UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

"Sanità e Self Sovereign Identity"

Relatore: Prof. Mauro Migliardi

Laureando: Spaziani Alberto

ANNO ACCADEMICO 2023 – 2024

Data di laurea 19.03.2024

Indice

Abstract

Con il progredire della digitalizzazione e la crescente diffusione di internet, l'identità digitale del singolo è stata sempre più frammentata nel mondo del web, i dati sensibili sono stati largamente condivisi e il controllo su di essi è diventato sempre più fragile e precario. L'utente è ora vittima di hackers e altri utenti che mirano quotidianamente ai dati preziosi con fini malevoli e il fatto che questi siano ormai sparsi per il web rende questo furto di dati più semplice. Recentemente anche gli annunci pubblicitari sono diventati monito di un'ulteriore perdita di controllo sui dati condivisi in rete, proponendo annunci personalizzati per mezzo di dati ottenuti ad esempio da data providers come google o i social networks. È così che nel corso degli anni è nata e si sta sviluppando sempre più la necessità di ridare all'utente il controllo che sin dalla nascita di internet gli era stato sottratto, motivo per il quale è stato introdotto un nuovo modello user centrico di gestione dell'identità digitale, il Self Sovereign Identity il cui scopo è quello di decentralizzare i dati e di fornire al singolo utente la capacità di condividere solo i dati che egli riterrà necessario al fine di garantirgli un maggior controllo sulla propria identità digitale e di ridurre così anche il rischio di essere vittima dei data leaks a cui possono essere soggetti i database sia fisici che non dei data providers o di altri siti terzi ai quali gli utenti si affidano per poter usufruire di contenuti che richiedono una registrazione presso di essi.

Questa tesi si propone come obiettivo quello di approfondire il concetto di Self Sovereign Identity, di parlare dei pilastri alla base del funzionamento di tale paradigma e di applicare tali concetti in un caso d'uso nel campo medico, in particolare nella gestione dei dati sanitari facenti parte della storia clinica dell'utente e nella loro condivisione controllata. Ci si concentrerà su alcuni aspetti quali la conservazione dei dati in un database criptato, la sicurezza di essi trattando l'aspetto dell'autenticazione e i componenti della Self Sovereign Sdentity, spiegando per ognuno di essi il proprio funzionamento e il loro ruolo all'interno del sistema affinchè sia GDPR compliant. Infine, si faranno delle considerazioni sui pro e i contro dell'impiego di tale paradigma in questo usecase ponendo attenzione alla blockchain e ad alcune caratteristiche quali la scalabilità, i costi e l'impatto ecologico della soluzione descritta in questo elaborato.

1. Introduzione

Alla creazione di internet, l'importanza della privacy dei dati digitali appartenenti agli utenti che ne usufruivano non era uno degli aspetti principali sui quali ci si era concentrati. Non vi erano regolamentazioni e non si valorizzava l'importanza della preservazione dei dati personali in ambito digitale come si faceva invece con i dati relativi all'identità fisica. La navigazione in rete era ancora un aspetto marginale e non conosciuto a molti e non ricopriva minimamente il ruolo importante che ha oggi giorno.

Inizialmente era nato dalla collaborazione tra il Dipartimento della Difesa degli Stati Uniti e diverse istituzioni accademiche, tra cui il CERN, con lo scopo di facilitare la condivisione di documenti elettronici, pertanto, non essendo stato creato agli albori con lo scopo di essere adatto all'uso civile non si pensò all'introduzione e alla regolamentazione di un cosiddetto "identity layer" trascurando un concetto che sarebbe diventato cruciale nel mondo profondamente interconnesso nel quale ci troviamo oggi.

Kim Cameron, Chief Architect di Microsoft, in The Laws of Identity (May 2005, p.1) afferma che "The Internet was built without a way to know who and what you are connecting to. This limits what we can do with it and exposes us to growing dangers. If we do nothing, we will face rapidly proliferating episodes of theft and deception which will cumulatively erode public trust in the Internet."

Negli ultimi decenni l'esigenza di preservare i dati personali degli utenti del web è diventata sempre più marcata a causa del numero sempre crescente di crimini informatici, furti d'identità e data leaks. Questi ultimi spesso causati da misure di sicurezza insufficienti e non aggiornate. Nell'ultimo decennio i reati informatici sono cresciuti di circa il 10% annuo. Nel dettaglio, tra il 2015 e il 2020 le truffe e frodi informatiche sono salite del 72,8%, mentre sono quasi raddoppiate (+96,3%) le denunce di crimini informatici. Un altro problema riguardante la tutela della privacy coinvolge i provider di servizi che vendevano i dati dei propri utenti ai broker pubblicitari, i quali provvedevano a fornire pubblicità personalizzate evidenziando nuovamente la perdita di controllo da parte degli utenti dei propri dati personali.

In questa tesi, verrà presentato e progettato un sistema basato sul modello Self Sovereign Identity (SSI) in ambito medico. Il sistema mira a conferire agli utenti il controllo completo delle proprie informazioni sanitarie, garantendo loro la facoltà di condividerle solo con soggetti specifici e in misura strettamenta necessaria. Il paradigma SSI in passato era già stato utilizzato per la realizzazione del progetto europeo EBSI (European Blockchain Services Infrastructure), che mirava a creare

un'infrastruttura basata sulla blockchain interoperabile per l'Europa e ha dimostrato la fattibilità di utilizzare la blockchain per creare un'infrastruttura di identità digitale sicura e interoperabile.

Nel secondo capitolo della tesi si effettuerà una panoramica sul concetto della Self Sovereign Identity in particolare si andrà ad analizzare brevemente la storia della SSI, il motivo dell'introduzione di tale nuovo paradigma e ci si concentrerà sull'esposizione dei principi cardine al suo funzionamento quali Identificatori Decentralizzati (DID), Credenziali Verificabili (VC) e Blockchain. Si analizzeranno i primi due componenti della Self Sovereign Identity (quindi DID e VC) su aspetti come le parti delle quali sono costituiti, il loro funzionamento, i punti di forza che hanno portato alla loro introduzione in questo sistema e degli esempi di implementazione. Per quanto riguarda la Blockchain verranno discussi aspetti quali la storia che ha portato alla nascita di tale struttura, le parti che la compongono, le diverse tipologie esistenti, il loro funzionamento e alcuni attacchi informatici che sono particolarmente efficaci. Si discuterà quindi dei pro e contro dell'impiego della Blockchain e del suo utilizzo ad oggi. Verrà infine introdotto e analizzato il regolamento GDPR.

Nel terzo capitolo si proporrà una visione di insieme del caso d'uso scelto in ambito medico e del funzionamento del sistema proposto in questa tesi mostrando il ruolo ricoperto dai componenti della SSI e il loro funzionamento, si discuterà di come il sistema sia compliant al regolamento europeo per la protezione dei dati personali (GDPR).

Nel quarto capitolo si affronteranno le tematiche relative agli aspetti fondamentali del sistema quali la conservazione dei dati, la crittografia delle comunicazioni e l'autenticazione. Verranno inoltre proposte e schematizzate delle query rappresentanti le comunicazioni più frequentemente utilizzate.

Nel quinto capitolo si tratterà della scalabilità del sistema e si faranno alcune considerazioni sulla sicurezza di questo e sull'impatto ambientale ed energetico della soluzione proposta.

Infine, nel sesto capitolo si trarranno delle conclusioni sul modello rappresentato dalla self sovereign identity.

2. Self Sovereign Identity (SSI)

2.1 Spiegazione, Storia e Confronto con Sistemi precedenti

Il concetto che sta alla base della SSI è il pieno controllo da parte della persona della propria identità e delle informazioni personali cosicchè possa decidere a chi fornire informazioni nella misura strettamente necessaria. Prima dell'introduzione della Self Sovereign Identity, l'identità digitale del singolo veniva frammentata nel web a causa del fatto che inizialmente ogni singolo sito o piattaforma richiedeva, per poter andare ad usufruire delle proprie funzionalità, la creazione di un account e questo

faceva sì che ad una persona fossero associate diverse identità digitali (modello centralizzato o modello silos). In questo sistema l'organizzazione crea l'identità dell'utente per il proprio servizio e questa, pertanto, rimane il punto centrale del modello. Tutti i dati personali dell'utente vengono detenuti all'interno dei database dell'organizzazione, anche chiamati "Silos" (da cui il nome di "Modello a silos").
Il modello centralizzato rende gli utenti completamente "dipendenti" dalle organizzazioni che detengono i propri dati, e porta con sé alcuni potenziali rischi e debolezze, tra cui possiamo citare i leaks del database dell'organizzazione che esporrebbero i dati sensibili degli utenti. I sistemi di identità digitale centralizzati hanno creato quello che viene definito come "fenomeno delle identità multiple". Gli utenti, ad oggi, con questo sistema si trovano ad avere un'identità digitale diversa per ogni servizio da loro utilizzato

Successivamente a questo sistema primordiale ci fu un passo avanti con la possibilità di registrarsi su più siti con la stessa identità virtuale, per esempio, attraverso il proprio account di Facebook o anche attraverso il proprio account di Gmail (modello federato) diminuendo così la frammentazione del singolo e aumentando il controllo che un utente poteva avere sui propri dati. Il problema del precedente sistema era che si perdeva il controllo dei propri dati con il progressivo aumento del numero di account posseduti, questo problema è stato parzialmente sistemato dal modello federato che garantiva una maggiore centralità dei dati. Il controllo di tali dati però era ancora affidato a terzi e non al singolo utente e pertanto molte società potevano decidere di vendere questi dati o di fornire delle pubblicità personalizzate in base all'attività online dell'utente andando così a minare la privacy di quest'ultimo. Nel caso del modello federato, l'entità che "detiene" effettivamente l'identità digitale dell'utente ed i dati ad essa associati è l'identity provider (IDP). L'utente è in grado di utilizzare la propria identità digitale con i vari servizi, sempre "passando" per l'IDP, che rimane al centro del modello.
Va anche detto che questi due sistemi rispetto alla SSI sono anche più vulnerabili a data leaks causati magari da attacchi informatici che hanno l'intento di rivendere poi i suddetti dati sul mercato nero o di utilizzarli per scopi malevoli.
Nella Self Sovereign Identity la sicurezza è maggiore perché l'utente è in possesso di tutte le sue credenziali e non deve richiederle in diverse occasioni ai vari identity providers.

ll concetto di SSI è completamente "centrico" verso l'utente, che è l'unico ed indipendente possessore della propria identità digitale e di tutti i dati ad essa associati grazie all'utilizzo di protocolli come quelli Blockchain. Grazie alle caratteristiche essenziali di una blockchain, quali immutabilità (vedremo in maniera più dettagliata cosa si intende per questo nel capitolo 2.4) e resilienza (no down time). Il modello Self-Sovereign Identity vanta una maggiore sicurezza verso tutti gli attori coinvolti attraverso l'utilizzo di blockchain, credenziali digitali con firme elettroniche e la crittografia.

La SSI può essere considerata sia una ideologia, sia una architettura tecnologica. Per quanto riguarda l'ideologia, si intende la volontà di ritornare in possesso della propria identità e rivendicare il controllo su di essa basandosi su quanto consegue dalla Dichiarazione Universale dei Diritti Umani che afferma che ogni persona ha il diritto alla privacy e alla protezione dei dati personali. Per quanto riguarda l'architettura, la SSI è una tecnologia, composta da diversi elementi, che permette ed abilita gli individui a soddisfare le condizioni poste nell'ideologia.

Uno dei primi riferimenti al concetto di "sovranità relativa alla propria identità digitale" si può ritrovare nello scritto dello sviluppatore Moxie Marlinspike "Sovereign Source Authority" del 2012 nel quale egli affermava: "gli individui hanno il diritto consolidato della propria identità, ma l'anagrafe ha distrutto la possibilità di avere il controllo su di essa". Quasi contemporaneamente nel web si è assistito alla diffusione di iniziative e proposte simili. Nel marzo 2012, Patrick Deegan iniziò a lavorare su Open Mustard Seed, un framework open source che offriva agli utenti il controllo della propria identità digitale grazie un sistema decentralizzato. Deegan intendeva affrontare il tema dell'identità sovrana attraverso la crittografia e strumenti matematici per proteggere l'autonomia dell'utente.
Open Mustard Seed non fu l'unico esperimento relativo alla SSI. Everynym Essentials, scritto da Samuel M. Smith Ph.D. and Dmitry Khovratovich Ph.D., fu anch'esso fondamentale alla realizzazione di un'identità digitale sovrana. Infine, dal 2016, Il *World Wide Web Consortium*, anche conosciuto come W3C, ovvero l'organizzazione non governativa internazionale che ha come scopo quello di sviluppare tutte le potenzialità del World Wide Web, iniziò a creare dei working group per sviluppare dei framework aperti che permettessero la standardizzazione di questa nuova infrastruttura digitale.

La Self Sovereign Identity è un concetto che viene abilitato da una serie di innovazioni, quali gli standard degli Decentralized Identifiers (DID), delle Verifiable Credentials (VC) e la Blockchain.

Figura 1: Rappresentazione del paradigma SSI con i pilastri alla base del suo funzionamento

2.2 Decentralized Identifiers (DID)

2.2.1 Introduzione

Uno dei pilastri del modello SSI è rappresentato dai Decentralized Identifiers (DIDs). Un DID è un elemento costitutivo di un nuovo livello di identità decentralizzata con quattro proprietà quali: persistenza, risolvibilità, crittografia, decentralizzazione. È un nuovo tipo di identificatore, più semplicemente consiste in una stringa alfanumerica che identifica una risorsa, più tecnicamente è una tipologia di URI.

2.2.2 URI, URL e URN

Un Uniform Resource Identifier (URI) è una sequenza di caratteri che identifica in modo univoco una risorsa digitale o fisica. In quest'ultimo caso un esempio potrebbe essere un indirizzo IP o un nome di dominio di un sito Web. Gli URI identificano la posizione di un sito Web su Internet e tale sito può essere considerato una risorsa fisica, poiché è ospitato su un server fisico.

Le stringhe di caratteri incorporate in un URI fungono da identificatori come, ad esempio, i valori dei campi relativi allo scheme o al file path.

Gli URI possono identificare diversi tipi di risorse come ad esempio documenti elettronici, pagine web, immagini e altro ancora.

Gli uniform resource locators (URLs) e gli uniform resourse names (URNs) sono due tipi di URI. Un URL è usato per identificare e localizzare pagine web, infatti a differenza di un URI, che identifica una risorsa non garantendo l'accesso ad essa, un URL non solo definisce una risorsa ma specifica anche come essa posso essere acceduta o dove sia localizzata.

Ad esempio, se la risorsa è una pagina web, l'URL inizia col protocollo http o https. Un URN invece è un codice univoco e immutabile che identifica una risorsa tramite il suo nome senza dare ulteriori informazioni sulla locazione di essa in rete. A livello pratico un URN ha lo stesso principio di funzionamento che il codice ISBN ha per un libro, lo identifica univocamente.

Figura 2: Esempi di URI, URL e URN

2.2.3 DID nel dettaglio e alcuni esempi di implementazione

I Decentralized Identifiers consentono agli individui e alle organizzazioni di generare i propri identificatori usando sistemi di cui si fidano. Questi permettono alle entità di dimostrare il controllo su tali identificatori autenticandosi attraverso delle prove crittografiche come delle firme digitali. Ogni entità può avere tanti DID quanti ne sono necessari per mantenere la separazione di identità, persone, interazioni.

Un generico DID è formato da tre componenti:

- DID URI SCHEME IDENTIFIER
- DID METHOD IDENTIFIER
- DID METHOD-SPECIFIC IDENTIFIER

Figura 3: Un semplice esempio di identificatore decentralizzato
Fonte: https://www.w3.org/TR/did-core/diagrams/parts-of-a-did.svg

Una particolarità dei Decentralized Identifiers è che ad ognuno di essi sono associate una chiave pubblica e una chiave privata e il possessore della chiave privata può dimostrare di essere il controllore del DID.

La stringa alfanumerica mostrata nella figura precedente mostra un esempio di DID URL. Quest'ultimo estende la sintassi di un DID base per incorporare altri componenti URI come path, query, e fragment per localizzare una particolare risorsa come, ad esempio, una chiave crittografica pubblica in un DID document.

I DID documents contengono informazioni associate a un DID. Tipicamente esprimono metodi di verifica come le chiavi crittografiche pubbliche e i servizi rilevanti alle interazioni con i DID subjects o con i DID delegates. Un DID document consiste di un'associazione di chiavi e valori. Le proprietà presenti in un DID document possono essere modificate. Tutte le chiavi nel DID document sono stringhe. Tutti i valori sono espressi usando uno dei tipi di dati astratti e ogni rappresentazione specifica il formato di serializzazione di ogni data type (la serializzazione è il processo di conversione dei dati in un formato che può essere memorizzato su disco o trasmesso su una rete). Le diverse rappresentazioni sono: map, list, set, string, datetime, integer, double, boolean, null.

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ]
  "id": "did:example:123456789abcdefghi",
  "authentication": [{
    // used to authenticate as did:...fghi
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "Ed25519VerificationKey2020",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyMultibase": "zH3C2AVvLMv6gmMNam3uVAjZpfkcJCwDwnZn6z3wXmqPV"
  }]
}
```

Figura 4: Esempio di un DID document base
Fonte: https://www.w3.org/TR/did-core/#example-a-simple-did-document

Un DID subject è per definizione l'entità identificata da un DID. Il DID subject può anche ricoprire il ruolo del DID controller.

```
{
  "id": "did:example:123456789abcdefghijk"
}
```

Figura 5: Implementazione di un DID Subject base
Fonte: https://www.w3.org/TR/did-core/#example-10

Un DID controller è l'entità che ha la capacità di applicare cambiamenti in un DID document. Questa capacità è tipicamente accertata dal possesso di un insieme di chiavi crittografiche. Un DID può avere più DID Controllers.

```
{
  "@context": "https://www.w3.org/ns/did/v1",
  "id": "did:example:123456789abcdefghi",
  "controller": "did:example:bcehfew7h32f32h7af3",
}
```

Figura 6: Implementazione di un DID Controller
Fonte: https://www.w3.org/TR/did-core/#example-did-document-with-a-controller-property

Un DID delegate è un'entità alla quale un DID Controller ha garantito i permessi di usare un metodo di verifica associato a un DID attraverso un DID Document. Ad esempio, un genitore che controlla un DID Document di un figlio potrebbe permettergli di usare il proprio (del figlio) device personale per autenticarsi. In questo caso il figlio è il DID Delegate.

Per essere risolvibili ai DID documents, I DID sono registrati su un sistema sottostante o una rete di qualche tipo. Ogni sistema che è in grado di registrare i DID e di produrre i DID documents si chiama Verifiable Data Registry.

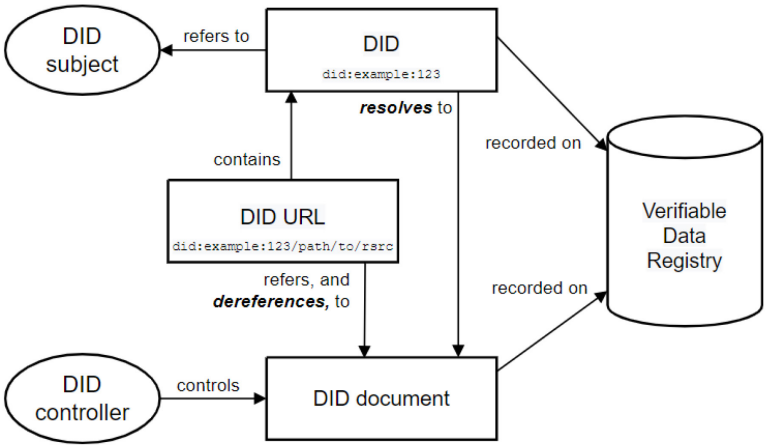

Figura 7: Visione di insieme dell'architettura DID e relazioni tra i suoi componenti base
Fonte: https://www.w3.org/TR/did-core/diagrams/did_brief_architecture_overview.svg

Infine, queste chiavi private e pubbliche utilizzano la blockchain come libro mastro o fonte di verità condivisa per dimostrare a terzi del proprio controllo, senza appoggiarsi in registri centralizzati.
Come già detto in precedenza un generico DID è formato da tre parti:

- Il method rappresenterebbe lo scheme e quindi la parte "did:"
- il DID method identifica su quale registro decentralizzato o blockchain tale DID è stato risolto
- il method specific-identifier è l'estensione che caratterizza il DID per la specifica risorsa.

Ogni DID viene registrato in modo autonomo, direttamente dal proprietario all'interno di blockchain permissionless e/o permissioned. Non ci sono intermediari, ed è per questo che viene definita Self-Sovereign Identity. Ogni utente è completamente sovrano di sé stesso e della propria identità.

Quando genero un Decentralized Identifier esso ha:

- una chiave privata che viene usata per generare la chiave pubblica di un DID
- una chiave pubblica che viene usata per verificare l'autenticità del DID (essa è diversa in ogni DID)
- informazioni aggiuntive utili per la particolare implementazione

Un DID Resolver è un componente di sistema che prende in input un DID e produce un DID Document come output. Questo processo si chiama DID Resolution. Le fasi per risolvere uno specifico tipo di DID sono definite dal DID Method.

Un DID Method rappresenta come uno specifico DID method scheme è implementato. Quest'ultimo è definito da un DID method e specifica le precise operazioni attraverso le quali i DID e i DID Document sono creati, risolti, aggiornati e disattivati.

2.3 Verifiable Credentials (VC)

Le veriable credentials rappresentano il secondo pilastro che è dietro il funzionamento del pradigma SSI. Sono un modo di rappresentare gli attributi che vengono associati alla propria identità. Gli attributi possono essere di qualunque tipo come, ad esempio, un certificato di salute che certifica il fatto che una persona abbia o meno una patologia, o qualunque documento rilasciato da un ente certificato.

Figura 8: Attributi nelle varie realtà

Le verifiable credentials hanno delle particolari caratteristiche che le rendono parte importante del sistema Self Sovereign Identity quali:

- sono sicure
- non modificabili
- verificabili in maniera indipendente

2.3.1 Flusso di Dati e Figure importanti delle Verifiable Credentials

Nel ciclo di funzionamento delle verifiable credentials vi sono tre figure importanti:

- L'issuer
- L'holder
- Il verifier

Un Issuer di credenziali può essere una qualunque entità autorizzata, come ad esempio un ministero. Le credenziali emesse possono essere di qualunque tipo, revocabili e non e offrono diversi livelli di sicurezza a seconda di chi le ha emesse.

Un Holder riceve delle credenziali firmate digitalmente da uno o più Issuer. Una volta che la credenziale viene inviata all'Holder, quest'ultimo è in grado di gestire tale credenziale in maniera del tutto autonoma, può quindi presentarla a coloro che la richiedono.

Un Verifier è qualunque entità che si occupi di verificare delle credenziali che gli sono state mostrare da un utente (Holder). La credenziale contiene tutti i dati necessari alla verifica della stessa, come ad esempio chi è l'Issuer, a chi è intestata e se sia stata modificata.
La tecnologia blockchain viene utilizzata per controllare la validità della credenziale.

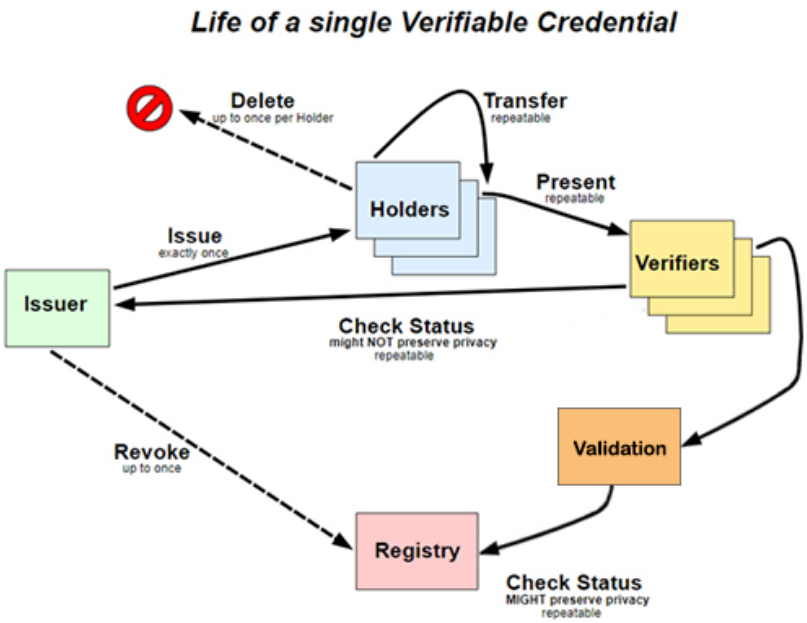

Figura 9: Flusso di lavoro di una Verifiable Credential
Fonte: https://www.w3.org/TR/vc-data-model-2.0/#life-cycle-details (modificata)

2.3.2 Caratteristiche, Componenti chiave ed esempi di implementazione

Le vefiable credentials sono anche privacy preserving, hanno come proprietà la selective disclosure, è possibile quindi per un Holder mostrare solamente i dati strettamente necessari al Verifier. Una credenziale digitale può rappresentare tutte le informazioni associate al suo corrispettivo fisico come, ad esempio, le informazioni per il riconoscimento del soggetto come foto, nome, codice fiscale o anche nazionalità, ente di emissione del documento, data di validità, ecc…

È possibile avere una credenziale come un certificato di matrimonio che contiene diverse claims appartenenti a diversi soggetti e non è richiesto che siano correlati.

È possibile anche avere una credenziale che non contiene nessuna claims sull'entità per la quale è stata emessa. Ad esempio, una credenziale che contiene solo claims su un animale domestico ma emessa al suo padrone.

Una credenziale verificabile aggiunge come ulteriore sicurezza la firma digitale che rende questo tipo di credenziali più resistente alla contraffazione.

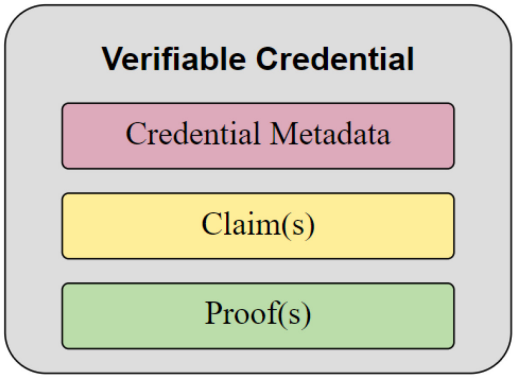

Figura 10: Componenti base di una Verifiable Credential
Fonte: https://www.w3.org/TR/vc-data-model-2.0/diagrams/vc.svg

Il titolare di una credenziale verificabile può generare delle "verifiable presentations" e condividere queste con un verificatore per provare che posseggono delle credenziali con delle determinate caratteristiche.

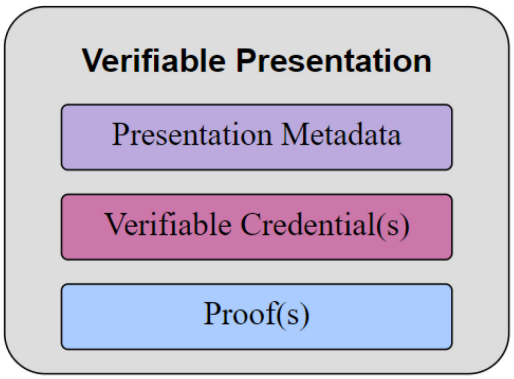

Figura 11: Componenti base di una Verifiable Presentation
Fonte: https://www.w3.org/TR/vc-data-model-2.0/diagrams/presentation.svg

Una presentation è l'insieme di dati provenienti da una o più credenziali verificabili che viene condiviso con uno specifico verificatore.

È la rappresentazione di un sottoinsieme della propria personalità come, ad esempio, la parte riguardante la vita professionale o la vita privata. Una verifiable presentation può rappresentare dati provenienti da più credenziali verificabili e contenere dati aggiuntivi codificati come JSON-LD. Le verifiable presentations sono utilizzate da un holder per presentare claims a un verifier.

Col termine "verificabile" si intende che una credenziale può essere verificata da un verificatore. La verificabilità di una credenziale non asserisce tuttavia la veridicità delle asserzioni (claims) in essa contenute; pertanto, prima di basarsi su di esse bisognerà verificarne l'autenticità.

Una claim in una verifiable credential è un'asserzione effettuata dal soggetto a cui appartiene tale credenziale. Un insieme composto da una o più claims emesse da un issuer viene chiamato credential, va notato inoltre che una credential può essere il risultato di claims provenienti da soggetti diversi.

Un soggetto è un qualcosa sul quale vengono fatte delle claims.

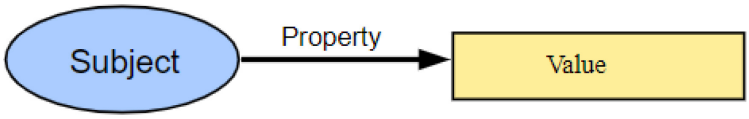

Figura 12: Componenti base di una claim
Fonte: https://www.w3.org/TR/vc-data-model-2.0/diagrams/claim.svg

Uno dei principi cardine delle verifiable credentials è la minimizzazione dei dati cioè, il limitare la condivisione di essi allo stretto necessario per performare un'azione.

Spesso le credenziali verificabili possono essere rappresentate più fedelmente attraverso l'uso di uno o più grafi interconnessi.

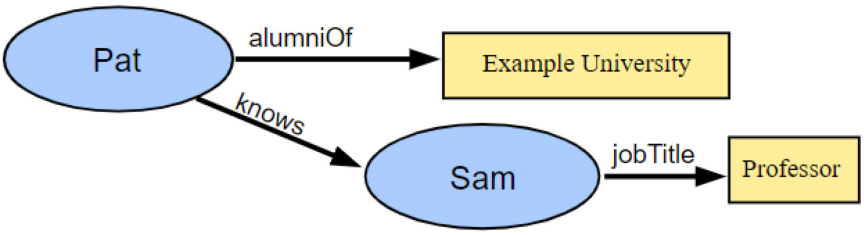

Figura 13: Più claims possono essere combinate per comporre un grafo delle informazioni
Fonte: https://www.w3.org/TR/vc-data-model-2.0/diagrams/claim-extended.svg

Un grafo è un insieme di claims che formano una rete di informazioni composta da dei subjects e dalle loro relazioni con altri subjects o con dati. Ogni claim è parte di un grafo.

default graph

è un grafo contenente tutte le claims che non sono esplicitamente parte di un named graph.

named graph

è un grafo associato con specifiche proprietà come le credenziali verificabili o le proofs. Queste proprietà figurano in grafi separati che contengono tutte le claims definite nei corrispondenti JSON objects.

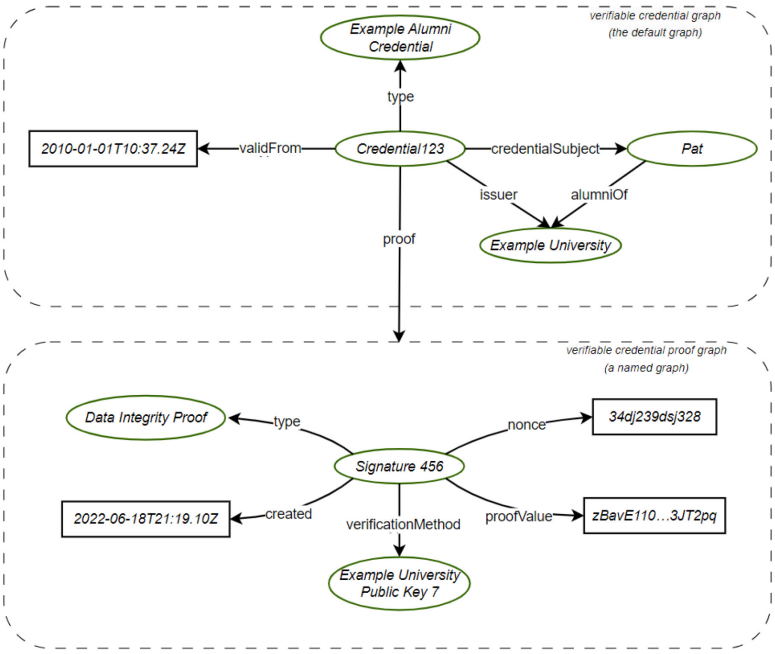

Figura 14: Grafi delle informazioni associati ad una Verifiable Credential
Fonte: https://www.w3.org/TR/vc-data-model-2.0/diagrams/vc-graph.svg

Una più fedele rappresentazione di una verifiable presentation è composta da almeno quattro grafi:

1. Il verifiable presentation graph
2. Il verifiable credential graph
3. Il proof graph
4. Il named graph

Il primo rappresenta la verifiable presentation stessa e contiene i metadati della presentation, il secondo è un grafico di credenziali verificabili indipendente che a sua volta contiene metadati di credenziali e altre claims. Il terzo grafico rappresenta il credential graph proof che è solitamente una firma digitale mentre l'ultimo grafico rappresenta la proof digitale della presentation che è anch'essa solitamente una firma digitale.

```
{
  "@context": ["https://www.w3.org/ns/credentials/v2"],
  "type": ["VerifiableCredential", "MyPrototypeCredential"],
  "credentialSubject": {
    "mySubjectProperty": "mySubjectValue"
  }
}
```

Figura 15: Esempio implementazione di una Verifiable Credential base
Fonte: https://www.w3.org/TR/vc-data-model-2.0/#example-a-template-for-creating-prototype-verifiable-credentials

Le verifiable credentials e le verifiable presentations hanno molti attributi e valori e sono identificati da URLs. Quest'ultimi possono essere mappati in delle rappresentazioni più user friendly con la proprietà @context.

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://www.w3.org/ns/credentials/examples/v2"
  ],
  "id": "http://university.example/credentials/58473",
  "type": ["VerifiableCredential", "ExampleAlumniCredential"],
  "issuer": "https://university.example/issuers/565049",
  "validFrom": "2010-01-01T00:00:00Z",
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "alumniOf": {
      "id": "did:example:c276e12ec21ebfeb1f712ebc6f1",
      "name": "Example University"
    }
  }
}
```

Figura 16: Verifiable Credential con più valori presenti nel campo Context
Fonte: https://www.w3.org/TR/vc-data-model-2.0/#example-usage-of-the-context-property

Qui con il primo URL si afferma che la semantica con la quale è stata realizzata la verifiable credential è quella contenuta nella documentazione di W3C. Col secondo URL si stabilisce che tale credenziale riguarda degli esempi contenendo il prefisso "https://www.w3.org/ns/credentials/examples/v2", che identifica la semantica di Verifiable Credentials v2.0 per esempi.

2.4 Blockchain

2.4.1 Descrizione e caratteristiche

Una blockchain è un registro digitale aperto e distribuito, in grado di memorizzare record di dati (solitamente, denominati "transazioni") in modo sicuro, verificabile e permanente. La blockchain è rappresentabile come una lista, in continua crescita, di blocchi collegati tra loro la cui sicurezza è

realizzata mediante l'uso della crittografia. Ad un blocco, inoltre, possono essere associate una o più transazioni. Ciascun blocco possiede al suo interno delle informazioni che vengono registrate al suo interno, l'hash del blocco stesso e l'hash del blocco precedente garantendo così la robustezza della struttura.

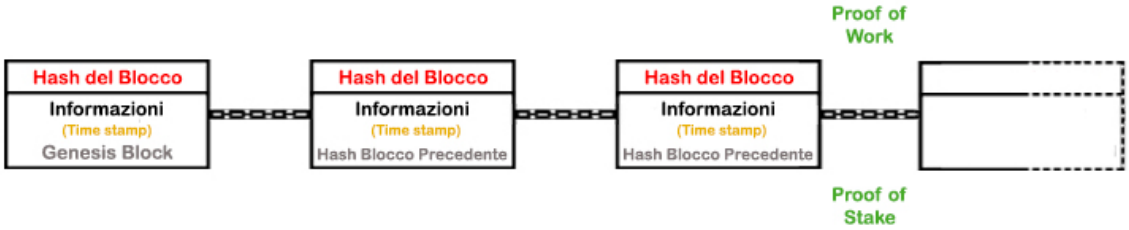

Figura 17: Rappresentazione di una blockchain

L'hash del blocco viene calcolato attraverso una funzione di hash che è una funzione matematica che ha il compito di garantire l'integrità del dato trasmesso. Controllare l'integrità di un dato significa assicurarsi che il file che viene inviato sia lo stesso di quello che viene ricevuto. L'integrità dei dati si riferisce alla loro completezza e correttezza, un dato è integro se è esatto e non è stato modificato e/o alterato in modo non autorizzato.
I partecipanti della rete blockchain sono i cosiddetti nodi, un nodo è ciascun computer partecipante alla rete che gestisce una copia del registro distribuito.
Un altro aspetto importante è quello che riguarda i protocolli e gli algoritmi di consenso.
ll protocollo di consenso è l'insieme delle regole di una blockchain, mentre l'algoritmo di consenso è il meccanismo attraverso cui queste regole vengono fatte rispettare dall'intera rete.
Il protocollo definisce il modo in cui i nodi interagiscono, come i dati devono essere trasmessi e quali sono i requisiti per convalidare le nuove operazioni o per modificare un'operazione già approvata.
L'algoritmo di consenso verifica la corretta interazione tra i nodi e conferma le operazioni eseguite attraverso il consenso della rete.

Le caratteristiche che rendono questa tecnologia così importante sono apertura, decentralizzazione e immutabilità. Con questo registro condiviso, le transazioni vengono registrate una sola volta, eliminando la duplicazione tipica delle reti tradizionali.

Nessun partecipante potrà modificare o manomettere una transazione, una volta registrata nel registro condiviso se non con l'approvazione di più del 50% dei nodi, pertanto, se la registrazione di una transazione contiene un errore, risulterà più semplice aggiungere una nuova transazione per correggere l'errore, ed entrambe le transazioni saranno visibili.
Per accelerare le transazioni tra due o più parti, un set di regole, chiamato contratto intelligente (smart contract), viene memorizzato sulla blockchain ed eseguito automaticamente. I contratti intelligenti sono semplicemente dei programmi archiviati su una blockchain che vengono eseguiti quando vengono soddisfatte delle condizioni prestabilite. Sono di norma utilizzati per automatizzare l'esecuzione di una transazione in modo che tutti i partecipanti possano essere immediatamente certi

del risultato senza il coinvolgimento di intermediari o perdite di tempo. Possono anche automatizzare un flusso di lavoro, attivando l'azione successiva quando vengono soddisfatte le condizioni.

Esistono vari modi per costruire una rete blockchain. Tali strutture possono essere pubbliche, private, basate su autorizzazioni o meno. Una blockchain pubblica è una rete a cui chiunque può accedere e partecipare, questa tipologia potrebbe però portare ad una necessità di una notevole potenza di calcolo, poca o nessuna privacy a tutela delle transazioni e scarsa sicurezza. Per sopperire in parte all'ultimo aspetto, la blockchain pubblica può anch'essa essere basata su autorizzazioni, ciò pone dei limiti relativamente a chi è autorizzato a partecipare alla rete e a quali transazioni può partecipare. I partecipanti devono ottenere un invito o un permesso per partecipare

Una blockchain pubblica utilizza computer connessi a Internet per convalidare le transazioni e ottenere il consenso. I computer sulla rete cercano di risolvere un problema crittografico complesso per creare proof of work e quindi convalidare la transazione. Al di fuori delle chiavi pubbliche, ci sono pochi controlli di identità e di accesso in questo tipo di rete.

2.4.2 Storia della Blockchain

Nel 1979 è stata introdotta una tecnologia pre-blockchain, il Merkle tree chiamato così dal suo ideatore Ralph Merkle. Egli descrisse nella propria tesi di dottorato alla Stanford University un "prototipo di sistema" che si basava sull'uso di chiavi pubbliche e di firme digitali chiamato tree authentication. Nel 1991 Stuart Haber e W. Scott Stornetta pubblicarono un articolo che descriveva come associare a documenti digitali i dati relativi all'ora e alla data di creazione e di ultima modifica (timestamp) così da prevenire il backdating e il forward-dating da parte degli utenti su documenti elettronici. Inoltre, aggiornarono la loro implementazione per incorporare anche i Merkle trees che permettevano a più documenti certificati di risiedere in un singolo blocco. Due anni più tardi ci fu l'introduzione delle proof of works (PoW).
Successivamente crebbe in popolarità il concetto di rete P2P rendendo così possibile costruire sistemi distribuiti che potevano beneficiare della potenza di calcolo e di memoria di migliaia di computers.
Nel 2000 Stefan Konst introdusse il concetto di "cryptographically secured chains" nel suo paper "Secure Log Files Based on Cryptographically Concatenated Entries". Il suo modello, che mostrava che le associazioni chiave-valore (entries) in una catena potevano essere percorse a ritroso fino al Genesis Block per provare l'autenticità di un'informazione, era la base per i modelli odierni di blockchains. Infine, nel 2008 fu pubblicato un paper ad opera di uno pseudonimo, Satoshi Nakamoto, che affermava che i sistemi basati su blockchain avrebbero fornito delle trasazione P2P sicure senza la necessità della presenza di terze parti certificate come banche o governi.

2.4.3 Principali minacce alla Blockchain

Le tecniche principali con le quali gli hacker e altri utenti malintenzionati minacciano i sistemi blockchain based sono quattro: attacchi di phishing, routing, Sybil e 51%'s attack.

La pratica del phishing comprende tutti i tentativi di furto di informazioni sensibili come ad esempio nomi utente, passwords, credenziali relative a dei metodi di pagamento o altre tipologie di dati con lo scopo di rivenderle. Il funzionamento di questa tecnica di estorsione dei dati si basa sull'utilizzo di link ipertestuali contenuti in e-mails o sms col fine di impersonale una fonte autorevole convincendo con qualche motivazione l'utente ad inserire le proprie credenziali. Deve la sua efficacia all'errore umano, nessun sistema informatico sarà mai quindi a prova di phishing se l'interazione umana con esso sarà coinvolta.

Spear phishing

Questa tipologia di phishing prende il nome dal suo aspetto di prendere di mira specifici individui o specifiche aziende. Il funzionamento dello spear phishing si basa sul comprare o raccogliere dettagli e informazioni su un particolare bersaglio, pertanto, la truffa può essere costruita in modo personalizzato. Lo spear phishing è la tecnica odierna più efficacie di phishing.

Clone phishing

Il clone phishing funziona imitando un'autentica e-mail consegnata precedentemente e modificando i links ipertestuali annessi o i files allegati mantenendo però l'indirizzo e-mail corretto del mittente col fine di persuadere gli utenti ad interagire col nuovo contenuto malevolo.

Whaling

Con questo termine si fa riferimento ad attacchi rivolti specificatamente ai vertici di un'azienda/istituzione o ad altri utenti di queste che hanno particolari privilegi all'interno della rete. Il funzionamento consiste nel persuadere un utente con motivazioni di comune interesse all'interno di un'azienda come, ad esempio, problemi esecutivi o legali.

Sybil Attack

Un Sybil attack, o "pseudospoofing", è un tipo di attacco informatico che mira a sovvertire il controllo di una rete. Coinvolge un singolo computer, conosciuto come nodo in una rete peer-to-peer (P2P) che cerca di utilizzare multiple identità false (Sybils o "sock puppets") simultaneamente con lo scopo di ottenere il controllo su una rete. Nel caso di una rete blockchain, i Sybils potrebbero cooperare per impedire il funzionamento di alcuni meccanismi del sistema come la conferma di transazione bloccando potenzialmente il funzionamento di sistemi che utilizzano una blockchain per registrare le transazioni effettuate come, ad esempio, il caso della rete Bitcoin. Questo tipo di attacco informatico potrebbe anche creare un consenso di massa illusorio facendo sì che una modifica fraudolenta

apportata ad un nodo venga confermata o che l'aggiunta di un'intero blocco con informazioni false sia permessa compromettendo così l'integrità dell'intero sistema.
Un altro scenario è rappresentato dal caso in cui l'hacker decida di congestionare l'intero sistema saturandolo di richieste fittizie e transazioni false in modo tale che il sistema non riesca a gestire tutta questa mole di informazioni e diventi inutilizzabile.
Se il numero di Sybils creati è sufficientemente elevato, in grado quindi di controllare più del 50% dei nodi di una rete, il Sybil attack può tramutarsi in un attacco del 51%.

Attacco del 51%

Questa tipologia di attacco informatico si verifica quando uno o più utenti riescono a prendere la maggioranza del controllo di una blockchain che si basa sul proof of work (ottengono sufficiente hash power). Con un tale potere di decisione, colui che sta performando l'attacco potrebbe decidere di censurare alcuni blocchi, di impedire il processo di conferma delle transazioni, quali protoccolli accettare e potrebbe anche decidere un nuovo protocollo di consensi sul quale basare il funzionamento dell'intero sistema. Ad oggi nessun attacco di questo tipo ha avuto successo su reti blockchain con hash power elevato come ad esempio la rete Bitcoin; tuttavia, altre reti di minore potenza e con minori misure di sicurezza sono state colpite con successo.

Routing Attack

È un tipo di attacco informatico diretto a un Internet service provider che mira a ridurre il tempo di attività o impedire agli utenti di accedere a un sistema web abilitato come una blockchain. Attraverso un routing attack, un hacker è in grado di dividere una rete in due o più parti separate impedendo quindi la comunicazione tra nodi diversi di una rete e creando uua blockchain parallela con lo scopo di simulare un corretto funzionamento del sistema e nel mentre sottrarre dati contenuti nella trasmissione.
Il Routing è una tipologia di attacco informatico efficiace contro i sistemi blockchain based perché mira all'intercettazione di grandi moli di dati durante il loro trasferimento.

2.4.4 Proof of Work e Proof of Stake

Le reti blockchain usano degli algoritmi di consenso come la proof-of-work (PoW) e la proof-of-stake (PoS) per garantire che tutti i nodi della rete siano d'accordo sulla cronologia delle transazioni. Questo è importante perché la blockchain è un registro distribuito, il che significa che non è controllato da un'autorità centrale, invece, la cronologia delle transazioni è mantenuta da tutti i nodi della rete.
Sistemi come quelli delle criptovalute come Bitcoin e Litecoin usano la proof-of-work mentre altri sistemi come Ethereum e derivati (ad esempio il progetto europeo EBSI: european blockchain service infrascrutcure) hanno preferito la proof-of-stake.

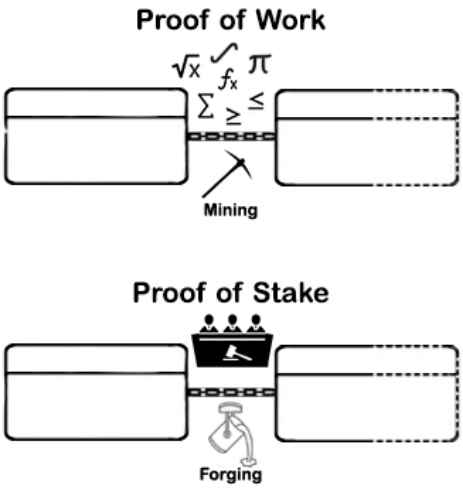

Figura 18: Rappresentazione grafica della Proof of Work e Proof of Stake

La proof of work (PoW) è stato il primo algoritmo di consenso per cryptovalute ad essere creato, è stato introdotto in un sistema nel 2009 dal creatore di Bitcoin e il suo originario scopo era quello di garantire la Byzantine Fault Tolerance, cioè la capacità di un sistema informatico distribuito di resistere a problemi come fallimento di consenso, fallimento di validazione, mancata verifica dei dati o errori nel protocollo di risposta. La PoW si basa sulla risoluzione di problemi matematici particolarmente complessi da parte della rete. Questa modalità di consenso utilizza il processo del mining: dei calcolatori molto potenti (miners) provano a risolvere un problema matematico complesso al fine di trovare una soluzione (l'hash di collegamento tra un blocco e l'altro). La risoluzione di tali problemi spesso può portare al ricevimento di una ricompensa per il lavoro svolto (come, ad esempio, un Bitcoin per aver risolto una proof of work nella rete Bitcoin avendo contribuito alla validazione di un blocco).

Più grande è la potenza di calcolo utilizzata, maggiori saranno i tentativi di risoluzione al problema matematico, più sarà probabile che una determinata PoW sia risolta; infatti, la risoluzione del quesito può essere performata solamente a tentativi; pertanto, il miner con la potenza di calcolo maggiore è quello con la probabilità più alta di risolvere il problema matematico e di ricevere quindi la ricompensa.

L'algoritmo di consenso proof of work consente ai miners di convalidare il nuovo blocco e aggiungerlo alla blockchain solo se gli altri nodi coinvolti nella rete concordano con la soluzione fornita, ripetendo l'operazione risolta. L'enorme dispendio energetico che penalizza questa modalità di validazione delle transazioni non è nato con l'introduzione della PoW ma si è presentato in seguito infatti, inizialmente chiunque poteva competere con gli altri miners per risolvere un problema matematico e validare il blocco successivo, in quanto la difficoltà iniziale dei quesiti proposti non richiedeva un'ingente potenza di calcolo. Col passare degli anni però, la crescente complessità dei problemi da risolvere portò alla nascita delle cosidette mining farm o mining pool, strutture molto costose finalizzate a risolvere complessi problemi matematici attraverso una grandissima potenza di calcolo dedicata. Il termine "mining" è conseguente al fatto che per ottenere la ricompensa derivante

dal risolvere il problema matematico i miners debbano utilizzare molte risorse come avviene per l'estrazione di minerali in una miniera.

La proof of stake è un algoritmo di consenso alternativo alla proof of work, è stata originariamente proposta nel 2011 da un utente sul forum Bitcointalk per risolvere le problematiche sollevate dall'utilizzo della PoW.
In questo sistema alternativo, il mining viene sostituito dall'utilizzo di "validatori del consenso distribuito" che hanno il compito di garantire la validità delle operazioni eseguite impegnando una parte delle proprie risorse o stake (ad esempio nella rete Ethereum queste risorse sono criptovalute). L'idea base risiede nell'evitare sprechi di energia e la competizione tra nodi basata sulla capacità di calcolo. La proceduta introdotta dalla proof of stake deve il suo funzionamento ad un sistema di selezione randomizzata dei validatori. I blocchi nella PoS non vengono minati come nella proof of work, ma vengono coniati (forge). I nodi per diventare validatori devono impegnare una parte delle proprie risorse nella rete come garanzia. Questa quota non potrà pertanto essere usata o spesa. Nella determinazione dei successivi validatori non concorre solo la quota di fondi impegnati ma anche altri fattori per impedire il monopolio della validazione da parte degli utenti con le quote maggiori. I criteri di decisione presi in considerazione nella fase di selezione generalmente sono:

- L'ammontare della quota versata
- La longevità dello stake (quanto è longevo il deposito, garantisce affidabilità)
- Un fattore di randomizzazione

Come affermato in precedenza, tanto più alti sono la quota depositata e il tempo di deposito, tanto è maggiore la possibilità di essere scelti come validatori, questo perché questi fattori concorrono nel garantire un maggior grado di affidabilità e integrità rispetto ad altri validatori. Una problematica che potrebbe nascere da questi fattori però potrebbe essere che i nodi con maggiori stake dominino la blockchain magari lasciando grandi depositi di risorse per elevati periodi di tempo, questo però non si verifica grazie al fatto che una volta scelto un nodo come validatore, la coin age dei propri depositi viene azzerata facendo sì che sembri che tali depositi siano stati appena depositati. Inoltre, una volta che un nodo viene scelto come validatore non potrà essere scelto una seconda volta di seguito. Quando un nodo viene selezionato come validatore del blocco successivo, dovrà controllare se le transazioni in esso contenute sono valide, firmare il blocco e aggiungerlo alla blockchain. Differentemente con quanto accade nei sistemi PoW based, nei sistemi basati su Proof of Stake la ricompensa per i validatori consiste in una percentuale trattenuta sulla transazione validata. Prima di poter ritirare la quota depositata in precedenza e incassare la propria ricompensa, la rete verifica l'operato del validatore, controllando che non siano stati aggiunti blocchi malevoli. Se viene individuata un'operazione fraudolenta, il nodo validatore perde parte della sua stake, oltre al diritto di essere selezionato come validatore in futuro. L'unico modo per raggirare i controlli del network e approvare transazioni fraudolente sarebbe quello di possedere più del 50% delle risorse del sistema ma in quel caso la risposta di quest'ultimo ad un attacco del genere sarebbe quella di ridurre ampiamente il valore

delle risorse (come, ad esempio, il valore delle criptovalute) diminuendo il numero di tali attacchi a causa dello scarso guadagno a monte di uno sforzo considerevole. Questa procedura di sicurezza risulta più efficacie rispetto al semplice disincentivamento proposto dalla proof of work che scoraggia le azioni fraudolente a causa degli alti costi computazionali ed energetici necessari per eseguirle. Il termine "forging" deriva dal fatto che per creare un nuovo blocco, gli utenti della blockchain debbano utilizzare delle loro risorse "forgiandolo" da esse.

I sistemi PoS based risultano più convenienti della controparte PoW based anche perché spesso i problemi computazionali che devono risolvere i miners di quest'ultima tipologia di sistemi risultano così complessi che richiedono tempo per la loro risoluzione. Questo significa che dal momento in cui viene effettuata una certa transazione al momento in cui questa viene validata e confermata, può passare un tempo più o meno lungo rendendo di fatto il sistema non affidabile. Nei sistemi PoS based questa problematica non si presenta perché il processo di selezione dell'algoritmo avviene in pochi secondi così come la validazione del nodo validatore.

2.4.5 Pro e Contro della Blockchain e Utilizzo ad oggi

Come affermato nei paragrafi precedenti, la blockchain è una tecnologia che permette di realizzare dei sistemi trasparenti e robusti contro alcune tipologie di attacchi informatici e sfrutta la potenza di calcolo di migliaia di dispositivi ad essa connessi come nei sistemi peer to peer. Il fatto che modificare delle informazioni contenute all'interno di una blockchain risulta particolarmente difficile, permette di creare delle infrastrutture con dati affidabili e rende la manomissione più complessa mentre il non necessitare di un controllo centrale consente di realizzare dei sistemi decentralizzati. La blockchain può automatizzare e semplificare molti processi attraverso gli smart contracts, riducendo i costi e il tempo necessario per le transazioni. Grazie alla sua caratteristica di essere difficilmente modificabile permette anche di garantire un'ottima tracciabilità delle informazioni memorizzate nei propri blocchi.

Oltre a tutti questi vantaggi, la blockchain presenta anche degli svantaggi, uno di essi è che essendo quasi immodificabile, nel caso di immissione di dati errati converrà immettere in seguito i dati corretti piuttosto che cercare di modificare l'informazione precedentemente inserita. Se da un lato la trasparenza della blockchain è un punto di forza, dall'altro non permette di conservare dati personali e sensibili senza violare il regolamento europeo GDPR. A causa dell'algoritmo di consenso della proof of work, il consumo energetico derivante dal processo di risoluzione dei problemi di calcolo e il relativo inquinamento solo uno dei maggiori problemi derivanti da questa tecnologia, nel 2020 l'attività di mining di Bitcoin ha richiesto 75,4 terawattora di elettricità, un consumo annuo di elettricità maggiore di stati come Austria o Portogallo infatti, con la crescente complessità dei quesiti matematici da risolvere con la proof of work, l'energia elettrica consumata è aumentata di 126 volte dal 2016 al 2021. Tuttavia, molte blockchain tra cui anche la nota Ethereum hanno effettuato la transizione verso l'algoritmo di consenso della proof of stake andando a diminuire significativamente

il loro impatto ambientale. La blockchain inoltre essendo una tecnologia particolarmente recente non garantisce una sicurezza dei dati ottimale in quanto non è un'infrastruttura largamente studiata e testata.

Per promuovere maggiormente l'impiego della tecnologia blockchain in Europa, l'Unione Europea ha dato il via a un nuovo progetto nel 2018 la European Blockchain Services Infrastructure (EBSI) a cui hanno aderito tutti gli stati membri, la Norvegia e il Liechtenstein creando così nel 2021 la European Blockchain Partnership con l'obiettivo di realizzare un ambiente favorevole allo sviluppo e all'adozione di questa tecnologia in tutti i settori dell'economia europea al fine di sfruttare appieno il potenziale della blockchain e promuovere l'innovazione. Tra i Paesi europei che si sono distinti in questo ambito troviamo la Svizzera, l'Estonia, Malta e l'Italia. Ad oggi il settore col maggior impiego della tecnologia blockchain è quello relativo alle criptovalute, tuttavia, per merito di numerose iniziative l'utilizzo di tale infrastruttura sta crescendo sempre più soprattutto in settori quali la finanza, la sanità, la sicurezza informatica e in politica (attraverso il voto online memorizzato su blockchain).

2.5 GDPR

2.5.1 Descrizione e Principi

È il regolamento generale sulla protezione dei dati (General Data Protection Regulation) nato nel 2016 il cui fine è quello di fornire sempre maggior privacy e controllo sui dati personali delle persone fisiche nel web.

Il GDPR basa l'efficacia del proprio funzionamento su sei principi cardine:

1. Correttezza e trasparenza nel trattamento dei dati personali
2. Limitazione nel trattamento stesso dei dati rispetto alle finalità per il quale sono raccolti
3. Minimizzazione dei dati personali trattati.
4. Esattezza ed aggiornamento dei dati personali trattati
5. Garantire integrità e riservatezza dei dati personali oggetto del trattamento.
6. Conservazione dei dati per un tempo non superiore a quello necessario rispetto agli scopi per i quali è stato effettuato il trattamento

In particolare, le aziende per poter usufruire dei dati personali degli utenti devono ottenere il consenso per il loro utilizzo. Il consenso è solo uno dei sei casi in cui è legittimo il trattamento dei dati ed è anche quello che presenta più restrizioni rendendo così più difficile alle aziende l'uso dei dati.

Gli altri casi di trattamento legittimo dei dati sono:

1. Obblighi contrattuali
2. Obblighi legali
3. Vitale interesse
4. Interesse pubblico

5. Interessi leggitimati

2.5.2 Differenza tra dati personali e dati sensibili

Il GDPR presta particolare attenzione al trattamento dei dati personali degli utenti, ma cosa rende personale un dato? E cosa lo contraddistingue da un dato sensibile?

Un dato personale è qualunque informazione che qualcuno può utilizzare per indentificare con un certo grado di accuratezza una persona. Un indirizzo e-mail, il domicilio e il numero di telefono sono considerati dati personali. Un'altra tipologia di dati personali è composta da qualunque tipo di informazione che possa provare la presenza in una data posizione di una persona come ad esempio delle impronte digitali. Oltre a tali casistiche in cui le informazioni personali sono prese singolarmente, si può anche prendere in considerazione il caso in cui le aziende prendano grandi moli di dati, se l'unione di parte di questi dati può portare ad una identificazione di un soggetto allora questi sono da considerare dati personali.

I dati sensibili invece sono un sottoinsieme dei dati personali che è necessario trattare con particolare attenzione. Essi devono inoltre essere conservati separatamente dai dati personali con un certo grado di sicurezza e comprendono alcune speciali categorie come, ad esempio, l'etnia di una persona, le sue idee politiche, le sue credenze religiose. Questo tipo di informazioni è considerato sensibile a causa delle implicazioni a cui può portare il fatto che esse siano in mano a qualcuno con fini illeciti. Per definizione, dei dati vengono considerati sensibili se la loro detenzione può portare a perdite finanziarie da parte di un'azienda, alla compromissione della privacy di un soggetto o a danni alla concorrenzialità di un'azienda.

2.5.3 Come un sistema su Blockchain può essere GDPR compliant

Generalmente vi è un contrasto tra il regolamento GDPR e la blockchain perché, se da un lato il primo si protae per garantire la protezione dei dati attraverso la loro riservatezza, la blockchain invece per garantire la sicurezza di questi opta per un'altra strategia, cioè quella della trasparenza; infatti, i dati contenuti all'interno dei nodi della blockchain sono leggibili da chiunque abbia accesso ad essa (questo varia in base al fatto che essa sia pubblica o privata) e questo garantisce che attraverso la struttura di cui una blockchain è composta, ogni transazione o modifica necessiti di essere validata da tutti gli altri nodi garantendone così un'elevata robustezza alle contraffazioni. La blockchain va quindi contro il principio di riservatezza del GDPR, inoltre un altro aspetto da considerare è che i dati salvati nella blockchain sono non cancellabili o difficilmente modificabili andando quindi contro ai principi di aggiornamento dei dati personali trattati e al principio di conservazione dei dati. Un esempio più estremo potrebbe essere quello in cui dei dati personali vengono condivisi sulla blockchain per una particolare procedura, successivamente tale procedura viene aggiornata e per effettuarla servono meno

dati di quelli richiesti in precedenza, in questo caso verrebbe meno anche il principio di minimalità del GDPR.

Un modo per integrare la tecnologia blockchain all'interno di un sistema GDPR compliant è quello di utilizzare questo tipo di struttura per una funzionalità che non coinvolge la manipolazione diretta di dati sensibili e personali, come quella di registrare le transazioni in una rete, un altro metodo potrebbe essere quello di svolgere una funzione ausiliaria alla memorizzazione dei dati personali, ad esempio, una blockchain potrebbe regolare gli accessi ad un database criptato all'interno del quale vi sono contenuti i dati personali.

3 Spiegazione del funzionamento del sistema

Come già detto in precedenza la self sovereign identity è un paradigma che permette al singolo utente di avere un maggior controllo sui propri dati digitali e questo è molto importante nel caso in cui i dati in questione siano personali o sensibili perché questi, oltre a richiedere un elevato grado di sicurezza, richiederanno un maggiore controllo su di essi data la loro importanza. Il campo di ricerca su cui verte questa tesi è quello medico che racchiude e tratta una grande mole di dati personali e sensibili, inoltre tale ambito richiede che i dati siano facilmente accessibili, non facilmente modificabili e quindi affidabili se consultati. In particolare, il caso d'uso in ambito medico scelto come oggetto di discussione di questa tesi è quello relativo alla gestione, aggiornamento, e condivisione della storia clinica di un paziente facendo sì che quest'ultimo sia in grado di condividere dati relativi alla propria storia clinica nella misura che egli riterrà necessario dando l'accesso ad essi solo a coloro che vorrà autorizzare potendo così realizzare i principi su cui si basa la SSI.

3.1 Ruolo dei Pilastri SSI nel Sistema

Decentralized Identifiers

Il primo pilastro della SSI, cioè i decentralized identifiers hanno un molteplice ruolo nel sistema in analisi, sui blocchi della Blockchain vi saranno salvati i loggings delle operazioni svolte sulle storie cliniche e questi loggings presenteranno il nome del documento su cui è stata apportata la modifica. Tale nome conterrà, come spiegato nel capitolo quattro, una parte relativa al DID URI corrispondente al DID document contenuto in un cryptowallet, precisamente il DID Subject (DID URI senza lo scheme) che consentirà una migliore distinzione dei loggings relativi a una storia clinica di un particolare paziente. Ogni utente utilizzerà un'applicazione che fungerà da cryptowallet con la quale potrà modificare il campo ServiceEndPoint del DID document relativo al DID URI precedentemente citato. Nel DID document potranno anche essere inserite informazioni aggiuntive per fini quali autenticazione o crittografia dei dati, in quest'ultimo caso verrà incluso il parametro relativo alla chiave pubblica del destinatario.

Pertanto, il ruolo dei decentralized identifiers in questo sistema è quello di permettere all'utente di accedere ai dati contenuti all'interno della propria cartella clinica, i quali verranno restituiti come conseguenza di query eseguite su un database esterno alla blockchain.

Verifiable Credentials

Le verifiable credentials in questo sistema hanno lo scopo di identificare una serie di documenti autenticati da una firma elettronica che risultano essere molto utili nelle interazioni quotidiane e spesso richiesti durante un controllo o una transazione. La tessera sanitaria è stata designata come verifiable credential perché viene richiesta in farmacia per l'acquisto di determinati farmaci, per prenotare un esame in un laboratorio di analisi, e per molti altri scopi.

Un'altra credenziale verificabile è la ricetta medica rilasciata per l'acquisto di un farmaco la cui assunzione è richiesta quotidianamente per periodi molto lunghi, spesso anche per tutta la vita come, ad esempio, farmaci per trattare l'ipertensione e altre patologie permanenti. In questo caso sarà necessario porre particolare attenzione su aspetti quali la revoca di tale credenziale, la modifica nel caso di cambio farmaco o dosi e la quantità massima acquistabile in un determinato periodo di tempo.

Un'altra credenziale verificabile potrebbe essere una certificazione come quella richiesta per portare in aeroporto dei farmaci speciali come i farmaci salvavita nel caso di diabetici o di persone affette da allergie che possono sfociare in shock anafilattici, farmaci per persone cardiopatiche, ecc…

Un'ulteriore credenziale verificabile potrebbe contenere delle informazioni importanti come se si posseggono protesi, tatuaggi o piercing, ci si è sottoposti a determinati interventi e anche informazioni relative ad eventuali allergie a farmaci specifici e il proprio gruppo sanguigno nel caso di necessità di trasfusione e di stato di incoscienza da parte del singolo. Sempre in questo ultimo gruppo si potrebbero inserire in maniera del tutto facoltativa il credo religioso di appartenenza.L'idea cosisterebbe nel disporre le prime quattro casistiche nella stessa credenziale verificabile e di inserire le informazioni su gruppo sanguigno, credo religioso e anamnesi familiare relativa sia alle patologie fisiche che mentali delle prime due generazioni di lontananza in un'unica VC.
Le credenziali verificabili come la tessera sanitaria saranno rilasciate dal cryptowallet di un ente istituzionale come il ministero dell'economia e delle finanze mentre le altre credenziali verificabili come i documenti per i farmaci per il trattamento delle allergie, gli interventi ai quali ci si è sottoposti, l'indicazione relativa a se si indossano protesi di qualche tipo e le allergie a determinati farmaci saranno rilasciate dai cryptowallet del personale medico. Infine, le rimanenti credenziali verificabili (come tatuaggi e piercing) saranno rilasciate dagli studi certificati.

Come affermato in precedenza, tutte le credenziali verificabili necessiteranno di una firma digitale per dimostrarne l'autenticità e preservarne l'integrità. Inoltre, tutte le credenziali verificabili erogate, indipendentemente dall'ente di erogazione di queste, saranno conservate all'interno del cryptowallet del proprietario di esse e sarà possibile per l'utente mostrarle ad un verificatore nella loro interezza o in parte.

Blockchain

L'ultimo componente chiave del paradigma SSI è la blockchain, in questo sistema avrà il compito di garantire l'integrità delle informazioni e fornire una maggiore resistenza contro eventuali attacchi informatici. La blockchain fungerà da libro mastro contenente i registri di tutte le operazioni di lettura/scrittura effettuate sulla particolare storia clinica. Ogni blocco della blockchain sarà identificato da un hash e conterrà quindi loggings associati ad un particolare DID document contenuto all'interno di un cryptowallet. In questo sistema la blockchain sarà permissioned e pertanto solo gli utenti autorizzati potranno visionarne il contenuto. Per verificare che un utente sia abilitato ad accedervi, il gateway della blockchain, quando riceve una richiesta di accesso ad essa, controlla che nella richiesta vi sia un parametro del DID document che corrisponde a un DID Subject di un file di logging del blocco che si cerca di accedere.

3.1.1 Ciclo del processo

Il sistema baserà il proprio funzionamento oltre che sui pilastri del modello SSI anche sull'uso di applicazioni simili a cryptowallets. Un cryptowallet è un software o dispositivo hardware che permette di archiviare e di gestire le chiavi crittografiche associate alle criptovalute possedute. Nel nostro sistema un'applicazione simil cryptowallet conserverà le chiavi pubbliche e private che consentono il corretto funzionamento delle transazioni; tuttavia, non conterrà nessun'informazione relativa a cryptovalute ma avrà al suo interno le credenziali verificabili associate col proprietario del cryptowallet e una modalità per modificare un DID document contenuto in esso.
Un utente all'interno dell'app che fungerà da cryptowallet andrà a modificare un DID document preesistente, questo potrà avere al suo interno anche dei parametri per effettuare diverse tipologie di queries per, ad esempio, ottenere solo i dati in un certo formato oppure solo i dati che hanno una certa tag che li contraddistingue come ad esempio "Oculista". Il DID document, attraverso la propria modifica, consentirà di inviare una richiesta personalizzata (entro certi limiti) o al gateway della blockchain, nel caso si volesse effettuare il logging delle operazioni precedentemente effettuate o al gateway del database nel caso in cui l'utente voglia reperire dei dati della propria cartella clinica per visionarli o condividerli con terzi. Nel caso in cui invece l'utente sia un membro del personale medico, potrebbe comunicare col gateway del database per inserire referti medici o modificare documenti precedentemente rilasciati.

La scelta implementativa riguardo l'impiego di due gateways è stata fatta per motivi di ridirezione, load balancing e robustezza agli attacchi informatici. I cryptowallets degli utenti del sistema saranno configurati in modo tale che si possano collegare unicamente con i due gateways, questi saranno intercambiabili cosicchè se un gateway è sovraccarico, l'altro provvederà ad alleggerire il primo. La configurazione relativa al collegamento tra cryptowallets degli utenti e gateways del sistema descritta in precedenza ha lo scopo di impedire a dispositivi terzi di routing di intercettare le comunicazioni. Quest'ultime saranno crittografate con modalità descritte nel capitolo seguente.

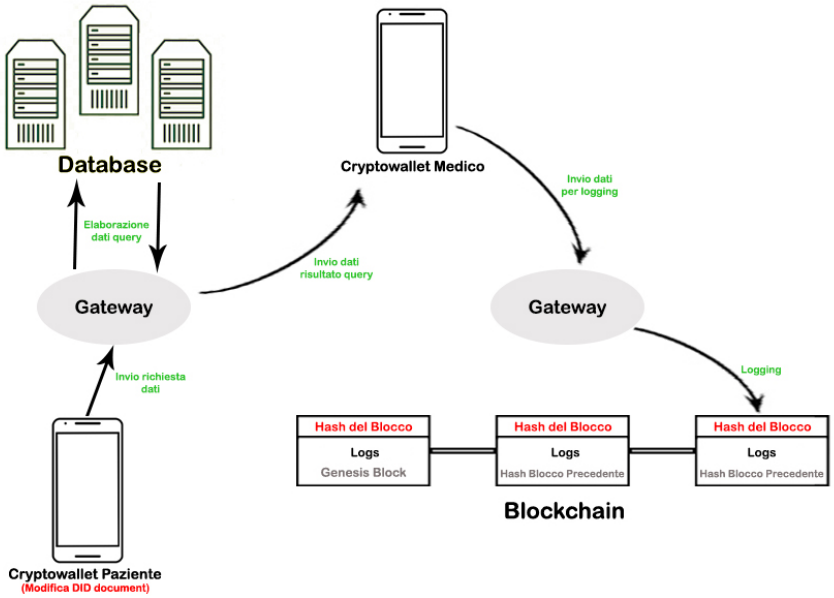

Figura 19: Rappresentazione del funzionamento del sistema

3.2 Come il sistema è GDPR compliant

Come affermato in precedenza, il regolamento europeo per la protezione dei dati (GDPR) ha la funzione di tutelare la privacy del singolo cittadino e di far sì che, qualora i dati di quest'ultimo siano necessari per una particolare procedura, siano presi in misura minima, specificandone l'utilizzo che se ne andrà a fare. Per garantire tali principi il GDPR pone particolare attenzione al ciclo di utilizzo dei dati personali. Una problematica dei sistemi SSI-Based sull'essere GPDR compliant sta nel terzo componente del paradigma self sovereign identity, ovvero la blockchain. Quest'ultima in particolare ha, tra le varie caratteristiche, quella della trasparenza, che permette a chiunque di consultare i dati memorizzati in essa e quindi presenti nei nodi componenti la blockchain (questo in realtà è particolarmente vero per le blockchain pubbliche mentre per quelle private o permissioned la situazione potrebbe variare), questa caratteristica non permetterebbe quindi di inserire nella blockchain dati personali che beneficerebbero della caratteristica dell'immutabilità di questo tipo di strutture. Questo perché nonostante si possa pensare, come nel caso in esame, di memorizzare nei vari blocchi i dati relativi alla storia clinica di un singolo omettendone il nome, quest'ultimo potrebbe comunque essere riconosciuto o il suo riconoscimento potrebbe essere facilitato compromettendo la sicurezza della persona e questo non è permesso dal GDPR. Pertanto, nel sistema in analisi in questa tesi si è

pensato di memorizzare sulla blockchain solo i file di loggings, identificati dal DID Subject nel loro nome, relativi ai DID documents dei cryptowallets degli utenti del sistema. I dati relativi alle storie cliniche dei pazienti saranno invece contenuti in un database esterno criptato.

4. Caratteristiche Sistema

4.1 Immagazzinamento dati

I dati relativi alle storie cliniche dei pazienti sono salvati all'interno di un database. Dopo un'attenta analisi sui pro e sui contro dell'utilizzo di varie tipologie di databases, è stato scelto di adottare un metodo di archiviazione su cloud, gli aspetti analizzati per effettuare questa scelta sono costi, gestione, scalabilità, disponibilità e sicurezza.

Per quanto concerne l'analisi dei costi, l'adozione di un sistema di archiviazione su cloud consente di non dover acquistare hardware e software, il cui acquisto risulterebbe necessario nel caso di un database dedicato, inoltre il costo deriva unicamente dal quantitativo di risorse effettivamente utilizzate per il funzionamento del sistema. La gestione di un database su cloud risulta esemplificata perché il fornitore del servizio cloud si occupa della manutenzione, degli aggiornamenti e può fornire degli strumenti aggiuntivi per funzioni come quella di monitoraggio. La scalabilità del sistema si interconnette all'aspetto dei costi, infatti, il database su cloud è in grado di scalare rapidamente sia orizzontalmente che verticalmente in base al carico di lavoro e questo porterà una maggior efficienza del sistema ma anche costi maggiori sempre relativi al quantitativo di risorse utilizzate.
Infine, per quanto riguarda gli aspetti di disponibilità e sicurezza, il fornitore di servizi cloud si occupa di fornire dei protocolli di disaster recovery, di fornire una replicazione geografica dei dati e di aggiornare periodicamente i protocolli di sicurezza così da mantenere un'alta resistenza agli attacchi informatici.

La soluzione di archiviare i dati su cloud è stata preferita quindi all'impiego di un database dedicato per tutti i vantaggi presentati in precedenza; tuttavia, questo ha comportato il dover fare un trade off tra benefici portati da questa soluzione adottata e i principi della Self Sovereign Identity perché, questo modello vuole che il singolo utente sia pienamente padrone dei propri dati tuttavia, il doversi affidare ad un servizio terzo per l'archiviazione dei dati va a ledere in parte questo aspetto.

4.1.1 Database per archiviazione dati

È un database SQL based che permette di eseguire sui propri dati delle query, queste saranno richieste da un gateway che fungerà da intermediario tra il cryptowallet del paziente o del medico e il database. Questo sistema di archiviazione conterrà una cartella contenente delle cartelle con le storie cliniche di tutti i vari pazienti, una cartella per il logging degli accessi, una cartella per gli utenti autorizzati, una cartella con i mappings dei codici di emergenza associati al personale sanitario e gli indirizzi web dei

rispettivi cryptowallet e un file robots.txt per gestire le richieste di questi programmi software automatici. Il logging però espone la privacy degli utenti, perciò, bisogna maneggiare tali dati con estrema cautela. La comunicazione avverrà attraverso un canale di comunicazione TLS mentre i dati contenuti all'interno del database dovranno essere criptati per maggiore sicurezza perché il TLS garantisce la crittografia dei dati solo durante il loro transito.

4.1.2 Query del sistema

Le query inizialmente disponibili saranno quelle relative all'aggiunta e alla modifica di dati nella cartella del paziente, questa può essere effettuata solo dal medico e la query per recuperare un determinato file o un particolare sottoinsieme di files della cartella della storia clinica, quest'ultima sarà invece eseguibile solo dal paziente che ottenute le informazioni può decidere se comunicarle direttamente al medico o visionarle. Per differenziare gli utenti tra personale medico e non, farei in modo che il Ministero dell'Economia e delle Finanze (ovvero l'ente responsabile dell'invio della produzione e della gestione delle tessere sanitarie), attraverso il proprio cryptowallet, invii solo ai cryptowallet dei medici la credenziale verificabile relativa alla tessera sanitaria con un parametro aggiuntivo per indicare che l'utente è un medico e poi la credenziale viene firmata digitalmente e viene messo un timestamp per la validità e ogni volta che il medico dovrà performare un'azione invierà la credenziale verificabile che dovrà essere verificata dal gateway del database.

Per valutare con maggior accuratezza l'aspetto della sicurezza dei dati, che è fondamentale nel caso di dati medici, ho suddiviso il flusso di dati del sistema nelle tre situazioni più frequenti ovvero quelle descritte dalle query precedentemente elencate.

Modifica/Aggiunta dati

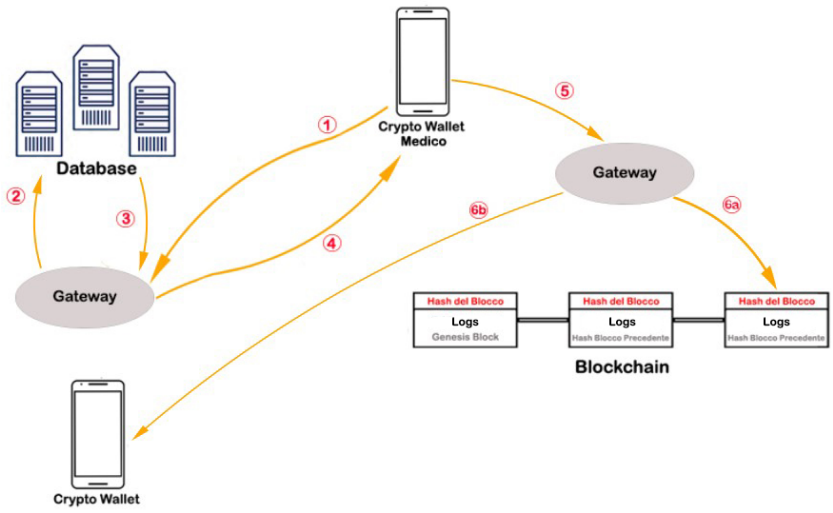

Figura 20: Flusso delle richieste per l'aggiunta o la modifica di dati nella storia clinica

1. Richiesta per aggiungere/modificare files mandata dal cryptowallet del medico al gateway del database

2. Il gateway inoltra la query al database
3. Il gateway prende dal database i dati restituiti dalla query
4. Il gateway notifica il cryptowallet del medico dell'esito dell'operazione
5. Il cryptowallet del medico notifica il gateway della blockchian dell'operazione fatta nel database

6a. Il gateway memorizza l'operazione in un blocco della blockchain

6b. il gateway notifica, infine, l'operazione svolta sul database al cryptowallet dell'utente coinvolto

Recupero dati per invio al medico

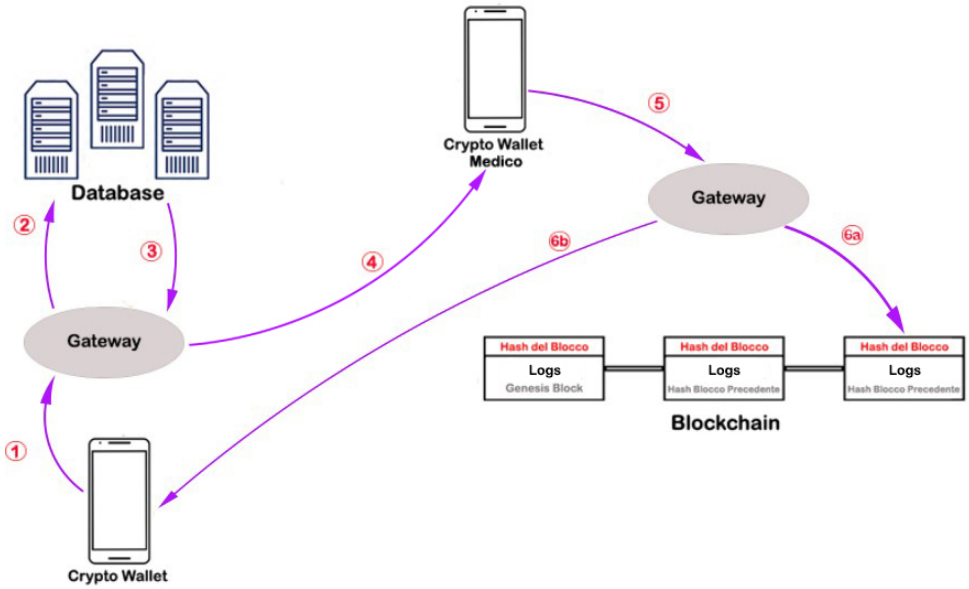

Figura 21: Flusso delle richieste per l'invio di referti medici della storia clinica a un altro utente (medico in questo caso)

1. Richiesta del cryptowallet del paziente al gateway del database
2. Richiesta del gateway al database per eseguire la query
3. Il database ritorna al gateway il risultato della query
4. Invio dei risultati della query al cryptowallet del medico
5. Richiesta di logging dell'operazione dal cryptowallet del medico al gateway della blockchain

6a. Logging eseguito dal gateway delal blockchain su un blocco della blockchain

6b. Invio della notifica relativa all'operazione svolta

Recupero dati per consultazione

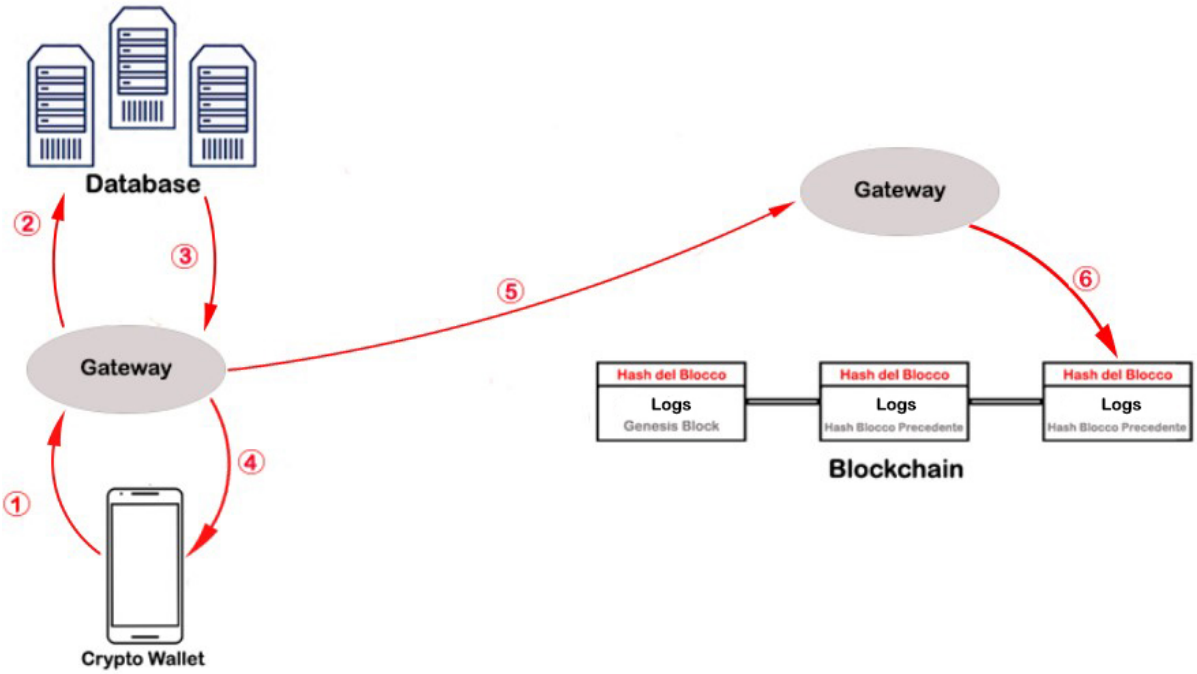

Figura 22: Flusso delle richieste per la visione dei referti della propria storia clinica

1. Richiesta del cryptowallet del paziente al gateway del database per recupero dati
2. Il gateway ridireziona la richiesta al database che effettua una query e ritorna i dati richiesti
3. Il gateway recupera i dati risultato della query
4. Il gateway li invia al cryptowallet del paziente
5. Infine, il gateway del database informa il gateway della blockchain dell'operazioni svolte
6. Il gateway della blockchain provvede a fare il logging delle operazioni in un blocco della blockchain

Un'ulteriore casistica è stata presa in considerazione oltre le precedenti tre, quest'ultima fa riferimento ad un protocollo di emergenza attuato nei casi in cui il paziente non sia in grado di comunicare i propri dati o le proprie credenziali verificabili, per esempio, perché è in uno stato di incoscienza o è in uno stato alterato. La definizione di tale protocollo però risulta particolarmente complessa a causa delle varie possibili complicazioni infatti, se inizialmente si potrebbe pensare di mostrare a schermo le credenziali verificabili relative a situazioni mediche di emergenza come gruppo sanguigno, particolari allergie a farmaci o religione di appartenenza, va considerato che il paziente potrebbe aver smarrito il telefono, non averlo con sé, essere stato derubato di questo oppure più semplicemente si potrebbe essere rotto in un incidente. Due ulteriori complicazioni sono date dal fatto che il paziente potrebbe non essere stato accompagnato e potrebbe, per dei motivi simili ai precedenti, non avere il portafoglio che potrebbe risultare essenziale per l'identificazione del soggetto.

Pertanto, per la definizione di tale protocollo mi sono basato su alcune assunzioni come, ad esempio, il fatto che la persona sia provvista di telefono e che questo sia funzionante. Il medico inserirà un codice

nel cryptowallet del dispositivo della persona. Questo inserimento non richiederà che il dispositivo sia sbloccato, le modalità di utilizzo sono simili alle chiamate di emergenza e alla fotocamera (funzioni permesse anche senza lo sblocco del dispositivo). Questo codice verrà inviato al database che lo confronterà con i codici di emergenza che possiede e notificherà il cryptowallet del medico corrispondente a tale codice per chiedere conferma della procedura. Una volta che il medico avrà acconsentito dal proprio cryptowallet, sul dispositivo del paziente verrà mostrata una schermata con le credenziali verificabili essenziali. Questo è possibile perché nel database saranno conservate le corrispondenze tra codici di emergenza e indirizzi web dei cryptowallet corrispondeti dei medici

4.1.3 Memorizzazione dei dati nel database

Un file nel database viene memorizzato con un nome specifico per garantire l'integrità e l'ordinamento dei dati, oltre a facilitare l'impiego delle query. Di seguito viene mostrato una possibile modalità di archiviazione dei dati:

AnnoMeseGiorno-OraMinSec-DID_Subject@TAG

Es.

20240227-182853- DID_Subject @Ottico

Il parametro TAG serve per catalogare il documento per poter in seguito facilitare l'uso di query, così da poter richiedere solo i referti medici appartenenti ad una certa categoria. Questo parametro deve essere impostato correttamente nel cryptowallet del personale medico per poter effettuare la query corretta e memorizzare senza errori il documento.

4.2 Crittografia dei dati nel sistema

Per preservare la sicurezza delle richieste e dei dati coinvolti in questo sistema, ho scelto come approccio quello della crittografia asimmetrica per le comunicazioni contenenti i dati maggiormente sensibili e ho scelto di utilizzare la crittografia simmetrica mediante canale TLS per le comunicazioni di dati meno sensibili. Questa è un sistema di crittografia che utilizza due chiavi matematicamente correlate ma distinte: una chiave pubblica e una chiave privata. La prima permette di codificare i dati in modo che non siano leggibili da chiunque mentre la seconda è necessaria per decodificare i dati criptati dalla corrispondente chiave pubblica. Per preservare l'integrità dei dati e del sistema si è scelto di utilizzare come chiavi private e pubbliche quelle degli Identificatori Decentralizzati (DID) presentati come componente essenziale del sistema. Invece per creare le suddette chiavi si è scelto di utilizzare l'algoritmo di crittografia RSA.

Crittografia RSA

Come detto in precedenza è un algoritmo di crittografia asimmetrica e pertanto fa uso di una coppia di chiavi che ne regola il funzionamento. Le caratteristiche che lo accomunano agli altri algoritmi asimmetrici sono:

- La generazione delle chiavi pubblica e privata deve essere non onerosa a livello computazionale
- Deve essere semplice per il mittente di un messaggio codificare con la propria chiave pubblica un messaggio o firmare digitalmente un documento con la propria chiave privata
- Anche la decodifica e l'analisi di validità di una firma digitale devono essere computazionalmente semplici
- Deve essere molto complesso il determinare una chiave privata conoscendo la chiave pubblica o decifrare un documento senza la chiave privata

L'algoritmo RSA è nato nel 1977 ed è tuttora largamente utilizzato. La sua resilienza agli attacchi informatici deriva dalla difficoltà computazionale di riuscire a fattorizzare numeri molto grandi (composti anche di centinaia di cifre).
Dato un numero n molto grande, non esistono metodi particolarmente efficaci per determinare due numeri primi p e q tali che il loro prodotto dia n. L'algoritmo RSA utilizza questa coppia di numeri primi per generare una chiave privata.
Sia M il messaggio da processare di lunghezza finita, il numero di bit processabili dipende da n perché la rappresentazione di M in binario deve necessariamente essere un numero compreso tra 0 e n-1.
Ciò nonostante, è possibile processare messaggi di lunghezza maggiore di n attraverso la loro scomposizione in blocchi di dimensione minore di n.

Attraverso il processo di codifica vengono quindi creati dei blocchi cifrati C che rispettano la relazione $C = M^e$
Per decodificare un messaggio sarà invece necessario utilizzare la relazione $M' = C^d mod\ n$

La scelta accurata degli esponenti 'd' ed 'e' fa in modo che il messaggio decifrato corrisponda con l'originale. È necessario però che sia mittente che destinatario del messaggio conoscano il valore di n a priori.

La conoscenza dell'esponente 'e' consentirà di codificare un messaggio e la conoscenza dell'esponente 'd' ne consentirà la decodifica.
La chiave privata sarà quindi costituita dalla coppia (d,n) mentre la chiave pubblica dalla coppia (e,n).

Per ottenere un certo margine di sicurezza con l'algoritmo RSA, è necessario far uso di interi che siano composti da almeno 300-500 cifre, secondo gli ultimi studi fatti, fattorizzare un numero di 200 cifre richiede circa 4 miliardi di anni di tempo macchina; fattorizzare un numero di 500 cifre, richide circa

10^25 anni (supponendo di utilizzare un computer con un tempo di 1ns ad istruzione e utilizzando un algoritmo ottimo per la scomposizione in fattori dei numeri primi). Queste stime però si basano sulla tecnologia odierna, pertanto, l'uso sempre maggiore dell'ancora poco utilizzata intelligenza artificiale e il futuro impiego di computer quantistici potrebbero richiedere l'uso di chiavi più lunghe o di adottare altre soluzioni alternative all'algoritmo RSA.
Ogni attore del sistema sarà pertanto dotato di una chiave privata e di una corrispondente chiave pubblica. Il cryptowallet degli utenti conterrà di default la chiave pubblica del database così da potergli inviare delle richieste decifrabili solo da quest'ultimo. Il cryptowallet del medico dovrà contenere le chiavi pubbliche di tutti i suoi pazienti (inizialmente saranno condivise da questi, poi saranno memorizzate in una rubrica interna al cryptowallet del medico per velocizzare i processi futuri). Infine, i pazienti dovranno contenere al loro interno la chiave pubblica del medico.

Nelle figure 20-21 il processo di crittografia avviene nel seguente modo:
- Le richieste 1/2 sono criptate con un canale TLS
- Le richieste 3/4 sono criptate con la chiave pubblica del medico
- Le richieste 5/6a/6b sono criptate con la chiave pubblica del paziente

Nella figura 22 il processo di crittografia avviene nel seguente modo:
- Le richieste 1/2 sono criptate con un canale TLS
- Le altre richieste sono criptate con la chiave pubblica del paziente

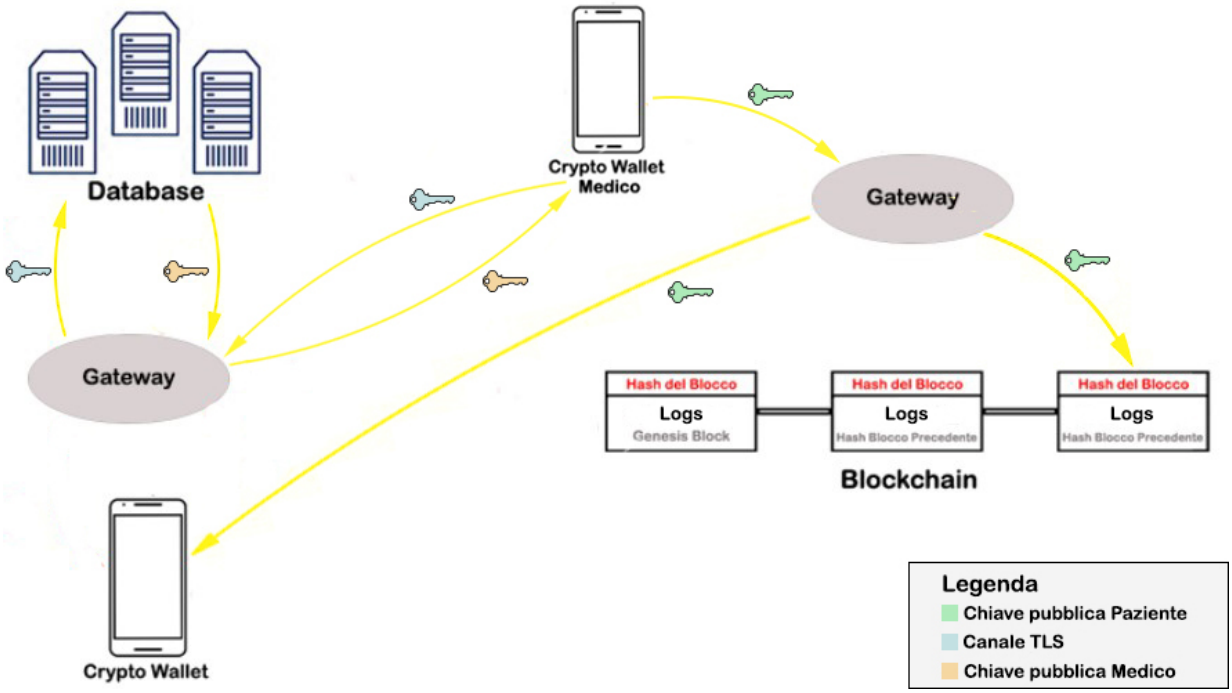

Figura 23: Esempio crittografia query per aggiunta/modifica referti nel database

4.3 Autenticazione

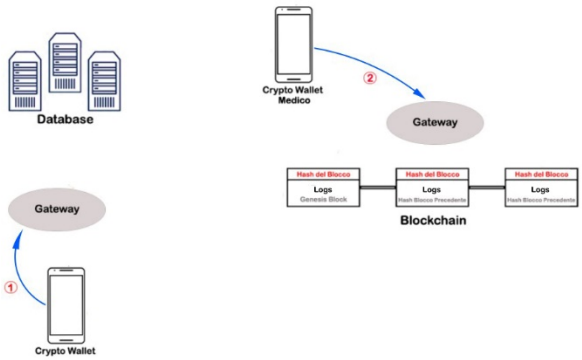

Figura 24: Autenticazione per processo di invio referti al medico

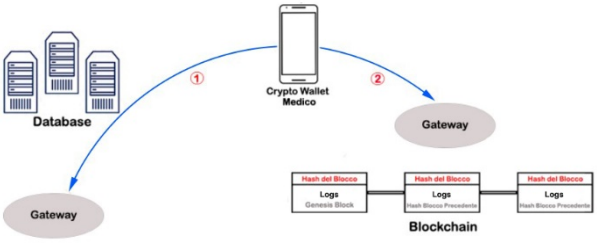

Figura 25: Autenticazione per processo di modifica/aggiunta referti

4.3.1 Firma elettronica

Per quanto riguarda il processo dell'autenticazione, questo processo viene realizzato attraverso l'uso di una Firma digitale creata con chiave privata e apposta all'interno di una richiesta, il mittente usa la chiave pubblica della controparte per verificare l'autenticità.

La firma digitale è una tipologia di firma elettronica che soddisfa requisiti particolarmente stringenti garantedo così autenticità, integrativa, affidabilità e validità legale ai documenti.
Si basa sui concetti di:
– autenticità: assicura e garantisce che colui che ha firmato il documento ne attesta l'autenticità e se ne assume le responsabilità
– integrità: proprietà che attesta il fatto che il dato non è stato modificato rispetto all'originale
– non ripudio: colui che ha apposto la propria firma elettronica sul documento non può disconoscerlo in nessuna situazione

Per garantire il rispetto di tali proprietà, la firma elettronica basa il proprio funzionamento sulla crittografia. Quindi nel sistema della firma digitale ogni utente ha assegnate una coppia di chiavi. L'assegnazione è a carico di un soggetto istituzionalmente qualificato (il certificatore), in Italia l'ente certificatore è l'Agenzia per l'Italia Digitale.
Contemporaneamente alla chiave pubblica viene generata un'altra chiave di pari lunghezza che è privata ed a controllo esclusivo del titolare. La chiave è conservata in un ambiente sicuro (nel nostro caso un cryptowallet). Per firmare, l'utente utilizza un software che calcola l'impronta digitale del documento tramite la funzione di hash. A documenti diversi corrispondono impronte digitali differenti.

Determinata l'impronta digitale, il software la invia all'ambiente sicuro dove è custodita la chiave privata (dispositivo di firma). Infine, il dispositivo di firma procede alla cifratura dell'impronta del documento con la chiave privata. Il risultato dell'operazione è la firma digitale del documento.
Per verificare la firma, il destinatario utilizza uno specifico software che estrae la chiave pubblica dal certificato del titolare, spacchetta il file con documento e firma. Ricalcola l'impronta e decifrando con la chiave pubblica la firma del titolare può verificare se l'impronta del mittente e quella ricalcolata dal destinatario sono identiche.
In caso positivo la firma è valida. Nell'altro caso la firma non è valida e bisogna indagare sul problema specifico che ha determinato l'errore.

4.3.2 Revoca delle credenziali

L'aspetto della revoca delle credenziali è molto importante in ambito medico, potremmo analizzare tale funzionalità con due chiavi di lettura diverse. Per revoca delle credenziali potremmo intendere sia la revoca dell'accesso dai dati contenuti nel database, sia la revoca di alcune o tutte le credenziali verificabili erogate dal sistema a un utente.

Per quanto riguarda la prima interpretazione, considerando il fatto che, quando un utente (medico o paziente) vuole accedere ai dati contenuti nel database per visualizzarli o modificarli, il gateway del database controlla che i mittenti della richiesta siano autorizzati (controllando in una cartella interna del database), sarà sufficiente rimuove da tale cartella l'identificativo dell'utente a cui si vuole revocare l'accesso.
Per quanto riguarda la seconda interpretazione si potrebbe dare un periodo di lifetime a una credenziale verificabile al termine del quale essa non potrà più essere considerata valida. Potendo trattare le credenziali verificabili come file json mostrati a schermo, l'aggiunta di un campo "scadenza" permette di garantire un certo periodo di validità. Questa scelta implementativa è stata preferità all'uso del timestamp nella firma elettronica perché non impedisce la modifica dei dati dopo la firma.

5. Considerazioni sul sistema

5.1 Scalabilità

La scalabilità di un sistema è la sua capacità di gestire un aumento del carico di lavoro o del volume di dati in modo efficiente e senza compromettere le prestazioni. Più semplicemente, un sistema scalabile è in grado di crescere insieme alle esigenze degli utenti, adattandosi a un numero maggiore di richieste e di dati senza rallentamenti o interruzioni.

Esistono due tipi di scalabilità:
La scalabilità verticale o scale up si riferisce alla massimizzazione delle risorse di una singola unità in modo da riuscire a gestire un aumento di carico. Questo comporta un potenziamento del processore e della memoria della macchina fisica su cui viene eseguito il server a livello hardware. Mentre a livello software, ci si riferisce alla scrittura di algoritmi di ottimizzazione. Le tecniche di ottimizzazione dell'hardware, come la parallelizzazione o l'avere un numero ottimizzato di processi in esecuzione, sono anche considerate tecniche di scale up. Nonostante la scalabilità verticale sembri facile da realizzare, presenta numerosi svantaggi. L'aggiunta di risorse hardware comporta un incremento dei costi da sostenere per l'espansione, oltre a ciò, il sistema dovrà essere mantenuto inattivo per un certo periodo di tempo affinchè venga ridimensionato. Inoltre, se tutti i servizi e i dati risiedono su una singola unità, la scalabilità verticale su questa unità non garantisce la disponibilità, perchè il sistema è "single point of failure" e, una volta che si guasta quella singola unità, il servizio non pùo più essere fornito.
La scalabilità orizzontale o scale out si riferisce all'aumento di risorse ottenuto andando ad aggiungere altre unità al sistema. Un modo per realizzare ciò risiede nell'aggiungere più unità con capacità ridotta piuttosto che aggiungere una singola unità con capacità maggiore. Le richieste vengono, in questo modo, distribuite su più unità, riducendo l'eccesso di carico su una singola macchina.
Avere più unità permette di mantenere il sistema attivo e disponibile, anche nel caso in cui qualche unità presenti guasti, evitando così la problematica del "single point of failure" che si verifica nel caso di scalabilità verticale aumentando così la disponibilità del sistema. Tuttavia, anche la scalabilità orizzontale presenta alcuni svantaggi. L'incremento del numero di unità del sistema si traduce nella gestione e manutenzione di più risorse. Anche il codice dell'applicazione necessita delle modifiche per ottenere parallelismo e distribuzione del carico tra le varie unità. Nella maggior parte dei casi, la scrittura di questo codice potrebbe essere particolarmente complessa rendendo così la scalabilità orizzontale difficile da attuare. Un altro problema che presenta tale tipologia di scalabilità è quello del load balancing. Si hanno più processori che risiedono su macchine fisiche diverse, ma bisogna trovare un meccanismo per distribuire le richieste su di esse in modo equo così da bilanciare il carico di lavoro tra le diverse macchine. In particolare, queste richieste vengono inviate allo stesso indirizzo IP, poichè il sistema distribuito viene visto dall'esterno come una singola entità. Il problema è decidere quale macchina o unità di processamento dovrà gestire una determinata richiesta. Anche in questo caso sarà compito del servizio di cloud hosting quello di bilanciare il carico delle richieste tra le varie macchine componenti il server del provider del servizio attraverso la pratica della redirezione.
Nel sistema descritto da questa tesi, la scelta del tipo di scalabilità da utilizzare ricade sul provider del servizio di clouding che ospita i dati. Spesso questi servizi offrono una combinazione della scalabilità verticale e della scalabilità orizzontale, la prima viene più comunemente impiegata per ridurre i costi e migliorare le prestazioni del sistema nel caso di elaborazioni complesse mentre l'utilizzo della seconda viene preferito per gestire grandi volumi di dati e gestire picchi di traffico.

5.2 Alcune considerazioni sull'impatto ecologico e sulla sicurezza del sistema

Nel sistema in analisi è stato scelto di utilizzare la Proof of stake per le sue emissioni quasi trascurabili, una prova di ciò si trova col passaggio della rete Ethereum dall'algoritmo di consenso proof of work all'algoritmo di consenso proof of stake, il quantitativo di emissioni annuo è diminuito circa del 99%. Questo perché, come spiegato in precedenza, la proof of stake ha un numero fisso di nodi validatori per i processi di validazione dei blocchi e quindi con l'aumentare della lunghezza della blockchain le tempistiche e la complessità rimangono pressocchè invariate. Nella proof of work invece, tutti i nodi facenti parte della blockchain dovevano partecipare ai processi di validazione e pertanto la lunghezza della blockchain e la complessità della validazione dei nuovi blocchi erano strettamente collegate. Oltre ai consumi legati all'impiego della blockchain vanno considerati anche quelli relativi a tutte le comunicazioni nel sistema tra i vari attori che lo compongono e i consumi del database, quest'ultimi verranno però gestiti dal servizio di cloud hosting al quale ci si sta affidando per la conservazione dei dati.

Una soluzione per gestire con modalità migliori la parte rimanente delle emissioni, cioè quella non eliminata dall'impiego della proof of stake è quella di adottare come fonte di alimentazione una tipologia di energia rinnovabile, come ad esempio l'energia solare, nonostante questa presenti attualmente ancora elevati costi per la produzione di pannelli solari oppure l'energia eolica che ad oggi è la fonte rinnovabile più largamente diffusa in europa, questo per i suoi aspetti quali fattori geografici del territorio, costi e tecnologia ben sviluppata. Tuttavia, per alimentare un sistema su larga scala come quello in esame, si dovranno comunque utilizzare altre fonti di alimentazione per motivi di sicurezza e affidabilità.

Nel sistema inoltre, si prova a mitigare la complessità della ricerca del blocco corretto sul quale effettuare il logging delle operazioni svolte in precedenza, utilizzando una cartella degli indici sul database nella quale sono presenti le associazioni tra hash dei blocchi della blockchain e DID URIs contenuti in essi, questo può risolvere in parte il problema tuttavia, aumenterà la complessità del sistema e richiederà particolare cura nell'aggiornare correttamente la cartella ad ogni aggiunta di un nuovo blocco nella blockchain.
Per quanto concerne la sicurezza delle transazioni e del sistema, ci si affida all'utilizzo della crittografia attraverso le chiavi pubbliche degli utenti coinvolti nella particolare transazione. Va precisato però che il database richiederà particolari misure di sicurezza a causa dell'importanza dei dati raccolti e inoltre, sebbene la soluzione proposta proponga di crittografare i dati del database con la chiave pubblica di quest'ultimo, questo non garantisce che i dati non possano essere trafugati e decrittografati in qualche altro modo. Per l'aspetto della sicurezza del sistema quindi, oltre agli aspetti e alle scelte implementative descritte in questa tesi, si fa affidamento sulla capacità del servizio di clouding che ospita i dati del sistema di aggiornare le proprie politiche di sicurezza e di mitigare possibili attacchi informatici e data leaks.

6. Conclusione

In conclusione, il modello Self Sovereign Identity può offrire molteplici vantaggi significativi per la tutela dei dati digitali degli utenti, tra cui il controllo individuale, la sicurezza avanzata e la trasparenza e può fornire un punto di partenza per la valorizzazione dell'identità digitale e della sicurezza dei dati ad essa associati. Conseguentemente all'analisi della SSI effettuata in questo elaborato si può affermare che la crittografia delle chiavi pubbliche e private associate ai decentralized identifiers può garantire un elevato grado di sicurezza, tuttavia, questo dipende dall'algoritmo utilizzato per la creazione delle chiavi e questo processo può essere molto oneroso a livello computazionale. Le credenziali verificabili invece permettono al singolo utente di controllare il quantitativo di dati esposti durante un controllo e di avere i propri documenti in un formato digitale e verificato attraverso la firma elettronica. Una problematica però, sta nel fatto che per poter usufruire dei servizi permessi dalle proprie VC è necessario possedere un dispositivo elettronico e portarlo sempre con sé e la dimenticanza di quest'ultimo potrebbe portare a non pochi disagi. Anche l'uso della blockchain è una grande innovazione che sta gradualmente venendo adottata da molteplici aziende in molteplici ambiti, tuttavia, va ricordato che tutte queste tecnologie sono relativamente nuove e pertanto non possono garantire appieno l'aspetto della sicurezza, basti pensare ad esempio che gli identificatori decentralizzati (DID) non sono ancora stati definiti totalmente. Il paradigma SSI offre un'ampia gamma di vantaggi e di innovazioni ma a pari passo con esse vi sono degli svantaggi portati dagli scarsi studi effettuati su tali tecnologie data la loro recente creazione.

In conclusione, il modello Self Sovereign Identity si presenta come una soluzione promettente ed efficace per la tutela dei dati digitali in diversi ambiti, sarà però necessario effettuare ulteriori studi e ricerche in modo tale da poter sfruttare tutte le potenzialità offerte da tale paradigma e analizzare maggiormente i suoi punti di forza e di debolezza.

7. Fonti

[1] the laws of identity (pag. 1): https://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf

[2] Cos'è la Self Sovereing Identity: https://www.selfsovereignidentity.it/category/guide/

[3] Decentralized Identifiers: https://www.w3.org/TR/did-core/

[4] Definizione URI: https://www.techtarget.com/whatis/definition/URI-Uniform-Resource-Identifier

[5] Documentazione URI: https://www.w3.org/Addressing/URL/uri-spec.html

[6] Verifiable Credentials: https://www.w3.org/TR/vc-data-model-2.0/

[7] Introduzione alla tecnologia Blockchain: https://www.ibm.com/it-it/topics/blockchain

[8] Storia della Blockchain: https://www.geopop.it/la-storia-della-tecnologia-blockchain-dalle-origini-ai-suoi-utilizzi-attuali-e-futuri/

https://www.techtarget.com/whatis/feature/A-timeline-and-history-of-blockchain-technology

[9] Integrità dei dati trasmessi e funzione di Hash: https://www.geopop.it/come-controlliamo-lintegrita-dei-dati-che-scarichiamo-online/

[10] Blockchain nel dettaglio: https://www.ibm.com/it-it/topics/blockchain

[11] Proof of Work e Proof of Stake: https://www.skrill.com/en/crypto/the-skrill-crypto-academy/advanced/the-difference-between-proof-of-work-and-proof-of-stake/

[12] Phishing: https://www.cloudflare.com/it-it/learning/access-management/phishing-attack/

[13] 51% Attack: https://www.bitpanda.com/academy/en/lessons/what-is-a-51-attack-and-how-is-it-prevented/

[14] Routing:
https://wazirx.com/guide/glossary/routing-attack/#:~:text=A%20cyberattack%20directed%20at%20an,(or%20more)%20separate%20parts.

[15] Sybil: https://www.ledger.com/academy/glossary/sybil-attack

[16] Regolamento GDPR:
https://www.garanteprivacy.it/documents/10160/0/Regolamento+UE+2016+679.+Arricchito+con+riferimenti+ai+Considerando+Aggiornato+alle+rettifiche+pubblicate+sulla+Gazzetta+Ufficiale++dell%27Unione+europea+127+del+23+maggio+2018

[17] Eccezioni che consentono il trattamento dei dati oltre al consenso:
https://www.itgovernance.co.uk/blog/gdpr-lawful-bases-for-processing-with-examples

[18] Differenza tra dati sensibili e dati personali:
https://www.itgovernance.co.uk/blog/the-gdpr-do-you-know-the-difference-between-personal-data-and-sensitive-data

[19] Crypto wallet: https://www.kaspersky.it/resource-center/definitions/what-is-a-crypto-wallet

[20] Gateways, URIs, Web hosting: David Gourley, Brian Totty, (2002), HTTP: The Definitive Guide, O'Reilly & Associates (Consultazione capitoli 2, 3, 6, 8, 18)

[21] Sicurezza dati del database:
https://www.oracle.com/it/security/database-security/what-is-data-security/

[22] Database su cloud: https://www.oracle.com/it/database/what-is-a-cloud-database/

[23] Algoritmi di crittografia asimmetrica:
https://www.computersec.it/2017/02/17/algoritmi-crittografia-asimmetrici/#:~:text=I%20pi%C3%B9%20diffusi%20e%20conosciuti,questo%20%C3%A8%20giunto%20a%20destinazione.

[24] Algoritmo di crittografia RSA:
https://www.computersec.it/2019/01/17/algoritmo-di-crittografia-rsa/

[25] Efficienza del RSA:
https://vitolavecchia.altervista.org/efficienza-del-rsa-rivest-shamir-adleman/

[26] Sicurezza del RSA:
https://vitolavecchia.altervista.org/sicurezza-del-rsa-rivest-shamir-adleman/

[27] Funzionamento del RSA in dettaglio: Nigel Smart, (2013), Cryptography: An Introduction 3rd Edition (pag. 172-177)

[28] Cos'è la Firma elettronica: https://www.agid.gov.it/it/piattaforme/firma-elettronica-qualificata

[29] Funzionamento della Firma elettronica:
https://www.agendadigitale.eu/documenti/firma-digitale-cose-come-funziona-e-come-ottenerla/

[30] Scalabilità:
https://vitolavecchia.altervista.org/che-cose-a-cosa-serve-e-tipi-di-scalabilita-nei-sistemi-distribuiti/

[31] Usman W. Chohan, (2019) , Blockchain and Environmental Sustainability: Case of IBM' s Blockchain Water Management
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3334154

[32] Adele Parmentola, Antonella Petrillo, Ilaria Tutore, Fabio De Felice, (2021), Is blockchain able to enhance environmental sustainability? A systematic review and research agenda from the perspective of Sustainable Development Goals (SDGs)
https://onlinelibrary.wiley.com/doi/pdfdirect/10.1002/bse.2882