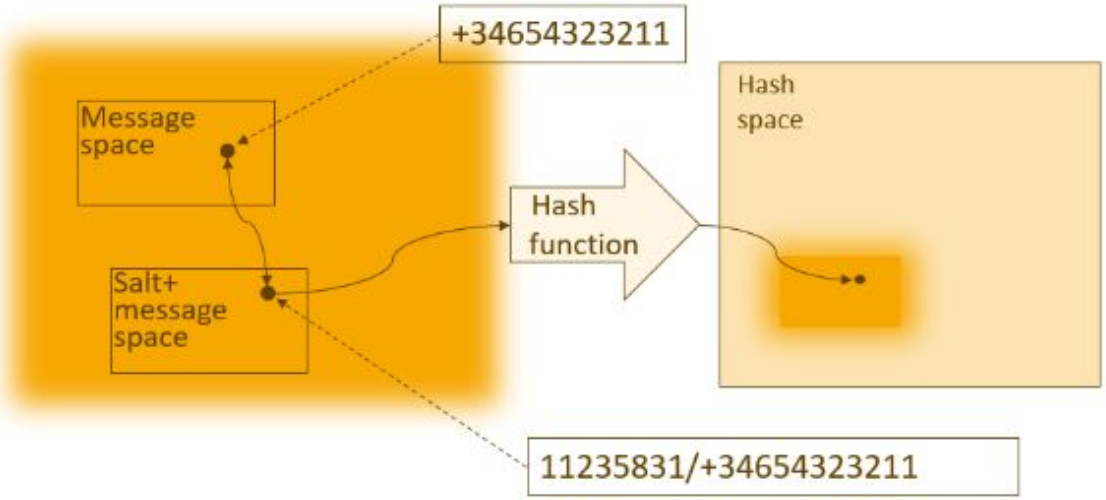
Pepper: Hashing con riutilizzo delle chiavi

- **Documento cifrato prima che l'hash sia eseguito**, tramite una chiave conservata in maniera sicura
- Risultato dell'hash **criptato dopo che è stato calcolato**



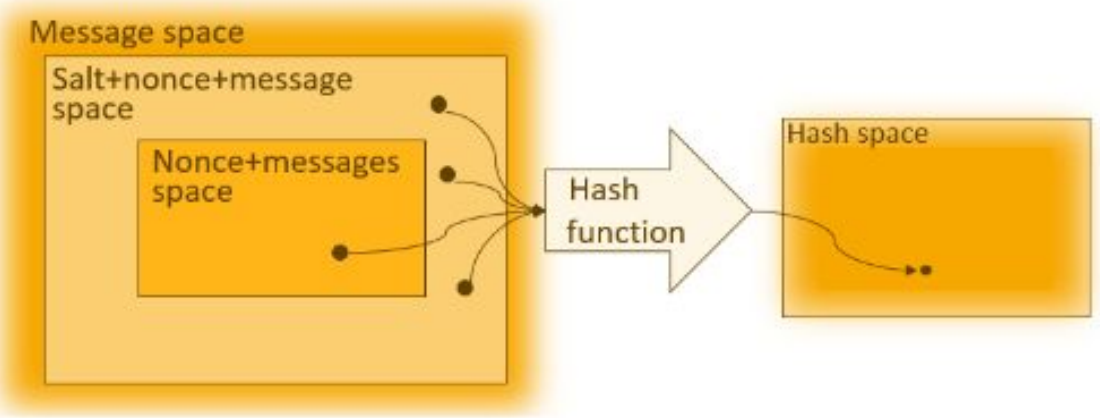
Salt: Aggiunta di un message heading

- Aggiungendo **un valore costante** a tutti i documenti prima di valutare l'hash
- “Sale” -> qualsiasi **valore casuale indipendente aggiunto** al messaggio originale
- E.s. “NumeroDiTelefono/+393495739567”



Salt: Single-use salting

- Aggiungendo **un valore variabile** a tutti i documenti prima di valutare l'hash
- Può essere usato un numero chiamato **Nonce** che varia per ogni nuovo documento (che deve essere conservato)
- E.s. “NumeroDiTelefono/**322**/+393495739567”



Differential Models

- Aggiungendo del **“rumore”** al documento, in maniera che il significato originale non venga perso, ma che il documento finale sia diverso da quello originale

Original

Noisy image

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Identità decentralizzate

Chiavi come Identità

Mirko Zichichi

associato a

X1457cb4jiuKlo98hgf

Indirizzo Pubblico

(alphanumeric)

Come creare una nuova Identità

- Creare una nuova coppia di chiavi asimmetriche (***sk***, ***pk***)

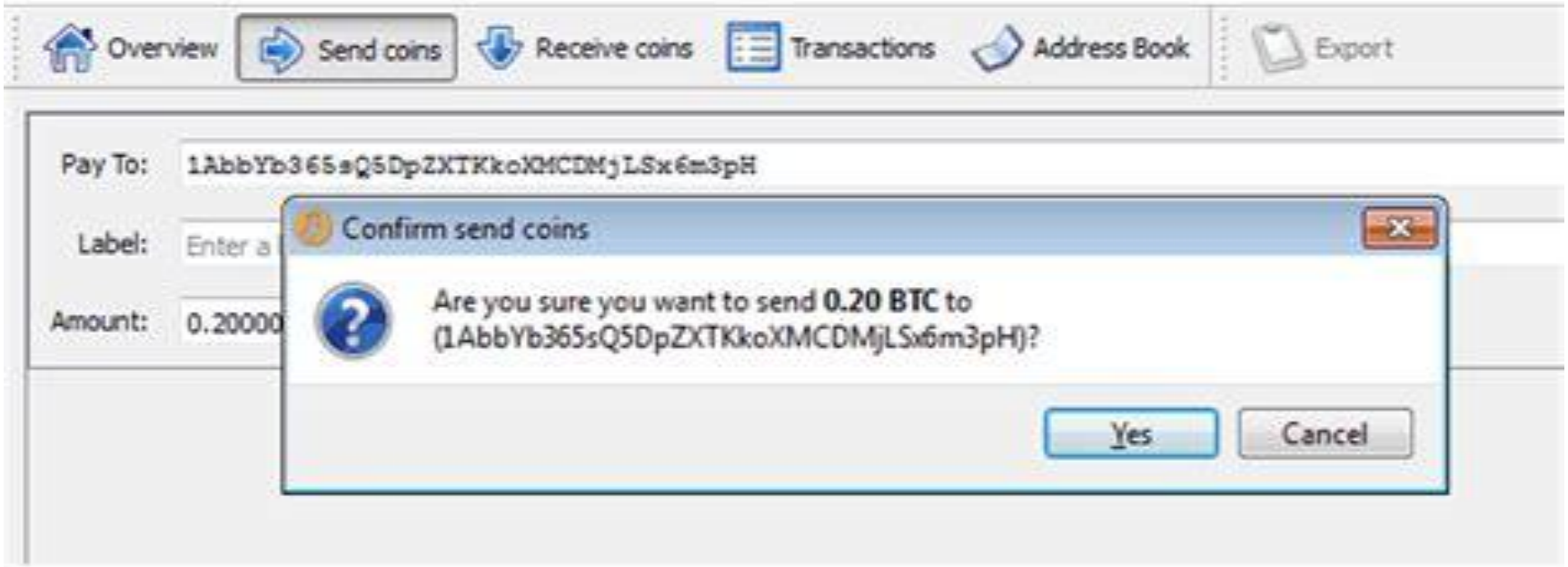
  - La chiave pubblica ***pk*** è il "nome" dell'identità
    - Pseudonimo
    - ***pk*** spesso chiamata **indirizzo** (address)
    - Meglio usare **Hash**(***pk***) come indirizzo
  - La chiave privata ***sk*** permette di "parlare a nome" dell'identità

- Ognuno può controllare la propria identità perché solo lui conosce ***sk***
- Se ***pk*** "sembra random", nessuno sa associarla ad un individuo

Gestione decentralizzata delle identità

- Chiunque può **creare una nuova identità in qualsiasi momento** e farne quante ne vuole!
- La probabilità di generare la stessa chiave di un altro utente è trascurabile
- **Nessun punto centrale di coordinamento**
- Nessuna autorità centrale che registri le identità nel sistema
- Queste "identità decentralizzate" sono chiamate **"indirizzi" in Bitcoin**

Gestione decentralizzata delle identità



Identificatori Decentralizzati (DID)

- Un tipo di identificativi che consentono un'**identità digitale verificabile e decentralizzata**.
- Essi si basano sul paradigma dell'**identità auto-sovrana** (self-sovereign identity).
- Un DID identifica **qualsiasi soggetto** (ad es. una persona, un'organizzazione, una cosa, un modello di dati, ecc.)
- Questi identificatori sono progettati per consentire al controllore di un DID di dimostrare il controllo su di esso e per essere implementati indipendentemente da qualsiasi registro centralizzato, fornitore di identità o autorità di certificazione.

Identificatori Decentralizzati (DID)

The standard elements of a DID doc

1. **DID** (for self-description)
2. **Set of public keys** (for verification)
3. **Set of auth methods** (for authentication)
4. **Set of service endpoints** (for interaction)
5. **Timestamp** (for audit history)
6. **Signature** (for integrity)

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Stefano Ferretti, Mirko Zichichi

s.ferretti@unibo.it
mirko.zichichi2@unibo.it

www.cs.unibo.it/sferrett