UNIVERSITÀ
DEGLI STUDI
DI PALERMO

NDS
NETWORKING AND DISTRIBUTED SYSTEMS
RESEARCH GROUP

Progettazione e sviluppo di un sistema di gestione delle aste online tramite Blockchain Algorand

Tesi di Laurea Magistrale in Ingegneria Informatica

Luca La Barbera

Relatore: Prof. Alessandra De Paola

Indice

1 Introduzione **4**

2 La Blockchain **7**

2.1 Storia della Blockchain 7

2.2 Cosa è la Blockchain 9

2.3 Tipi di Blockchain 10

2.3.1 Algoritmi di consenso 10

2.3.2 Partecipazione alla blockchain 12

2.4 Smart Contract 13

2.4.1 Casi d'uso 14

3 E-Auctions **16**

3.1 Teoria delle Aste 16

3.1.1 Tipi di Asta 17

3.2 E-Auction Systems 18

3.2.1 Smart Auctions 20

3.3 Sicurezza e Privacy nelle piattaforme di E-Auction 22

3.3.1 Modello con distribuzione logica e gerarchica delle chiavi 22

3.3.2 Modello con cifrario asimmetrico e schema di firma digitale 24

3.3.3 Modello con crittografia omomorfica 25

3.3.4 Modello con Computazione Multiparte e Prove a conoscenza zero non interattive 26

4 Analisi Blockchain Algorand **31**

4.1 Algorand Blockchain 31

4.2 Rounds . . . 32
4.3 Blocchi . . . 32
4.4 Account . . . 32
4.5 L' algoritmo di consenso . . . 36
4.5.1 Verifiable Random Function . . . 36
4.5.2 Procedura operativa . . . 38
4.6 Smart Contract in Algorand . . . 44
4.6.1 Stateless Smart Contracts . . . 44
4.6.2 Statefull Smart Contracts . . . 44
4.7 Vantaggi di Algorand . . . 45

5 Sistema Proposto **47**
5.1 Limiti di approcci standard, sfide e scenari applicativi . . . 47
5.2 Meccanismo d'asta proposto . . . 48
5.3 Architettura del sistema . . . 49
5.4 Smart Contract . . . 51
5.4.1 Storage Globale e Locale . . . 51
5.4.2 Struttura Smart Contract . . . 52
5.5 Protocollo per il mantenimento della privacy . . . 54
5.6 Flusso delle operazioni . . . 62
5.6.1 Fase 1 - Creazione dell'asta . . . 62
5.6.2 Fase 2 - Partecipazione all'asta e registrazione dell' offerta . . . 63
5.6.3 Fase 3 - invio della chiave . . . 64
5.6.4 Fase 4 - Collezione delle chiavi e delle offerte, calcolo dei valori delle offerte e decretazione vincitore dell'asta . . . 66
5.7 Web Application . . . 68
5.8 Valutazione Sperimentale . . . 71
5.8.1 Commissioni e saldo minimo in Algorand . . . 71
5.8.2 Simulazione costi asta . . . 73
5.8.3 Compromessi realizzativi . . . 75

6 Conclusioni **76**

A Appendice A - Algoritmi di Sicurezza **78**
A.1 AES - Advanced Encryption Standard 78
A.2 RSA 79
A.3 Diffie-Hellman 81
A.4 Crittografia delle curve ellittiche 83
A.5 ECDH - Elliptic Curve Diffie-Hellman 84
A.6 SHA-256 85

B Appendice B - Web Application **88**

Elenco delle figure **93**

Elenco delle tabelle **94**

Bibliografia **96**

Capitolo 1

Introduzione

Negli ultimi decenni le piattaforme elettroniche d'asta hanno radicalmente trasformato le dinamiche preesistenti, offrendo nuove opportunità e democratizzando l'accesso a questa forma di commercio. In passato, le aste erano intrinsecamente legate a eventi fisici, in cui gli offerenti convergevano presso una determinata location per contendersi i lotti in palio. Questo formato, però, presentava una serie di ostacoli: la necessità di spostarsi fisicamente per partecipare, la limitata visibilità delle aste e l'accessibilità limitata al pubblico. Inoltre, l'organizzazione e la gestione di aste fisiche richiedevano un impegno logistico considerevole. L'avvento delle piattaforme elettroniche ha superato tali limiti, aprendo le porte a un mercato globale. Oggi, grazie alle piattaforme di aste online, i partecipanti possono prendere parte a queste transazioni da qualsiasi luogo del mondo, comodamente seduti al proprio computer o utilizzando dispositivi mobili. Questo ha enormemente ampliato la portata delle aste, consentendo a un numero considerevolmente più vasto di offerenti di partecipare e concorrere per l'acquisizione di beni ambiti. L'evoluzione della blockchain [22] e l'emergere delle applicazioni distribuite rappresentano una pietra miliare nell'evoluzione tecnologica contemporanea, ridefinendo le fondamenta dei sistemi informatici e aprendo nuovi scenari d'innovazione. La blockchain, come tecnologia di registrazione distribuita e immutabile, ha catalizzato l'attenzione degli esperti di settore, spianando la strada per lo sviluppo di applicazioni decentralizzate che sfidano le convenzioni tradizionali. In tempi passati, la fiducia e la validazione delle transazioni dipendevano da intermediari centralizzati, quali banche o istituzioni finanziarie, che agivano da custodi e garanti. Tuttavia, con l'avvento della blockchain, si è assistito a una svolta epocale in cui la fiducia non è più delegata a un'unica entità centrale, ma è distribuita tra i partecipanti di una rete peer-to-peer [12]. La

blockchain offre un registro immutabile e trasparente, dove le transazioni vengono verificate e validate in modo crittograficamente sicuro e reso accessibile a tutti i nodi della rete.
Algorand rappresenta una innovazione nel campo della blockchain, riuscendone a rompere il trilemma, introducendo una nuova variabile: l'*"efficacia computazionale*. Questa è stata scelta proprio per le sue caratteristiche principali: velocità ed efficienza. Le Smart auction, o aste intelligenti, sono una forma di asta che sfrutta tecnologie avanzate come la blockchain e i contratti intelligenti per offrire un'esperienza di asta completamente innovativa [44]. Questo approccio combinato permette di ottenere maggiore efficienza, trasparenza e automazione nei processi di asta. Grazie all'utilizzo della blockchain, le Smart auction garantiscono un registro immutabile e trasparente delle transazioni, eliminando il rischio di frodi o manipolazioni. I contratti intelligenti, invece, consentono di automatizzare le fasi dell'asta, inclusa la verifica delle offerte, la gestione dei pagamenti e la consegna degli oggetti, riducendo al minimo l'intervento umano e garantendo un processo equo e affidabile.
Una piattaforma di Smart auction offre i vantaggi di decentralità uniti alla sicurezza in un ambiente senza alcun intermediario. Ciò comporta un abbassamento dei costi a favore dei partecipanti dell'asta.
Un limite che pone la blockchain è quello della mancanza di privacy dei dati delle transazioni. Infatti se da un lato abbiamo una grande riduzione dei costi con una garantita trasparenza delle transazioni, non è possibile strutturalmente garantire la privacy, in contesti come quelli delle aste, dove la privacy dei dati delle offerte risulta fondamentale per il corretto svolgimento delle operazioni. I moderni sistemi di aste elettroniche superano questo problema introducendo un protocollo basato su tecniche crittografiche, per preservare la privacy dei dati. Questo lavoro propone la progettazione e lo sviluppo di una piattaforma di aste elettroniche basata sulla blockchain Algorand, introducendo un protocollo di mantenimento della privacy modulare che è ritenuto il giusto *trade-off* tra complessità computazionale, costi e livello di sicurezza garantito. Il sistema proposto può essere integrato in un ambiente con dispositivi intelligenti e autonomi, che interagiscono con esso al fine di gestire in maniera totalmente automatizzata le loro manutenzione.
Il resto della tesi procede in tal modo:

- Nel capitolo due verrà fatto un excursus della blockchain, riassumendone in breve la storia, analizzando le caratteristiche che contraddistinguono una blockchain da un'altra. In fine vi è una sezione dedicata agli Smart Contract e al loro uso nel mondo tecnologico odierno.

- Nel capitolo tre viene effettuata una analisi introduttiva della teoria delle aste e dei tipi di asta principali. Successivamente viene effettuata una analisi delle architetture per sistemi E-auction presenti, accompagnata da una analisi delle principali tecniche di privacy preserving di questi sistemi, riscontrate in letteratura. Infine verrà fatta una analisi delle Smart Auction.
- Nel capitolo quattro verrà affrontata e analizzata approfonditamente la Blockchain Algorand. Ne verranno analizzati i punti cardine, con una particolare attenzione al protocollo di consenso.
- Nel capitolo cinque viene esposto il sistema proposto in questo lavoro, analizzando: l'architettura del sistema, la struttura dello Smart Contract implementato, il protocollo per la privacy dei dati, il flusso operativo di una asta nel sistema e la progettazione e implementazione della Web Application. Verranno effettuata una valutazione sperimentale del sistema con attenzione particolare alla scalabilità della soluzione implementata.
- Nel capitolo sei verranno fatte le considerazioni in merito a possibili sviluppi del sistema e nuove opportunità.

Capitolo 2

La Blockchain

In questo capitolo viene effettuata una descrizione della blockchain, come funziona, quali tipologie di blockchain esistono e un confronto tra queste. Vengono introdotti anche gli Smart Contract, e il loro uso nella blockchain.

2.1 Storia della Blockchain

La blockchain è stata introdotta per la prima volta nel 2008 da una persona o un gruppo di persone sotto lo pseudonimo di Satoshi Nakamoto. Nakamoto ha pubblicato un documento tecnico intitolato "Bitcoin: A Peer-to-Peer Electronic Cash System" [35] che ha presentato i principi di base del funzionamento della blockchain e del Bitcoin come la prima criptovaluta basata su questa tecnologia. Nel gennaio 2009 è stata rilasciata la prima implementazione di software della blockchain e del Bitcoin [34], consentendo alle persone d'iniziare a utilizzare la criptovaluta e partecipare alla rete blockchain. Sebbene Satoshi Nakamoto abbia introdotto la blockchain con il Bitcoin, non è noto se Nakamoto sia una singola persona o un gruppo d'individui. L'identità di Satoshi Nakamoto rimane ancora un mistero non risolto, e molte speculazioni sono state fatte nel corso degli anni senza una conferma definitiva. Da allora, la tecnologia della blockchain si è evoluta e sono state sviluppate numerose altre criptovalute e applicazioni basate su blockchain. Oggi, la blockchain viene adottata e utilizzata da diverse aziende, istituzioni finanziarie e organizzazioni in tutto il mondo per una vasta gamma di scopi oltre alle criptovalute. Il grafico Hype cycle 2.1 mostra una proiezione dell'aspettativa di evoluzione delle applicazioni della blockchain. Nel grafico traviamo nell'asse delle ascisse il tempo, suddiviso in cinque fasi:

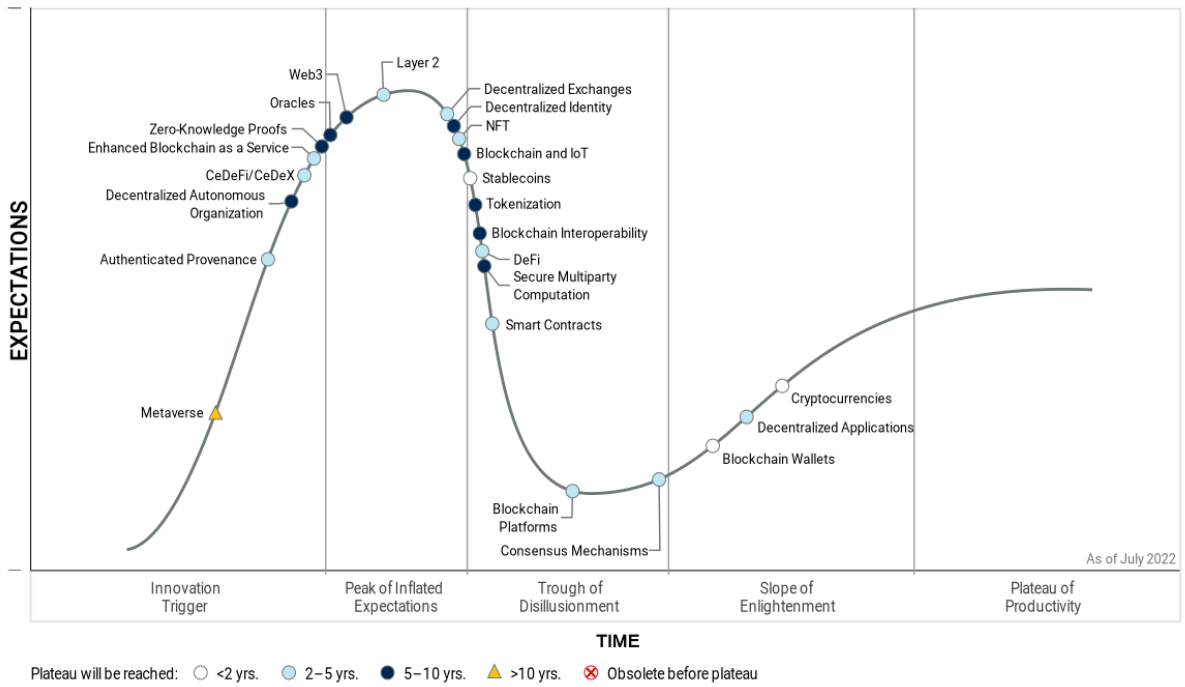

Figura 2.1: Ciclo Gartner Blockchain[1]

1. ***Innovation Trigger:*** Una nuova tecnologia potenzialmente dirompente viene avviata. Lo sviluppo di primi *Proof of concept* e l'attenzione dei media scatenano una notevole pubblicità. Spesso non esistono ancora prodotti utilizzabili e non c'è prova della validità commerciale della tecnologia.

2. ***Peak of Inflated Expetation:*** La pubblicità iniziale dà luogo a una serie di storie iniziali di successo, spesso accompagnate da molti casi di fallimento. Alcune imprese agiscono, anche se molte non lo fanno

3. ***Tought of Disillusionament:*** L'interesse nella tecnologia svanisce quando la sperimentazione e l'implementazione non producono i risultati sperati. I produttori della tecnologia entrano in crisi o falliscono. Gli investimenti continuano solo per quelle imprese sopravvissute grazie al miglioramento delle tecnologie di prodotto per soddisfare gli utenti precoci (o early adopters).

[1] `https://blogs.gartner.com/avivah-litan/2022/07/22/gartner-hype-cycle-for-blockchain-and-web3-2022/`

4. ***Slope of Enlightenement:*** Incomincia a diffondersi e ad affermarsi la consapevolezza di come la tecnologia possa avvantaggiare le imprese in diversi modi. Gli sviluppatori della tecnologia aggiustano il tiro creando prodotti di seconda e terza generazione. Un numero crescente d'imprese finanzia progetti pilota, mentre quelle conservatrici restano prudenti.

5. ***Plateau of Productivity:*** L'adozione della tecnologia è la principale tendenza che prende sempre più piede. Vengono stabiliti più dettagliatamente i criteri di valutazione dell'affidabilità dei fornitori. L'applicabilità e la rilevanza della tecnologia per il mercato di massa stanno chiaramente producendo frutti.

Dal grafico si nota come le applicazioni della blockchain siano per la maggior parte nella fase di picco di aspettativa. Da ciò si evince che a oggi si è ancora in una fase di sperimentazione per la maggior parte dei sistemi e delle soluzioni che utilizzano la tecnologia blockchain.

2.2 Cosa è la Blockchain

A livello tecnico, la blockchain è una struttura dati distribuita e immutabile che registra le transazioni in modo sicuro e trasparente. Per comprendere come funziona la blockchain, è necessario esaminare alcuni dei suoi elementi chiave:

1. ***Struttura a blocchi:*** La blockchain organizza le transazioni in blocchi. Ogni blocco contiene un insieme di transazioni confermate, insieme a un timestamp e a un riferimento al blocco precedente. Questo crea una catena continua di blocchi, da cui deriva il nome "blockchain".

2. ***Rete peer-to-peer:*** La blockchain è gestita da una rete di nodi o partecipanti. Ogni nodo possiede una copia del registro completo della blockchain e collabora con gli altri nodi per verificare, validare e diffondere le transazioni nella rete.

3. ***Algoritmi di consenso:*** Per raggiungere un consenso sulla validità delle transazioni, la blockchain utilizza algoritmi di consenso. Il più noto è la prova del lavoro (proof-of-work), in cui i nodi competono per risolvere problemi crittografici complessi. Una volta che un nodo risolve il problema, il blocco viene aggiunto alla blockchain e il nodo viene ricompensato con una certa quantità di criptovaluta. Esistono anche altri algoritmi di consenso, come la prova della partecipazione (proof-of-stake), che si basa sulla detenzione di una quantità significativa di criptovaluta per validare i blocchi.

4. ***Crittografia:*** La blockchain utilizza la crittografia per proteggere le transazioni e garantire l'integrità dei dati. Le transazioni sono firmate digitalmente per autenticare l'autore e impedire la falsificazione. Inoltre, i blocchi sono collegati tramite funzioni crittografiche, in modo che la modifica di un blocco richieda la modifica di tutti i blocchi successivi, rendendo la blockchain resistente alle modifiche fraudolente.

5. ***Decentralizzazione:*** La blockchain è decentralizzata, il che significa che non esiste un'autorità centrale che controlla il registro. Le transazioni vengono validate e confermate dai nodi distribuiti nella rete, rendendo la blockchain resistente alla censura e al single point of failure.

6. ***Trasparenza:*** Tutte le transazioni sulla blockchain sono pubblicamente visibili e accessibili a tutti i partecipanti della rete. Ciò garantisce la trasparenza e la rende utile per la verifica e l'auditing delle transazioni.

Quando viene effettuata una transazione sulla blockchain, viene trasmessa alla rete e verificata dai nodi. Una volta che la transazione è stata validata e inclusa in un blocco, viene propagata attraverso la rete e aggiunta alla blockchain. Una volta che un blocco è stato aggiunto, diventa quasi impossibile modificarlo senza invalidare l'intera catena, rendendo la blockchain resistente alla manipolazione dei dati. Complessivamente, la blockchain fornisce un meccanismo per creare registri digitali sicuri, immutabili e distribuiti, consentendo la condivisione d'informazioni in modo affidabile e trasparente tra le parti coinvolte.

2.3 Tipi di Blockchain

É possibile effettuare una classificazione delle blockchain esistenti in base a due diversi criteri : il protocollo di consenso e il metodo di partecipazione alla blockchain.

2.3.1 Algoritmi di consenso

Gli algoritmi di consenso sono fondamentali per il funzionamento delle blockchain, in quanto definiscono il modo in cui i nodi all'interno di una rete raggiungono un accordo sullo stato del registro distribuito. Ci sono diversi algoritmi di consenso utilizzati nelle blockchain, e ognuno

[2] `https://focus.namirial.it/blockchain/`

Figura 2.2: Blockchain in generale[2]

di essi presenta caratteristiche uniche. Di seguito sono descritti alcuni dei principali algoritmi di consenso.

1. ***Proof of Work (PoW)***: La PoW è l'algoritmo di consenso utilizzato dalla blockchain di Bitcoin. In PoW, i nodi della rete competono per risolvere complessi problemi crittografici al fine di aggiungere nuovi blocchi alla catena. Il primo nodo che riesce a risolvere il problema ottiene il diritto di aggiungere il blocco e viene ricompensato con nuove criptovalute. PoW richiede un'enorme potenza di calcolo e consuma molta energia.

2. ***Proof of Stake (PoS)***: In PoS, la selezione del nodo che aggiunge un nuovo blocco alla catena è basata sulla quantità di criptovaluta che il nodo stesso possiede e blocca come "garanzia". Questo significa che più criptovaluta un nodo possiede, maggiori sono le probabilità che venga scelto per validare il blocco successivo. PoS richiede meno energia rispetto a PoW, ma può portare a una maggiore concentrazione di potere nelle mani di coloro che possiedono più criptovaluta.Un esempio di blockchain che utilizza il protocollo PoS è Ethereum [9]

3. ***Delegated Proof of Stake (DPoS):*** DPoS è una variante del protocollo PoS in cui i possessori di criptovalute delegano il loro potere di voto a un certo numero di nodi (generalmente

chiamati "delegati") per validare i blocchi al loro posto. I delegati sono selezionati in base ai voti ricevuti dagli utenti. L'algoritmo è considerato più efficiente in termini di scalabilità e velocità delle transazioni rispetto ai metodi PoW e PoS.

4. ***Proof of Authority (PoA):*** Nella PoA, il consenso è basato sull'identità dei nodi invece che sulla quantità di risorse o criptovaluta posseduta. Un gruppo selezionato di nodi di fiducia, noti come "autorità", viene autorizzato a validare i blocchi e mantenere la rete. PoA è spesso utilizzato in blockchain private o consorziate, in cui l'identità dei partecipanti è nota e controllata.

5. ***Practical Byzantine Fault Tolerance (PBFT):*** PBFT è un algoritmo di consenso progettato per reti blockchain con un numero limitato di nodi. Richiede un accordo tra i nodi partecipanti (generalmente due terzi dei nodi) per raggiungere il consenso. PBFT offre un alto grado di finalità delle transazioni e può raggiungere un consenso rapido, ma richiede un numero relativamente piccolo di partecipanti [47] [25].

2.3.2 Partecipazione alla blockchain

Esistono diversi tipi di blockchain che vengono distinti in base a come è possibile parteciparvi e al numero di nodi che ne fanno parte:

1. ***Blockchain pubbliche***: Queste sono blockchain accessibili a chiunque e non sono controllate da una singola entità. Tutti possono partecipare alla validazione delle transazioni e alla conservazione dei dati. Esempi di blockchain pubbliche includono Bitcoin [34], Ethereum [9] e Algorand [10].

2. ***Blockchain private***: Queste blockchain sono controllate da un'organizzazione o da un gruppo ristretto di entità. L'accesso e la partecipazione alla validazione delle transazioni sono limitati a queste entità. Le blockchain private sono utilizzate principalmente per scopi aziendali, in cui è necessario mantenere un controllo centralizzato. Ad esempio, Hyperledger Fabric [2] è una blockchain privata utilizzata in ambito aziendale.

3. ***Blockchain consortili***: Queste blockchain sono un ibrido tra le blockchain pubbliche e private. Sono controllate da un consorzio di organizzazioni, in cui ciascuna entità ha un certo grado di controllo e partecipazione. Le blockchain consortili consentono la

condivisione di dati tra le organizzazioni coinvolte, mantenendo comunque una certa forma di controllo centralizzato.

4. ***Blockchain permissioned***: Queste blockchain richiedono un'autorizzazione per partecipare alla validazione delle transazioni e alla conservazione dei dati. Sono utilizzate per garantire la conformità e il controllo centralizzato. Un esempio di blockchain *permissioned* è Corda [6]

5. ***Blockchain ibride***: Queste sono blockchain che combinano caratteristiche di blockchain pubbliche e private. Consentono la condivisione di dati pubblici e la gestione di dati privati all'interno di gruppi di partecipanti autorizzati. Le blockchain ibride offrono un equilibrio tra la trasparenza e la privacy dei dati

2.4 Smart Contract

Gli Smart Contract, termine che in italiano possiamo tradurre con "contratti intelligenti", sono dei software basati sulla blockchain, vengono utilizzati per automatizzare l'esecuzione di un accordo in modo che tutti i partecipanti possano essere immediatamente certi dell'esito, senza intermediari e senza perdite di tempo. In parole più semplici, se con i contratti tradizionali una parte viola i termini di un accordo, l'altra può portarla in tribunale, gli Smart Contract rafforzano tali accordi, in modo che le regole vengano applicate automaticamente senza che tribunali o terze parti siano chiamati in causa. Uno Smart Contract funziona seguendo il principio dell' "***if - then***" ovvero: se si verifica una determinata condizione allora effettua una determinata azione descritta dalla logica del programma. All'interno di uno Smart Contract dunque è possibile riportare tutte le clausole e i vincoli di un contratto legale come siamo abituati a intenderlo. I vantaggi degli Smart Contract coincidono con quelli più generici della blockchain, essi garantiscono:

1. ***Trasparenza***: Dal momento che non sono coinvolte terze parti e che i dati crittografati delle transazioni sono condivisi dai partecipanti, si riduce al minimo la possibilità che vengano manipolate le clausole del contratto a proprio vantaggio.

2. ***Sicurezza***: Poiché gli Smart Contract si basano sulla blockchain, garantiscono l'immutabilità dei dati consentendo di stipulare accordi senza il rischio di possibili violazioni o errori. Questa trasparenza fornisce alle parti sicurezza e fiducia.

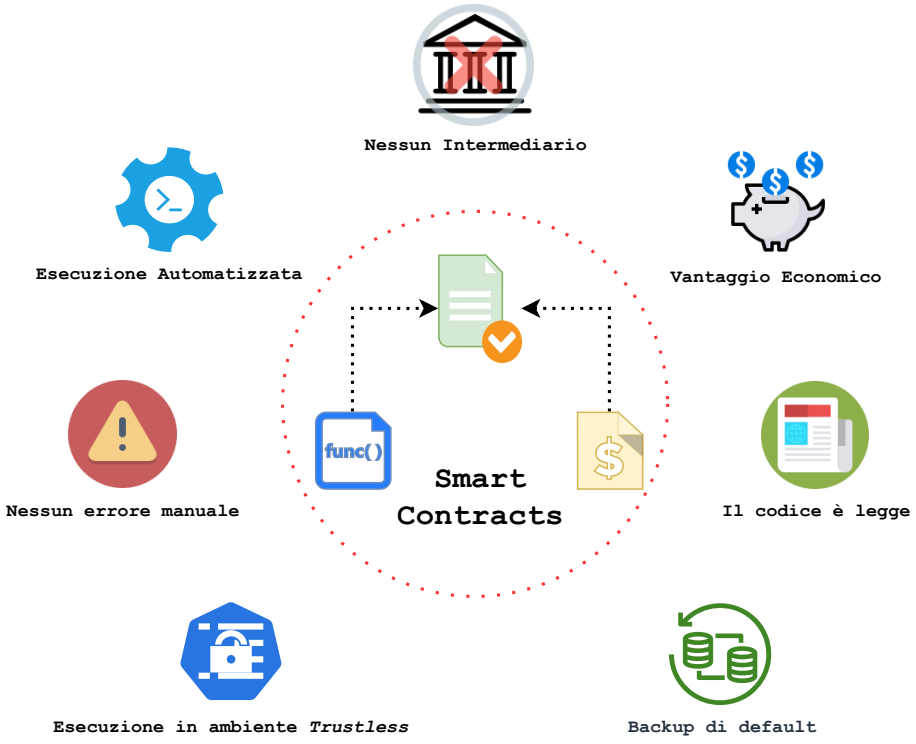

Figura 2.3: Caratteristiche Smart Contracts

3. ***Risparmio***: Negli Smart Contract l'eliminazione degli intermediari si traduce naturalmente in una riduzione dei costi. Non avendo bisogno di una terza parte per verificare i termini di contratto e assicurarne la validità, le commissioni e i costi associati alla presenza di uno o più intermediari scompaiono.

4. ***Velocità***: Il risparmio non è solo economico, ma anche di tempo. Niente intermediari, scartoffie, burocrazia e perdite di tempo per riconciliare gli errori dovuti spesso a errori umani: lo Smart Contract è digitale e automatizzato e, una volta soddisfatte le condizioni prestabilite, viene eseguito immediatamente.

2.4.1 Casi d'uso

Oggi gli Smart Contract sono un elemento tecnologico fondamentale di molte applicazioni decentralizzate (dApp), ossia quelle applicazioni che operano su un sistema computazionale distribuito, e l'uso e i potenziali vantaggi degli Smart Contract oltrepassano i confini della finanza, interessando settori e aziende di vario tipo. Alcuni esempi sono:

- ***Identità Digitale***: La blockchain, con gli Smart Contract, può aiutare il campo dell'identità digitale in termini di sicurezza e privacy, riducendo il rischio di frodi, violazioni e furti d'identità, e automatizzare il processo di creazione e condivisione dell'identità digitale, migliorando l'interoperabilità e la compliance [40]. Per esempio è stata implementato un esempio di gestore delle identità digitali sfruttando gli Smart Contract in [16]

- ***Assicurazioni***: I vantaggi degli Smart Contract nel settore delle assicurazioni sono evidenti: facilitano la stipula della polizza, possono ridurre i costi dell'assicurazione, migliorare l'esperienza d'uso dei prodotti assicurativi, e rendere più veloci pagamenti e rimborsi in caso di danni e sinistri [11].

- ***Aste di beni e servizi***: gli Smart Contract sono uno strumento fondamentale per l'implementazione di piattaforme di aste per beni e servizi. Combinati con gli NFT (Non Fungible Token), un tipo di token crittografico che rappresenta la proprietà o la prova di autenticità di un oggetto digitale o di un bene virtuale, rappresentano il migliore strumento per il trasferimento di beni tramite l'implementazione di un meccanismo di asta.

In conclusione gli Smart Contract rappresentano una opportunità per sfruttare un' innovazione tecnologica come la blockchain per trasportare i processi del mondo reale, che soffrono di problemi di trasparenza e affidabilità, in un contesto: sicuro, trasparente, certo, affidabile, senza intermediari, con costi altamente contenuti.

Capitolo 3

E-Auctions

In questo capitolo verrà effettuata una analisi dei principali metodi di asta classici, prendendo spunto dalla teoria delle aste, ramo della teoria dei giochi, inoltre verranno esaminate le soluzioni innovative dello stato dell'arte per quanto riguarda la realizzazione di piattaforme elettroniche per le aste. In fine si analizzerà il contesto delle piattaforme per aste intelligenti, ovvero quelle piattaforme che implementano le aste sfruttando la blockchain e gli Smart Contract.

3.1 Teoria delle Aste

Asta deriva dal latino ***"hasta"***. Presso gli antichi Romani una vendita pubblica era annunciata da un'hasta, simbolo di proprietà, che si piantava sul luogo del pubblico incanto e come segno della pubblica autorità. È proprio attraverso le aste che gli antichi Romani ripartivano il tesoro conquistato in guerra. In latino esistevano le seguenti espressioni: sub hastā vendĕre o hastae subicĕre (vendere all'incanto); hastam ponĕre ("piantare l'asta", cioè "annunciare una pubblica vendita"); ab hastā (acquisto all'incanto); ius hastae (diritto di vendita all'incanto). La teoria delle aste è una branca dell'economia che si occupa di studiare il funzionamento delle aste, ovvero dei meccanismi di vendita di beni e servizi attraverso un processo competitivo di offerte da parte di potenziali acquirenti. La teoria delle aste si occupa di analizzare gli incentivi degli offerenti, i loro comportamenti razionali e le strategie che possono adottare per massimizzare il proprio interesse economico durante un'asta. Studia anche il ruolo delle informazioni asimmetriche, ovvero quando alcuni offerenti dispongono d'informazioni privilegiate rispetto ad altri. La teoria delle aste ha diverse applicazioni pratiche, ad esempio nel settore delle privatizzazioni, delle

vendite all'asta di opere d'arte, delle assegnazioni di frequenze radio, delle licenze di utilizzo di risorse naturali e in molte altre situazioni in cui è necessario stabilire un meccanismo equo ed efficiente per allocare risorse tra potenziali acquirenti.

3.1.1 Tipi di Asta

Esistono quattro tipi principali di meccanismo di asta, e sono:

1. ***Asta in busta chiusa***: In cui gli offerenti dispongono la loro offerta in una busta sigillata e simultaneamente la passano al banditore. Le buste vengono aperte e l'individuo con la più alta offerta vince l'asta, pagando un prezzo pari all'ammontare offerto. La simultaneità temporale non è essenziale, ciò che conta è che quando uno fa la sua offerta, entro il termine fissato, non conosca le offerte fatte dagli altri.

2. ***Asta in busta chiusa al secondo prezzo***: il meccanismo di offerta e di vincita dell'asta è uguale a quello dell'asta con busta chiusa, con la sola differenza che il vincitore paga il prezzo della seconda offerta più alta.

3. ***Asta inglese***: meccanismo di asta per un solo bene dove è possibile trovarvi un numero di partecipanti molto elevato. L'asta è di tipo ascendente, cioè vince il prezzo massimo. Ogni partecipante ha una sua valutazione massima che costituisce la sua massima disponibilità a pagare il bene, il banditore, invece, comunica il prezzo di riserva, rappresentante il prezzo minimo richiesto per ottenere il bene messo all'asta. A ogni round il banditore comunica il prezzo del round precedente, aumentato di una quota che all'inizio è nota a tutti i partecipanti. Se il valore offerto dal partecipante è minore della quota comunicata, questi esce dall'asta e non può più rientrare: vince l'ultimo rimasto. Qualora uscissero tutti, si verificherebbe una situazione di parità che potrebbe essere risolta mediante un sorteggio random certificato da enti appositi.

4. ***Asta olandese***: In cui è fissato un prezzo sufficientemente alto a dissuadere tutti gli offerenti e viene progressivamente ridotto fino a che qualcuno non è disposto a comprare all'ultimo prezzo corrente. Il vincitore pagherà l'ultimo prezzo.

A questi si aggiungono dei meccanismi derivati come:

1. ***Asta a busta chiusa a ribasso***: il meccanismo di offerta è uguale a quello dell'asta a busta chiusa, ma l'offerta vincente è quella più bassa. Questo tipo di asta viene anche

detto "a ribasso", ed è spesso utilizzato per l'assegnazione di servizi a parte di aziende o pubbliche amministrazioni. Lo scopo di questo tipo di asta è d massimizzare il rapporto prezzo-servizio, facendo si che i concorrenti tendano a effettuare offerte più vantaggiose a ribasso.

2. ***Asta olandese inversa***: L'asta inizia con un'offerta iniziale molto bassa specificata dal banditore. Durante la fase di svolgimento dell'Asta, il prezzo dell'offerta viene poi aumentato dal Banditore fino a quando un fornitore accetta l'offerta, non appena un fornitore accetta un'offerta, l'asta termina.

3.2 E-Auction Systems

Un sistema di e-auction (asta elettronica) è un'infrastruttura online che consente a venditori e acquirenti di partecipare a un'asta virtuale. Questo sistema si basa su piattaforme di e-commerce che permettono ai partecipanti di presentare offerte e fare offerte su prodotti o servizi in modo completamente digitale, senza la necessità d'incontri fisici o di una sede fisica per l'asta.
Nel sistema di e-auction, i venditori pubblicano informazioni sul prodotto o servizio oggetto dell'asta, specificano le condizioni di vendita e stabiliscono un periodo di tempo entro il quale gli acquirenti possono fare offerte. Gli acquirenti, a loro volta, presentano le loro offerte attraverso la piattaforma e possono seguire l'andamento dell'asta in tempo reale. Al termine del periodo stabilito, l'offerta più alta o quella che soddisfa determinati criteri predefiniti viene selezionata come vincitrice dell'asta.
Al giorno d'oggi le principali tecnologie con le quali sono implementate le piattaforme per aste elettroniche sono il cloud e la blockchain.
Nel primo caso, l'architettura della piattaforma è centralizzata di tipo client/server, le operazioni sui dati e le transazioni vengono gestite da un server centrale, con la presenza di uno o più server di backup al fine di garantire la business continuity e di piani ben definiti per il disaster recovery [38]. Questa è una tencologia oggi consolidata, con la quale si è raggiunto un buon livello di sicurezza e di affidabilità.
Il secondo modello architetturale che prevede l'utilizzo della blockchain, si mette in netta contrapposizione con i modelli basati sul cloud. Si passa da una architettura centralizzata a una architettura distribuita peer-to-peer, dove le transazioni non sono gestite da un server centrale ma vengono approvate dalla rete e vi restano in maniera immutabile per sempre. Con l'utilizzo

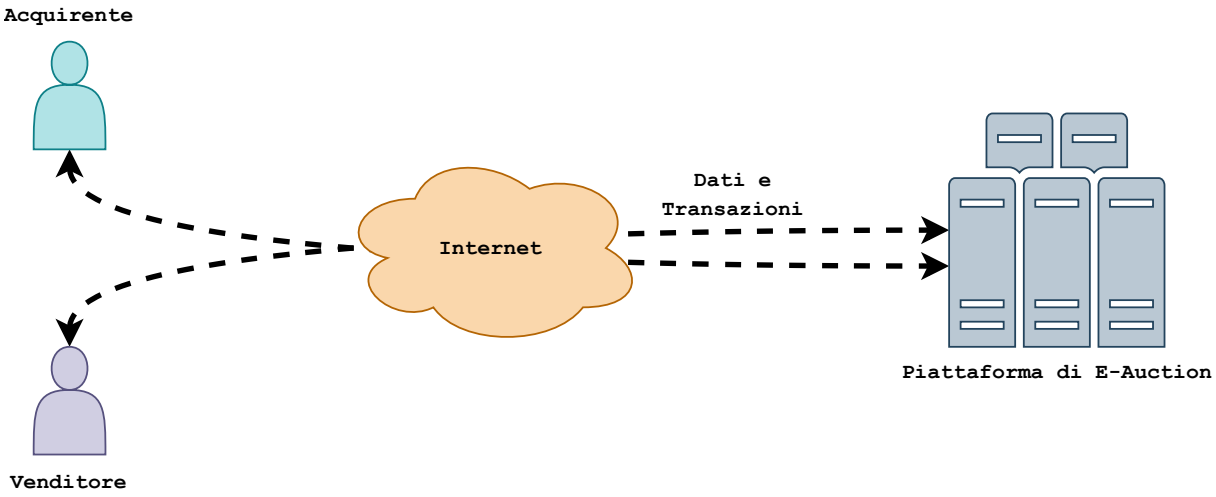

Figura 3.1: Schema generale di un sistema per la gestione di aste elettroniche basato su cloud

della blockchain come basi di dati distribuita sicura, si applica un concetto di sicurezza nuovo: *security by infrastracture*. Dunque sono la tecnologia e l'infrastruttura con cui si sviluppano i prodotti a garantire degli standard di privacy e sicurezza.

3.2.1 Smart Auctions

Le aste sono da sempre uno strumento fondamentale per la vendita di beni e servizi, ma negli ultimi anni si è assistito a un'evoluzione significativa grazie all'avvento delle aste intelligenti. Le Smart Auctions (aste intelligenti), rappresentano un nuovo approccio all'organizzazione e alla gestione delle aste, sfruttando le tecnologie digitali e, in particolare, la blockchain.
Queste si distinguono dalle aste tradizionali per la loro capacità di combinare l'efficienza e la trasparenza della tecnologia blockchain con la flessibilità e la praticità delle piattaforme online. Grazie a queste innovazioni, offrono numerosi vantaggi sia per i partecipanti che per gli organizzatori. Uno dei principali vantaggi è la trasparenza. Utilizzando la blockchain come base di dati distribuita e immutabile, ogni transazione viene registrata in modo sicuro e accessibile a tutti i partecipanti. Ciò garantisce che tutte le informazioni relative all'asta siano disponibili in tempo reale, eliminando così il rischio di manipolazioni o frodi.
Inoltre, le aste intelligenti offrono una maggiore sicurezza grazie all'utilizzo di contratti intelligenti. Ciò significa che gli offerenti possono partecipare all'asta con la certezza che le regole saranno rispettate senza la necessità di un intermediario di fiducia. I pagamenti e le transazioni sono gestiti in modo automatico e sicuro, riducendo o annullando il rischio di errori.
Un altro vantaggio delle smart auctions è la possibilità di raggiungere una vasta audience globale. Grazie alla natura digitale delle piattaforme online, le aste possono essere accessibili da qualsiasi parte del mondo, permettendo a un numero maggiore di partecipanti di competere per un oggetto o un servizio desiderato. Questo amplia notevolmente le opportunità per i venditori e offre una maggiore concorrenza, il che può portare a risultati migliori per tutti i partecipanti.
Ai vantaggi descritti in precedenza, si accompagnano una serie di sfide da superare come per tutte le nuove tecnologie. Riuscire a gestire la scalabilità delle e-auction basate su blockchain è una delle challenge maggiori, poiché questa può avere delle limitazioni sulla velocità e sulla capacità di elaborazione delle transazioni.
Inoltre, risulta di fondamentale importanza, riuscire a creare un modello che riesca a sfruttare la blockchain abbassando i costi, risultando conveniente anche economicamente.
In fine una sfida fondamentale è quella di unire la trasparenza della blockchain con la necessità di preservare la privacy dei dati delle aste, al fine di garantire un corretto svolgimento delle operazioni.
Molte delle soluzioni di e-auction su blockchain oggi sono realizzare con Ethereum [9]. Tramite l'uso degli Smart Contract è stato possibile realizzare: sia piattaforme general purpose per lo

svolgimento di aste di diversi tipi [36], che priattaforme di e-auction anticollusione bidder-to-bidder [48]
Le smart auction vengono implementate anche nel contesto smart grid ed energy trading, dove viene affidato agli smart contract il compito di condurre un asta doppia per il trading energetico [20].

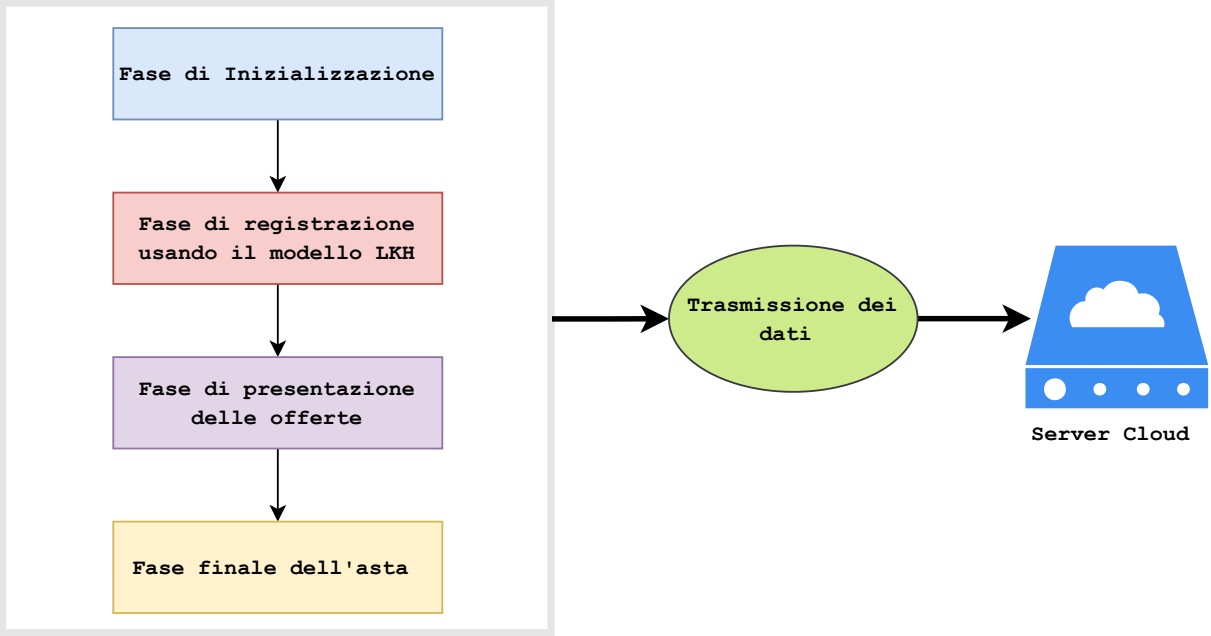

Figura 3.2: Schema a quattro fasi del sistema basato su cloud e LKH

3.3 Sicurezza e Privacy nelle piattaforme di E-Auction

Garantire che le operazioni durante lo svolgimento delle aste avvengano seguendo le norme di legge nelle piattaforme di e-auction, è uno degli obiettivi principali di chi progetta questi sistemi. L'utente nell'utilizzo di tali piattaforme, ha un approccio *"zero-trust"*, ovvero dà per scontato che non è possibile fidarsi ne di altri utenti che interagiscono con queste, ne di esse stesse. La sicurezza di questi sistemi consiste nel riuscire a creare uno spazio dove effettuare le transazioni in modo sicuro, affinché gli stakeholder abbiano la cognizione di operare in un contesto trasparente e senza alcuna possibilità di frode.

3.3.1 Modello con distribuzione logica e gerarchica delle chiavi

Al fine di raggiungere tale obiettivo, bisogna progettare dei sistemi in cui le comunicazioni con la piattaforma siano protette da crittografia. Un appproccio possibile è quello di gestire le chiavi secondo un modello *Logic Key Hierarchy* (LKH) [39]. Questo modello viene utilizzato per creare un sistema sicuro di data-sharing nelle e-auction con numero di membri dinamico, sfruttando *one-way hashing*, *reverse hashing*, generazione di un albero di chiavi e memorizza-

zione di queste in un server centralizzato. Questo sistema prevede lo scambio sicuro di dati in un sistema di aste elettroniche diviso in quattro fasi come mostrato in Figura 3.2.

Il server che gestisce le *e-auction* comunica con il gruppo di utenti **G** scambiando una chiave di gruppo **K** che viene generata nel server stesso a partire da due valori: un seme casuale **s** e un valore nascosto **x**. Da qui LKH, considerando come nodo radice il server, genera l'albero delle chiavi e memorizza nella radice tutti i valori generati [Figura 3.3]. In questo modo tutti i soli membri del gruppo *G* possono vedere le strategie dei vari *bidder* per una determinata asta. Un sistema che si basa su un modello LKH soffre di un problema che risiede nell'architettura del sistema stesso: il server centralizzato. Questo tipo di piattaforma prevede una architettura gerarchica dove nel server viene a configurasi un singolo punto di fallimento. L'utilizzo di una tecnologia distribuita risolverebbe questo problema, decentralizzando il sistema, garantendo che non vi sia un collo di bottiglia. Questo schema non evita la presenza di una figura terza rispetto venditori e partecipanti alle aste, ciò non garantisce che ilo svolgimento dell'asta possa avvenire escludendo la possibilità di collusione tra un gruppo di partecipanti e il gestore del servizio di *e-auction*.

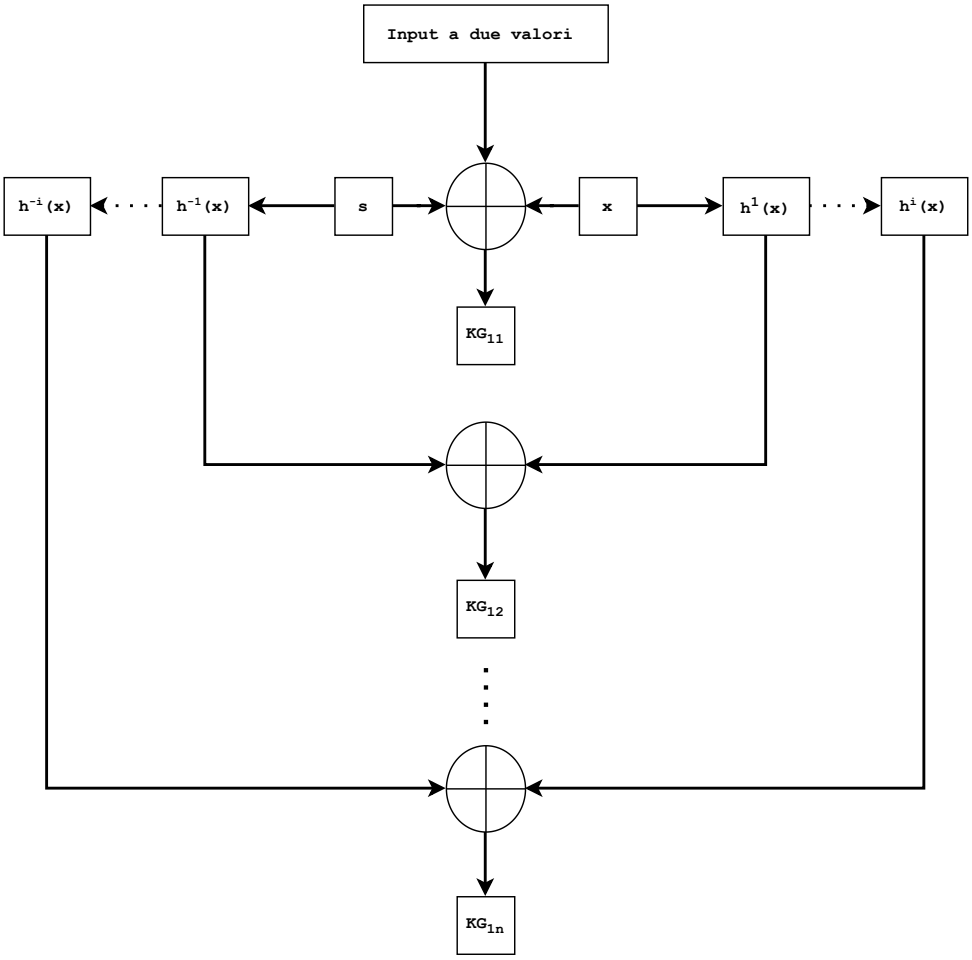

Figura 3.3: Generazione delle chiavi con LKH

3.3.2 Modello con cifrario asimmetrico e schema di firma digitale

Per garantire la sicurezza e la privacy dei dati, la correttezza dello svolgimento delle aste elettroniche, un approccio possibile è quello di utilizzare uno schema di cifratura asimmetrica accompagnato da uno schema di firma digitale [43]. Questo metodo se da un lato consente di raggiungere un buon livello di sicurezza e privacy nei dati, dall'altro non azzera l'ipotesi di collusione tra gestore dell'asta e un gruppo di *bidder*.
Un *bidder*, è colui che partecipa a un'asta, effettuando un'offerta.
Inoltre il metodo specificato in [43] prevede la generazione da parte degli utenti delle chiavi che vengono memorizzate dal sistema, creando dunque un singolo punto di fallimento. Inoltre l'uso di un algoritmo come RSA risulta avere un onere computazionale elevato se si pensa a

una piattaforma che viene utilizzata anche da dispositivi mobili. Uno schema crittografico che preveda l'uso di crittografia asimmetrica con curve ellittiche per la firma digitale e la crittografia simmetrica per la cifratura e la decifratura dei dati, risulta avere nel complesso un onere computazionale minore e una efficienza maggiore a parità di livello di sicurezza garantito. Dall'analisi di tale approccio emerge la necessità di una infrastruttura decentralizzata per la piattaforma di e-auction, che possa sfruttare i vantaggi delle cifratura asimmetrica, riuscendo a unire la trasparenza delle operazioni con la sicurezza e la privacy dei dati.

3.3.3 Modello con crittografia omomorfica

La crittografia omomorfica, è una tecnica avanzata di crittografia che consente di eseguire operazione matematiche su dei dati crittografati senza che vi sia la necessità di decifrarli. Esistono due grandi famiglie di tecniche di crittografia omomorfica: la crittografia parzialmente omomorfica (PHE), e la crittografia completamente omomorfica (FHE)
Si definisce uno schema omomorfico tra $\mathbb{M}$, insieme dei testi in chiaro e $\mathbb{C}$, insieme dei testi cifrati se [14]:

$$\forall m_1, m_2 \in \mathbb{M} : \mathbb{E}(m_1 \odot_{\mathbb{M}} m_2) \longleftarrow \mathbb{E}(m_1) \odot_{\mathbb{C}} \mathbb{E}(m_2) \tag{3.1}$$

Queste tecniche sono utilizzate in molti ambiti, dove la privacy dei dati che si elaborano è fondamentale, come per esempio l'elaborazione e lo storage di dati sanitari, l'elaborazione di dati finanziari e anche le piattaforme per le aste elettroniche.
In questo contesto è stato applicato in letteratura il *"il crittosistema omomorfico di Paillier"* [37]: schema di crittografia parzialmente omomorfica che supporta l'operazione di addizione sul testo cifrato.
Un sistema che implementa una asta a busta chiusa sfruttando questa tecnica di cifratura, utilizza nel suo flusso di lavoro una *bidding window* (finestra di offerta), ovvero una finestra temporale definita dal creatore dell'asta, dentro la quale è possibile per i partecipanti, effettuare le offerte. Inoltre nel sistema proposto sono presenti quattro entità principali:

1. *Piattaforma Intermediaria*: introdotta per evitare la possibilità di collusione tra gestore dell'asta e venditore o tra gestore dell'asta e un gruppo di *bidder*.

2. *Gestore dell'asta*: Per ogni richiesta di asta di un venditore genera una coppia di chiavi asimmetriche $(K_{\text{pub}}, K_{\text{prv}})$ per lo schema omomorfico di Paillier, e pubblica solamente la

chiave K_{pub}, mentre la K_{prv} è conservata solo da lui stesso in modo che né la piattaforma intermediaria, né i partecipanti all'asta possano decifrare il testo cifrato. Inoltre il gestore riceve le offerte cifrate con il *padding* aggiunto dalla IP durante la finestra di offerta e confronta i valori con *padding* per determinare il valore più alto senza conoscerne il valore effettivo.

3. *Venditore*: ha il compito di richiedere le aste al gestore, fornendo tutti i dati.

4. *Bidder*: colui che effettua le offerte.

3.3.4 Modello con Computazione Multiparte e Prove a conoscenza zero non interattive

Computazione Multiparte

La computazione multiparte, o MPC (Multiparty Computation), è un campo della crittografia e della sicurezza informatica che riguarda la collaborazione tra più entità o partecipanti per eseguire un calcolo senza rivelare le informazioni private di ciascuna parte coinvolta. In un protocollo di computazione multiparte, le parti coinvolte condividono i loro input e lavorano insieme per ottenere un output comune desiderato, mantenendo al contempo la riservatezza dei loro dati.
L'obiettivo principale della computazione multiparte è consentire la cooperazione sicura tra entità che non si fidano completamente l'una dell'altra, senza dover rivelare le loro informazioni sensibili. Ciò significa che ogni parte coinvolta nel protocollo deve essere in grado di ottenere solo le informazioni che sono necessarie per la computazione, senza conoscere gli input specifici delle altre parti.
I protocolli di computazione multiparte possono essere utilizzati in diversi contesti, come l'elaborazione di dati sensibili, le aste elettroniche, la firma digitale distribuita e molto altro.
Ogni protocollo basato su MPC si può ritenere sicuro se rispetta le seguenti proprietà [27]:

- ***Privacy***: Nessuna parte dovrebbe apprendere nulla più del suo output previsto. In particolare, l'unica informazione che dovrebbe essere appresa sugli input delle altre parti è ciò che può essere dedotto dall'uscita stessa. Ad esempio, in un'asta in cui viene rivelata solo l'offerta del miglior offerente, è possibile dedurre chiaramente che tutte le altre offerte era-

no inferiori all'offerta vincente. Tuttavia, non dovrebbe essere rivelato nulla sulle offerte perdenti.

- ***Correttezza***: Ogni parte ha la garanzia che l'output che riceve sia corretto. Per continuare con l'esempio di un'asta, ciò implica che la parte con l'offerta più alta ha la garanzia di vincere e nessuna parte, compreso l'organizzatore dell'asta, può influenzare questo risultato.

- ***Indipendenza degli input***: Le parti corrotte devono scegliere i loro input indipendentemente dagli ingressi delle parti oneste. Questa proprietà è cruciale in un'asta sigillata, in cui le offerte sono mantenute segrete e le parti devono fissare le loro offerte in modo indipendente dalle altre. Si nota che l'indipendenza degli input non è implicata dalla privacy. Ad esempio, potrebbe essere possibile generare un'offerta più alta senza conoscere il valore dell'offerta originale. Un tale attacco può effettivamente essere eseguito su alcuni schemi di crittografia.

- ***Garanzia di consegna dell'output***: Le parti corrotte non dovrebbero essere in grado d'impedire alle parti oneste di ricevere il loro output. In altre parole, l'avversario non dovrebbe essere in grado d'interrompere il calcolo tramite un attacco *denial of service*.

- ***Equità***: Le parti corrotte dovrebbero ricevere i loro output solo se anche le parti oneste li ricevono. Non dovrebbe essere consentito che si verifichi lo scenario in cui una parte corrotta ottiene l'output mentre una parte onesta no.

Prove a conoscenza zero non interattive

Dall'introduzione negli anni '80, le prove a conoscenza zero (ZKPs) sono diventate un elemento fondamentale nella crittografia moderna [18], trovando ampia applicazione come componente chiave in diverse costruzioni crittografiche. Queste includono schemi d'identificazione, schemi di firma di gruppo, credenziali anonime, sistemi di voto e computazione sicura [46].
Nel contesto in cui un dimostratore P cerca di dimostrare a un verificatore V la sua conoscenza di una soluzione per un problema complesso, si può ricorrere a una prova a conoscenza zero per evitare che V possa ingannare P e scoprire la soluzione o qualsiasi informazione che potrebbe agevolarlo nel calcolo più rapido. Pertanto, ciò che V può apprendere tramite il protocollo dovrebbe essere limitato a ciò che potrebbe dedurre autonomamente.

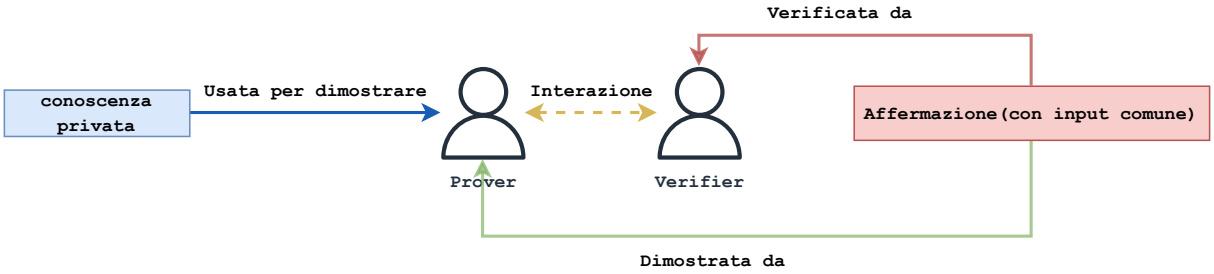

Figura 3.4: Schema generale sistema di prove a conoscenza zero

Una prova a conoscenza zero deve soddisfare tre importanti proprietà, note come completezza, correttezza e conoscenza zero. La completezza garantisce che per ogni input valido, il dimostratore P possa sempre completare con successo la prova; la correttezza assicura che nessun dimostratore malintenzionato P possa costruire un sistema di prova valido; e la conoscenza zero implica che nessun verificatore malintenzionato V possa ottenere conoscenze aggiuntive dall'interazione [30].
Formalmente uno schema ZKPs è definito come segue:

> Sia$\{0,1\}*$ l'insieme di tutte le stringhe e R una testimonianza. Consideriamo un linguaggio $L \subseteq \{0,1\}*$.
> Definiamo un sistema di prova a conoscenza zero non interattivo per il linguaggio L come una coppia di macchine di Turing probabilistiche (P,V) , in cui P ha la capacità di calcolo probabilistico polinomiale e V ha la capacità di calcolo deterministico polinomiale.
> Tale sistema soddisfa le seguenti condizioni di correttezza e sicurezza nei confronti di dimostratori e verificatori malevoli:
>
> 1. Completezza : $\forall\ p(\cdot)$ ed un input comune $x \in L$, x è accettata da V con un probabilità maggiore di 1 -ε:
>
> $$Pr[V(x,R,P(x,R) = 1)] \geqslant 1 - \frac{1}{p(|x|)} \tag{3.2}$$
>
> 2. Correttezza : $\forall$ macchina di Touring interattiva P' che rappresenta un dimostratore disonesto, ogni polinomio $p(\cdot)$ e qualsiasi input comune $x \in L$ fornito

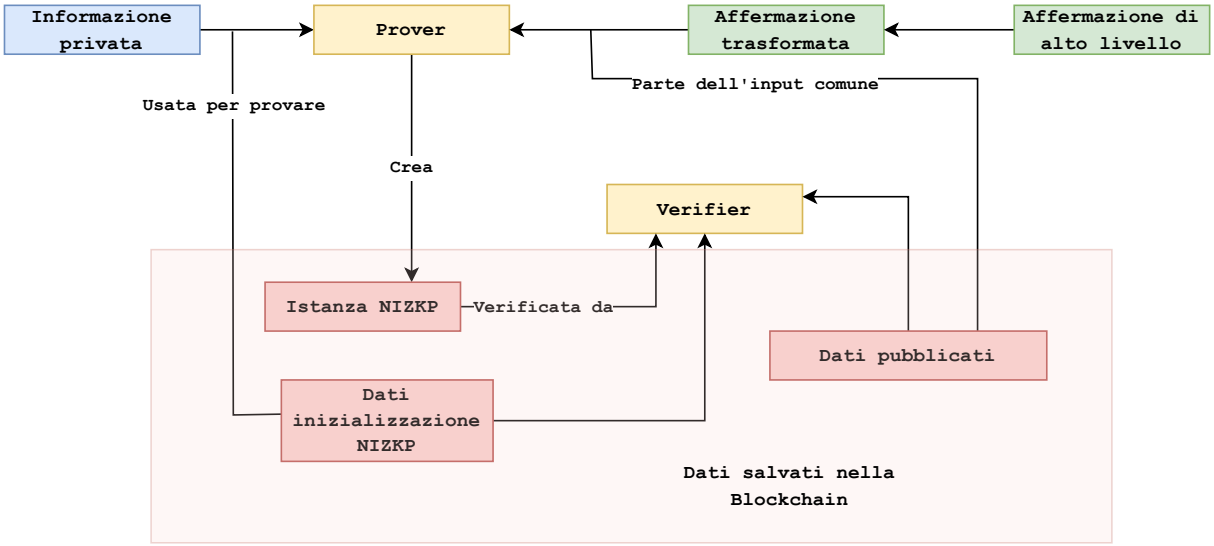

Figura 3.5: Schema generale NIZKPs con uso di Blockchain

da P', x viene accettato da V con una probabilità al più di ε:

$$Pr[V(x,R,P'(x,R)=1)] < \frac{1}{p(|x|)} \tag{3.3}$$

3. Conoscenza zero: $\forall x \in L$ fornito da P, nessuna informazione viene rivelata da x a V, che V non potrebbe calcolare da solo prima dell'esecuzione del protocollo, il che significa che esiste un algoritmo di tempo polinomiale probabilistico M tale che:

$$V(x) = x, (R \in 0,1^{c(|x|)}, P(x,R)) \approx_c M(x)_{x\in L} \tag{3.4}$$

I meccanismi *"Non-Interactive-Zero-Knowledge-Proofs (NIZKPs)"*, affondano le basi sullo stesso concetto principale di ZKPs, ma non prevedono alcuna interazione tra il *prover* e il *verifier*.
I principali algoritmi NIZKPs sono:

- ***zk-SNARKs*** : Zero Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARKs).La dimensione della prova e il tempo di verifica sono inferiori alla dimensione lineare dell'input o addirittura costanti. Alcuni zk-SNARKs utilizzano configurazioni fidate come Sonic [29], mentre più recentemente sono stati creati zk-SNARKs senza configurazioni fidate (configurazioni trasparenti o anche utilizzabili in modo interattivo) come Ligero [1].

- ***zk-STARKs***: Zero Knowledge Scalable Transparent Arguments of Knowledge (zk-STARKs). Utilizzano le funzioni hash crittografiche per garantire la sicurezza.
- ***Bulletproofs***: Bulletproofs [7] è un meccanismo NIZKP, utilizzato per dimostrare che un segreto $v \in \mathbb{Z}_p$ si trova in un determinato intervallo $[0, 2^{n-1}]$. Come assunzione di sicurezza si basa sul problema del logaritmo discreto.

Questi meccanismi combinati con la MPC sono alla base di molti sistemi di e-auction basati su blockchain [15] [26] [28].
Le dimostrazioni a conoscenza zero rappresentano un metodo unico per verificare la veridicità delle informazioni preservando la privacy, ma non forniscono una sicurezza assoluta. Inoltre, gli algoritmi utilizzati richiedono risorse computazionali intense. In alcuni tipi di ZKP, il calcolo intensivo è necessario perché richiede molte interazioni tra *verifier* e *prover*. In altri, gli algoritmi sono estremamente intensi dal punto di vista computazionale, cosa che potrebbe limitare le applicazioni delle ZKP.

Capitolo 4

Analisi Blockchain Algorand

Nei seguenti paragrafi di questo capitolo, viene introdotta e appofondita la blockchain Algorand, chè è stata scelta come blockchain sulla quale sviluppare il sistema di aste online, in quanto questa garantisce al contempo efficienza, velocità e bassi costi. oltre ad essere una *green blockchain*. Verrà esposto il meccanismo di funzionamento degli smart contract in questa specifica blockchain.

4.1 Algorand Blockchain

Algorand nasce dall'idea di Silvio Micali e il suo team di proporre un nuovo metodo per implementare un registro pubblico che offrisse la comodità e l'efficienza di un sistema centralizzato gestito da un'autorità fidata e inviolabile, senza le inefficienza e le debolezze delle attuali implementazioni decentralizzate [10]. Algorand presenta un approccio democratico, nel senso che ne in principio ne de facto crea diverse classi di utenti, come per esempio accade in Bitcon, al contrario in Algorand il potere risiede nell'insieme di tutti gli utenti. La blockchain sviluppata da Micali supera i problemi che si riscontrano con Bitcoin, proponendo una soluzione dove non vi è: spreco di calcolo, concentrazione di potere e ambiguità. L'obiettivo è rompere il trilemma della blockchain, che afferma:

"Non è possibile avere una blockchain che sia decentralizzata, scalabile e sicura"

4.2 Rounds

Algorand è organizzato in unità logiche, $r = 0, 1, \ldots$, chiamate round. [1] All'inizio del round $r > 0$, l'insieme di tutte le chiavi pubbliche è PK_r, e lo stato del sistema è $S_r = n^{i,a(r)}, \ldots : i \in PK_{r0}, i$ dove $a(r)$ è la quantità di denaro disponibile per la chiave pubblica i. Notare che PK_r è deducibile da S_r, e che S_r può specificare anche altri componenti per ogni chiave pubblica i. Per il round 0, PK_0 è l'insieme delle chiavi pubbliche iniziali, e S_0 è lo stato iniziale. Sia PK_0 che S_0 sono considerati conoscenza comune nel sistema. Per semplicità, all'inizio del round r, anche $PK_1, \ldots, PK_r$ e $S_1, \ldots, S_r$ sono considerati conoscenza comune. In un round r, lo stato del sistema passa da S_r a S_{r+1}: simbolicamente, Round r: $S_r \rightarrow S_{r+1}$.

4.3 Blocchi

In Algorand, il blocco B_r corrispondente a un round r specifica: il round stesso r; l'insieme dei pagamenti del round r, PAY_r; una quantità Q_r, da spiegare, e l'hash del blocco precedente, $H(B_{r-1})$. Pertanto, partendo da un blocco fisso B_0, otteniamo una blockchain tradizionale: $B_1 = (1, PAY_1, Q_0, H(B_0))$, $B_2 = (2, PAY_2, Q_1, H(B_1))$, $B_3 = (3, PAY_3, Q_2, H(B_2))$, ... In Algorand, l'autenticità di un blocco è effettivamente garantita da un'informazione separata, un "certificato di blocco" $CERT_r$, che trasforma B_r in un blocco provato, B_r. La *Magic Ledger* è quindi implementata dalla sequenza dei blocchi provati, $B_1, B_2, \ldots$

4.4 Account

Gli account rappresentano entità presenti sulla blockchain di Algorand che sono associate a dati specifici presenti sulla catena, come ad esempio un saldo. Un indirizzo di Algorand, invece, funge da identificatore per un account sulla rete di Algorand.
Dopo aver generato una chiave privata e l'indirizzo corrispondente, l'invio di Algos (la criptovaluta di Algorand) all'indirizzo sulla rete di Algorand avvierà il processo d'inizializzazione dello stato dell'account sulla blockchain di Algorand.

[1] Utilizziamo costantemente gli apici per indicare i round. Per indicare che una quantità non numerica Q (ad esempio, una stringa, una chiave pubblica, un insieme, una firma digitale, ecc.) si riferisce al round r, scriviamo semplicemente Q_r. Solo quando Q è un vero numero (in contrapposizione a una stringa binaria interpretabile come numero), scriviamo $Q(r)$, in modo che il simbolo r non possa essere interpretato come l'esponente di Q.

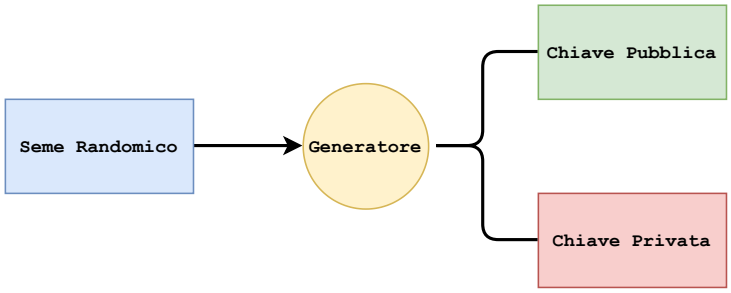

Figura 4.1: Generazione coppia di chiavi Pubblica e Privata in Algorand

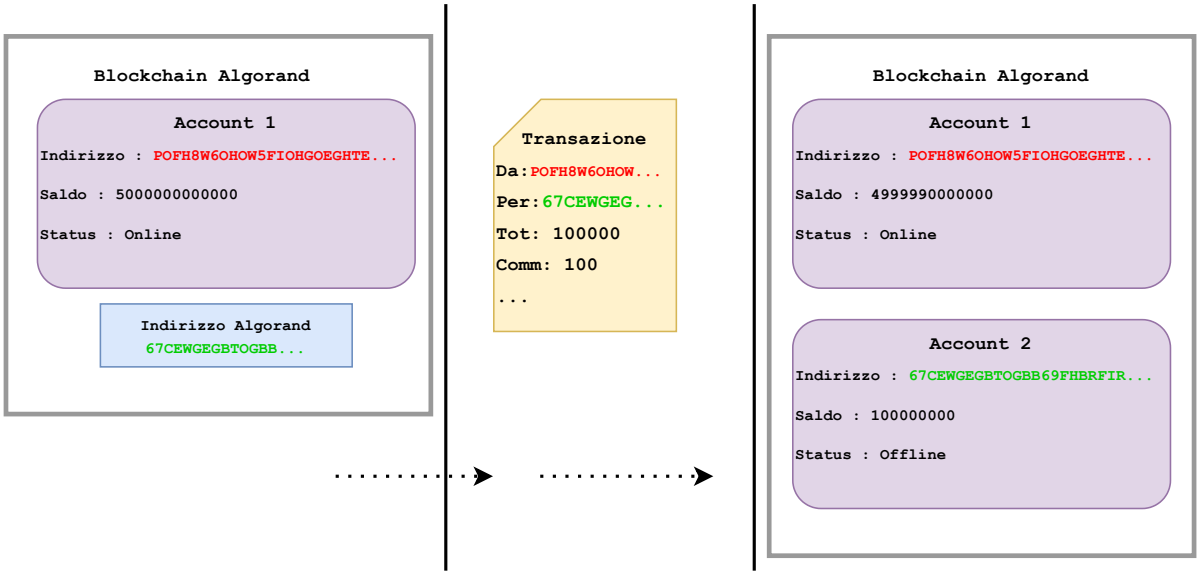

Figura 4.2: Inizializzazione Account Algorand

Questo meccanismo d'inizializzazione dello stato è essenziale per garantire l'integrità e la corretta gestione degli account sulla rete. Una volta che i fondi vengono inviati all'indirizzo associato all'account, il suo stato verrà registrato sulla blockchain di Algorand, consentendo agli utenti di eseguire transazioni e interagire con l'account in modo sicuro e affidabile.

L'utilizzo di una chiave privata per generare un indirizzo univoco e associarlo all'account rappresenta un importante aspetto della sicurezza e della privacy nella rete di Algorand. La generazione casuale e crittograficamente sicura della chiave privata garantisce che solo il proprietario dell'account abbia accesso e controllo sui fondi e sui dati associati a quell'indirizzo specifico. Dunque la creazione di un account Algorand consiste nella generazione di una coppia di chiavi $(K_{\text{PUB}}, K_{\text{PRV}})$. Algorand utilizza le firme a curva ellittica Ed25519 ad alta velocità e

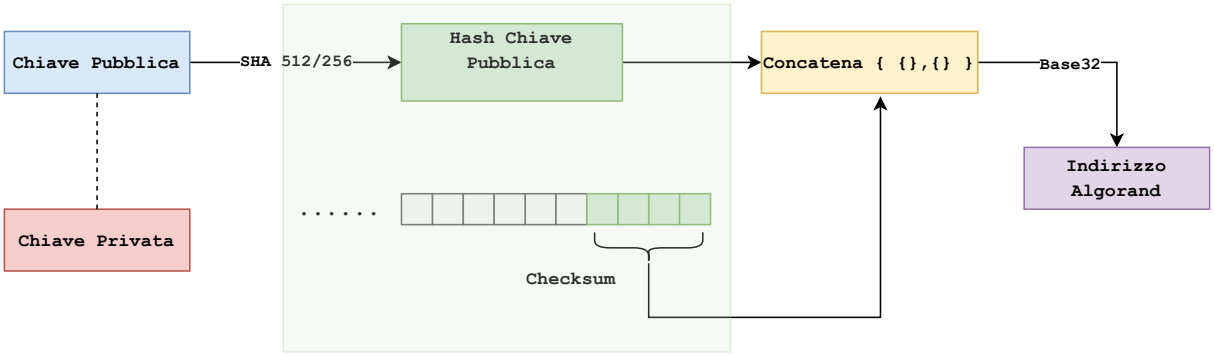

Figura 4.3: Trasformazione chiave pubblica in indirizzo Algorand

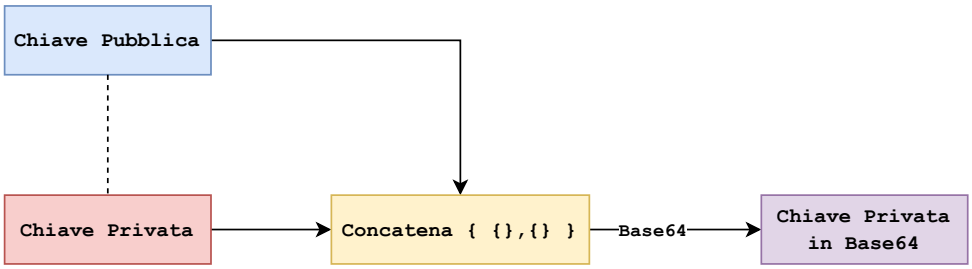

Figura 4.4: Trasformazione chiave privata in codifica base64

alta sicurezza. Per generare una coppia di chiavi viene preso un seme randomico e dato in pasto a un generatore che genera una coppia di array di 32 byte. Queste chiavi svolgono importanti funzioni crittografiche come la firma dei dati e la verifica delle firme. Al fine di rendere queste chiavi più leggibili per l'uomo e maggiormente resistenti all'errore umano quando vengono trasferite, entrambe le chiavi pubblica e privata subiscono delle trasformazioni.

La chiave pubblica viene trasformata in quello che è chiamato **Indirizzo Algorand**, aggiungendo una *checksum* di quattro byte alla fine della chiave pubblica e codificando il tutto in codifica **base32**. Il risultato è un indirizzo lungo 58 caratteri [Figura 4.3].

La chiave privata invece viene può essere rappresenta in due modi diversi: codifica **base64**, **25-word mnemonic**.

La codifica in base64 è la codifica utilizzata maggiormente dagli sviluppatori che si interfacciano maggiormente con gli SDKs[2]. Vengono prima concatenate la chiave privata e la chiave pubblica e successivamente ne viene fatta una codifica in base64 [Figura 4.4] Il mnemonico di 25 parole rappresenta una soluzione estremamente intuitiva e facilmente comprensibile per rappresentare

[2]SDKs = Software Development Kits

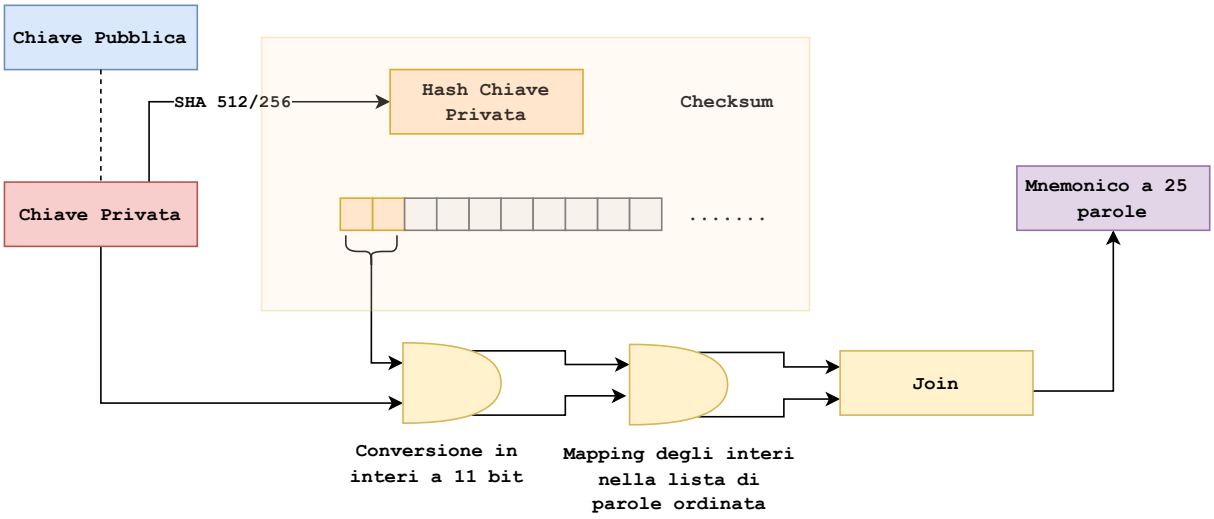

Figura 4.5: Trasformazione chiave privata in mnemonico 25 parole

una chiave privata. Questa rappresentazione viene ottenuta attraverso un processo di conversione dei byte della chiave privata in interi a 11 bit e successivamente associando a ciascun intero una parola appartenente alla lista di parole inglesi definita dallo standard bip-0039. In particolare, ogni intero n viene mappato alla parola situata nella *n*-esima posizione della suddetta lista.
Questo procedimento genera originariamente un mnemonico composto da 24 parole. Tuttavia, per garantire l'integrità e la sicurezza del mnemonico, viene aggiunto un checksum. Tale checksum viene ottenuto prelevando i primi due byte dell'hash della chiave privata e convertendoli anch'essi in interi a 11 bit. Successivamente, viene associata a questi interi la corrispondente parola della lista di parole.
Infine, questa parola ottenuta attraverso il checksum viene aggiunta alla fine del mnemonico di 24 parole, creando così un mnemonico di 25 parole completo e sicuro [Figura4.5].

4.5 L' algoritmo di consenso

La blockchain di Algorand utilizza un algoritmo di Accordo Bizantino decentralizzato che sfrutta il puro proof of stake (Pure POS). Ciò significa che può tollerare utenti maligni, raggiungendo un consenso senza un'autorità centrale, purché la maggioranza assoluta del capitale sia in mani non maligne. Questo algoritmo è molto veloce e richiede una potenza di calcolo minima per nodo, conferendogli la capacità di finalizzare le transazioni in modo efficiente.
Prima di procedere con la descrizione dell' algoritmo di consenso, vengono introdotti alcuni punti cardine che aiutano a capire il funzionamento dell'algoritmo nel suo complesso.

4.5.1 Verifiable Random Function

Le ***Verifiable Random Functions*** (VRF), pubblicate da Micali, Rabin e Vadhan nel 1999 [31], rappresentano un concetto fondamentale nell'ambito della crittografia. Prima dell'introduzione delle VRF, le funzioni pseudorandom erano conosciute come *"pseudorandom oracle"* e presentavano limitazioni significative. Senza informazioni sul seme iniziale e sul valore x, non era possibile distinguere l'output z dalla generazione casuale di una stringa della stessa lunghezza. Di conseguenza, non era possibile verificare il calcolo eseguito e si doveva fare affidamento sulla fiducia che fosse stato correttamente generato a partire da un determinato seme.
Le ***Verifiable Random Functions***, invece, richiedono una chiave pubblica pk per ogni utente. Questa chiave consente di provare la correttezza del calcolo del valore di $f(x)$ a chiunque sia interessato a verificarlo. In altre parole, grazie alle VRF è possibile dimostrare in modo verificabile che il risultato è stato generato correttamente a partire da un certo seme. Questo rappresenta un notevole vantaggio rispetto alle funzioni pseudorandom precedenti, poiché introduce un meccanismo di verifica che permette di accertare la validità dell'output senza dover dipendere esclusivamente dalla fiducia.
In maniera formale:

> *Una funzione VRF richiede una coppia di chiavi (pk, sk), rispettivamente chiave pubblica e privata, tale che solo chi possiede la chiave privata sk può calcolare l'hash in un α*
>
> $$\beta = F_{\text{sk}}(\alpha) \tag{4.1}$$
>
> *mentre tutti coloro che vedono la chiave pubblica pk devono essere in grado di verificarla.*

La proprietà di essere pseudorandom garantisce che β sia non distinguibile da un'altra stringa casuale. Solo chi possiede la chiave privata sk può farlo e tale prova va mostrata:

$$\pi = \prod_{\text{sk}}(\alpha) \tag{4.2}$$

Algorand adotta una VRF costruita sulla curva ellittica Curve25519 :
Sia $\mathbb{Z}_{\text{p}}$, un gruppo ciclico di ordine p primo, sia g un generatore di $\mathbb{Z}_{\text{p}}$. La tupla $(p, g, \mathbb{Z}_{\text{p}})$ rappresenta i parametri pubblici della curva.
Sia H la funzione hash che mappa stringhe di bit di lunghezza arbitraria nel gruppo $\mathbb{Z}_{\text{p}}$.
La chiave segreta è un numero $x \in \mathbb{Z}_{\text{p}}$, mentre la chiave pubblica è la coordinata x del punto sulla curva $P = xG$, con G punto base $= (9, y_{\text{G}})$ della curva. Di seguito viene mostrato l'algoritmo della VRF, gli algoritmi **VRFGen** e **VRFVerify**:

Algorithm 1 VRF Protocol

1: **procedure** VRFGEN(α,p, $\mathbb{Z}_{\text{p}}$,g, h)
2: Trova l'elemento del gruppo $\mathbb{Z}_{\text{p}}$ con la funzione hash $h(\alpha)$ e sia H il corrispettivo punto sulla curva
3: Calcola il punto della curva $\Gamma = xH$
4: Scegli un nonce $k \in \mathbb{Z}_{\text{p}}$
5: Calcola $c = h(G, H, xG, xH, kG, kH)$
6: Sia $s = k - c \cdot x \mod p$
7: **return** $(prova, hash)$: $\pi = (\Gamma, c, s)$; $\beta = H_2(\Gamma)$

H_2 restituisce in output la coordinata x del punto sulla curva ellittica.
Ognuno dei punti sulla curva individuati viene espresso con una compressione a 256 bit, rappresentando solo l'ascissa più un singolo bit che indica se l'ordinata è positiva o negativa.

Algorithm 2 VRF Protocol

procedure VRFVERIFY(α,xG,π, h)
    Calcolare $u = c(xG) + sG$, con xG chiave pubblica. Notare che $kG = u$.
    Dato α input, calcolare $h(\alpha)$ e H come in **VRFGen**
    Calcolare $v = c\Gamma + sH$ considerando sempre solo la coordinata delle ascisse.
    Verificare che $c = h(g, H(\alpha), xG, \Gamma, c(xG) + sG, c\Gamma + sH) \longrightarrow V_1 = True$
    Verificare che $\beta = H_2(\Gamma) \longrightarrow V_2 = True$
    if V1 and V2 **then**
        return True
    return False

4.5.2 Procedura operativa

Algorand genera un nuovo blocco mediante l'impiego di un nuovo algoritmo crittografico di accordo bizantino basato sulla comunicazione di messaggi, il quale soddisfa molteplici proprietà aggiuntive pur garantendo celerità. In sintesi,l'algoritmo si compone di un ciclo a tre fasi, in cui un partecipante i invia un singolo messaggio m_i a tutti gli altri partecipanti. Espletato all'interno di una rete completa e sincrona, con oltre i $\frac{2}{3}$ dei partecipanti classificabili come onesti e una probabilità superiore a $\frac{1}{3}$, l'algoritmo giunge a un accordo dopo ogni ciclo. Algorand sfrutta questo algoritmo binario di accordo bizantino per raggiungere un consenso, all'interno di un modello di comunicazione specifico, su ciascun nuovo blocco. Il blocco convenuto viene successivamente certificato mediante un numero prescritto di firme digitali da parte dei verificatori adeguati e diffuso nell'intera rete.

Ordinamento Crittografico

Al fine di evitare un problema legato alla concentrazione di potere, a ogni nuovo blocco B_r viene attribuito un insieme separato di verificatori selezionati, attraverso l'esecuzione di un nuovo algoritmo di Accordo Bizantino. La selezione di tali verificatori potrebbe essere altrettanto complessa quanto la scelta del blocco B_r stesso. Per superare questa problematica, viene adottato un approccio denominato *"Cryptographic Sortition"*, che si basa sulla pratica di selezionare in modo casuale funzionari da un vasto insieme d'individui idonei. In un sistema decentralizzato, l'estrazione casuale delle monete necessarie per selezionare i verificatori, tra l'intera popolazione degli utenti, deve essere automatica e casuale. Fondamentalmente, si

utilizza una funzione crittografica che, partendo dal blocco precedente B_{r-1}, determina in modo automatico un utente "leader" responsabile della proposta del nuovo blocco B_r e l'insieme di verificatori SV_r incaricato di raggiungere l'accordo sul blocco proposto dal leader. Per garantire l'imprevedibilità nella selezione del leader e dei verificatori, si impiegano input aggiuntivi costruiti appositamente e si ricorre alla VRF descritta in 4.5.1.

Modello di selezione

La selezione degli utenti avviene in proporzione al loro capitale, considerando ciascuna unità di Algorand come un "sottoutente" distinto. Se un utente i possiede w^i unità di Algorand, viene simulato un utente (i, j) con $j \in 1, ..., w^i$ che rappresenta la j-esima unità di valuta posseduta da i e viene selezionato con una probabilità $p = \frac{r}{W}$, dove W è il totale delle unità di valuta in Algorand. La *sortition* viene eseguita calcolando $hash, \pi \leftarrow \text{VRF}_{sk}(\text{seed}|\text{role})$, dove sk è la chiave segreta dell'utente.
L'hash pseudo-casuale determina il numero di sottoutenti selezionati. L'algoritmo di ordinamento divide l'intervallo $[0, 1)$ in intervalli consecutivi della forma

$$I^j = [\sum_{k=0}^{j} P(k), \sum_{k=0}^{j+1} P(k)] \tag{4.3}$$

Se:

$$2^{\frac{hash}{hashlen}} \in [\sum_{k=0}^{j} P(k), \sum_{k=0}^{j+1} P(k)] \tag{4.4}$$

l'utente avrà esattamente j sottoutenti selezionati. La verifica pubblica del numero di sottoutenti selezionati avviene utilizzando la prova π derivata dall'output della funzione crittografica di verificabilità casuale (VRF).

Block Proposal

Dopo aver selezionato un insieme di partecipanti, ciascuno di essi presenterà, insieme alla prova di selezione, un blocco di transazioni da aggiungere alla catena. Tuttavia, la scelta tra le varie proposte deve essere effettuata in modo univoco e nel minor tempo possibile da parte dell'intera rete. Per raggiungere questo obiettivo, viene ancora una volta impiegato il *sortition*

[3] `https://developer.algorand.org/docs/get-details/algorand_consensus/`

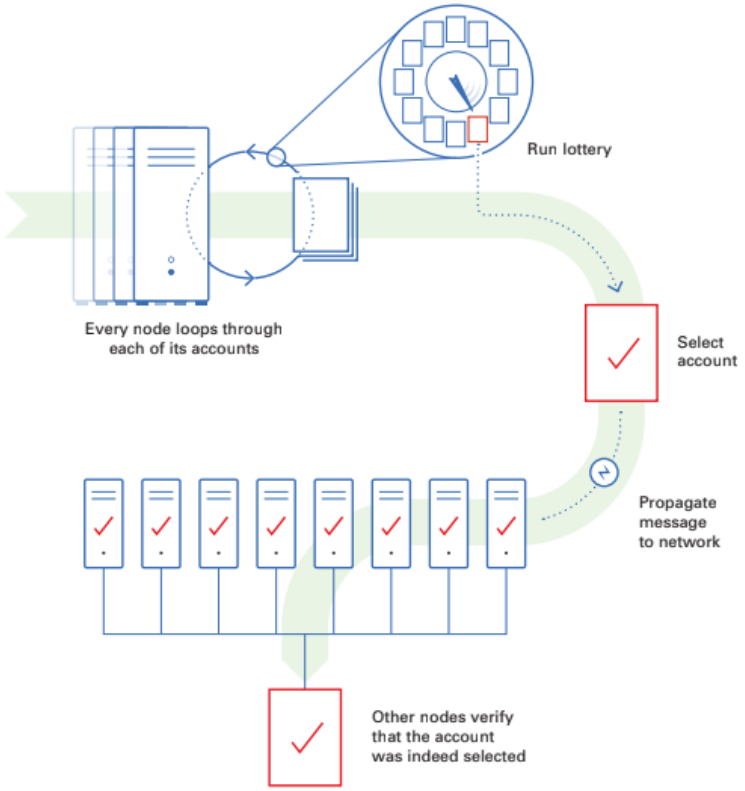

Figura 4.6: Fase Block Proposal[3]

hash: per ogni sub-utente vincente di un dato utente i, la priorità di un blocco viene calcolata facendo l'hash dell'output $(hash)$ del VRF concatenato con l'indice j del sub-utente. L'utente che propone un blocco con il valore di priorità più elevato diventa il leader e, dopo opportuni controlli di verifica, la sua proposta viene aggiunta alla catena.
Al fine di evitare un sovraccarico della rete, le informazioni vengono divise in due messaggi: il primo contiene la prova di selezione e del blocco proposto, mentre il secondo messaggio contiene l'intero blocco. In questo modo, l'informazione si propaga più velocemente selezionando il leader e solo successivamente attingendo alla seconda comunicazione.
Un aspetto rilevante da sottolineare è il tempo di attesa che ogni utente deve rispettare prima di convalidare un blocco. È evidente che impostare questo valore troppo basso potrebbe portare alla convalida di blocchi vuoti con una certa frequenza. D'altro canto, un tempo di attesa eccessivamente lungo potrebbe causare problemi all'algoritmo a discapito della scalabilità.

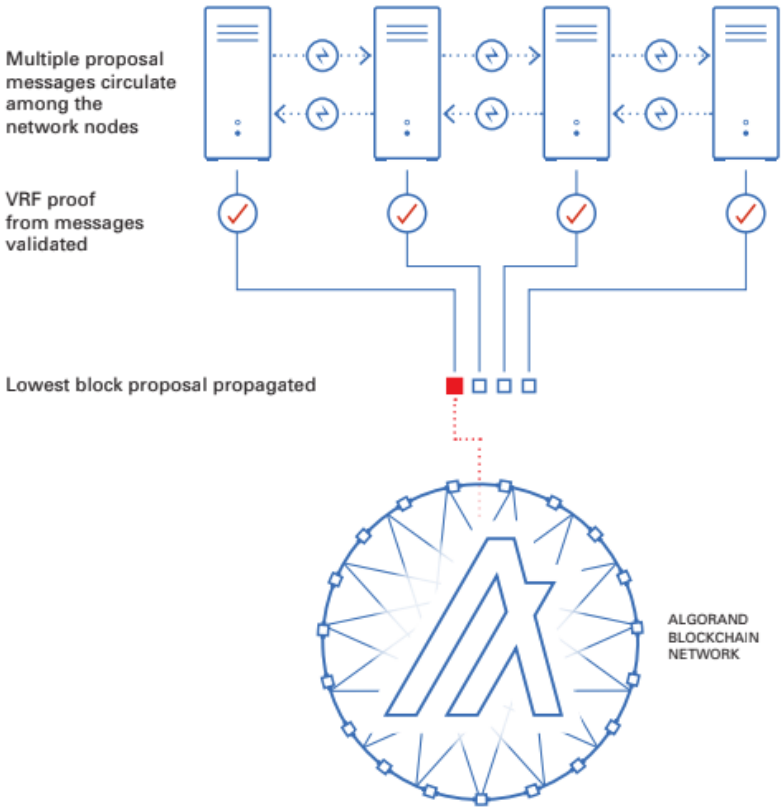

Figura 4.7: Fase Soft Vote [4]

La convenzione adottata è di attendere un tempo

$$t = \vartheta_{r-1} + \gamma_r \tag{4.5}$$

in cui il primo addendo è necessario per completare il round $r-1$, mentre il secondo è necessario per identificare la priorità più elevata nel round r.

Soft Vote

Lo scopo di questa fase è quello di filtrare il numero di proposte fino a ottenere una sola, garantendo la certificazione di un unico blocco. Ogni nodo nella rete riceverà molti messaggi di proposta da parte degli altri nodi. I nodi verificheranno la firma del messaggio e successivamente

[4] `https://developer.algorand.org/docs/get-details/algorand_consensus/`

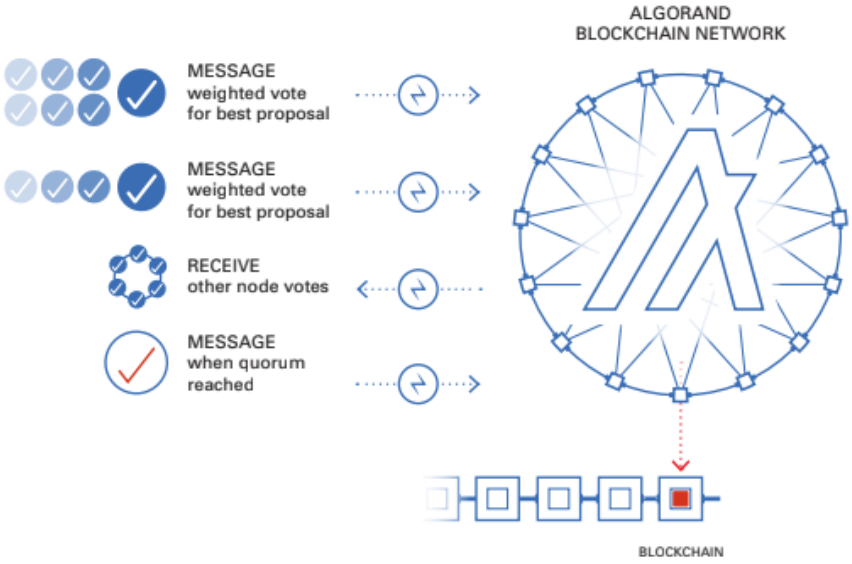

Figura 4.8: Fase Certify Vote[5]

convalideranno la selezione utilizzando la prova VRF. Successivamente, il nodo confronta l'hash di ciascuna prova VRF convalidata dei vincitori al fine di determinare il valore più basso e propaga solo la proposta di blocco con il VRF hash più basso. Questo processo continua per un intervallo di tempo fissato al fine di consentire la propagazione dei voti in tutta la rete.

Certify Vote

Dopo la fase di soft-vote, un nuovo comitato viene incaricato di verificare la proposta di blocco che è stata oggetto di votazione preliminare, al fine di rilevare eventuali anomalie quali spese eccessive, doppie spese o altri problemi. Qualora la proposta risulti valida, il nuovo comitato procede con una votazione ulteriore per certificare il blocco. Questa fase viene condotta in modo simile al soft-vote, in cui ciascun nodo esamina i propri account gestiti al fine di selezionare un comitato e inviare i voti. I voti vengono quindi raccolti e validati da ogni nodo fino al raggiungimento di un quorum, il quale determina la conclusione del round e spinge il nodo a generare un certificato per il blocco, da registrare successivamente nel ledger. A questo punto, si avvia un nuovo round e il processo riparte da capo. Se entro un certo limite di tempo,

[5] `https://developer.algorand.org/docs/get-details/algorand_consensus/`

un quorum non viene raggiunto nella votazione del comitato di certificazione, la rete entrerà in modalità di ripristino.

4.6 Smart Contract in Algorand

Gli Smart Contract sono parti di logica che risiedono nella blockchain di Algroand, con i quali è possibile implementare un servizio, un comportamento, applicare un contratto, effettuare pagamenti, gestire delle transazioni. Gli Smart Contracts in Algorand permettono anche la generazione e il trasferimento di asset, essi sono il metodo principale per l'implementazione della logica delle applicazioni distribuite (dApps). Vi sono due tipi di Smart Contract in Algorand: ***statefull e stateless***.
Gli Smart Contract in Algorand vengono scritti in TEAL: Transaction Execution Approval Language, un linguaggio assembly per la scrittura di codice che viene convertito in bytecode per la AVM (Algorand Virtual Machine).
Per una migliore astrazione, è stata creata una libreria python che permette la scrittura di codice TEAL con una sintassi più simile a quella di un linguaggio di alto livello: PyTeal. Ogni contratto compilato utilizzando TEAL ha dei limiti imposti sulla dimensione del codice compilato (massimo 1 Kb) e sul costo operativo massimo per l'esecuzione del contratto. Questi vincoli sono stati implementati per consentire ad Algorand di raggiungere un'elevata capacità di elaborazione e di mantenere commissioni di transazione basse per gli utenti finali."

4.6.1 Stateless Smart Contracts

Gli Smart Contracts *stateless* sono usati per validare le transazioni di fondi tra due parti. Essi hanno la funzione di *escrow (deposito fiduciario)*, ovvero sostituiscono quella terza parte imparziale fidata che gestisce i fondi durante gli accordi tra due parti. Nell'ambito manageriale e della finanza questi oggi sono applicati nell'ambito delle transazioni immobiliari, in caso di fusioni o acquisizioni, contratti commerciali complessi, in contesti in cui è necessario garantire la sicurezza delle risorse coinvolte fino a quando tutti gli obblighi contrattuali non siano stati adempiuti.

4.6.2 Statefull Smart Contracts

I contratti intelligenti con stato sono un elemento fondamentale della rete Algorand. Vengono chiamati attraverso le transazioni elaborate dalla blockchain. Questi consentono di essere combinati insieme ad altre funzionalità come ASAs (Algorand Standard Assets) e NFT per creare applicazioni molto complesse. Gli Smart Contracts statefull si distinguono da quelli

stateless poichè offrono la possibilità di memorizzare dei dati on-chain, a due livelli: Globale e Locale. Lo storage dei dati risulta utile per il funzionamento sulla blockchain Algorand delle applicazioni distribuite.
Ad esempio, un contratto intelligente con stato potrebbe essere utilizzato come metodo di voto [13], memorizzando dati a livello globale in base al risultato di diversi voti.

4.7 Vantaggi di Algorand

Efficacia Computazionale

La piattaforma di Algorand rappresenta, senza alcun dubbio, una innovazione nel campo blockchain. Essa amplia il concetto di ***Trilemma Blockchain***, sul quale vengono effettuati dei benchmark per valutare le prestazioni delle varie chain, aggiuggendo anche una quarta variabile importante per la valutazione: ***l'efficacia computazionale*** (computational effectiveness) [33], una misura del potenziale di una blockchain per raggiungere efficacemente gli obiettivi dell'utente a un costo accettabile.
La blockchain Algorand grazie al meccanismo di consenso PPOS richiede una potenza computazionale minima ed è altamente scalabile [45], ciò rende ideale la chain di Micali per essere applicata in contesti in cui vi è l'esigenza di una entità indipendente che garantisca il corretto svolgimento delle operazioni.

Integrazione processi delle PA

Algorand oggi si pone come un punto di riferimento nel palcoscenico internazionale, cercando di integrare i processi delle amministrazioni governative in un progetto che vede l'utilizzo della blockchain come base solida e sicura per la gestione dei principali processi. In Italia la Algorand foundation ha realizzato un accordo con SIAE per la creazione di quattro milioni di NFT per i diritti d'autore [19].
Per le ragioni di cui prima la Banca d'Italia sta sperimentando una piattaforma finanziaria su Algorand. Il progetto propone la realizzazione di una piattaforma di tokenizzazione di Titoli di Stato, con la gestione dei processi di emissione, collocazione sul mercato primario e scambio sul mercato secondario, regolati tramite Smart Contracts eseguiti in modo sicuro, scalabile ed

efficiente sulla Algorand Virtual Machine [24].

Ecosostenibilità

A differenza di altre blockchain come Bitcoin e tutte quelle chain basate su mining, Algorand è classificabile come ***blockchain verde***, ovvero ecologicamente sostenibile [3].

**** OMISSIS ****

Capitolo 6

Conclusioni

In questo lavoro è stata sviluppata una piattaforma per la gestione delle aste basata sulla blockchain Algorand. Lo scopo di questa tesi era di riuscire a realizzare un sistema che potesse essere scalabile, sicuro, trasparente ed economicamente vantaggioso, in maniera tale da presentare una valida alternativa ai sistemi centralizzati che prevedono la presenza di una entità intermediaria e il conseguente aumento dei costi. Ciò è stato possibile proprio grazie al connubio di un sistema decentralizzato e peer-to-peer come la blockchain. Sono stati progettati e sviluppati, uno Smart Contract, un protocollo per la privacy *off-chain* e una piattaforma web.
L'architettura del sistema è multistrato, suddivisa il layer, in ognuno dei quali si trova una componente fondamentale della piattaforma.
Nel sistema che è stato proposto grazie alla divisione delle operazioni tra *on-chain* e *off-chain*, come mostrato dalle valutazioni sperimentali, si è riusciti nell'intento di bilanciare i costi, rendendo questa piattaforma sostenibile e conveniente sia per chi crea un'asta che per chi vi partecipa. Gli esperimenti hanno mostrato il grande fattore di scalabilità che permette di mantenere l'andamento dei costi di un'asta lineari con il numero di transazioni e con il numero di partecipanti.

Sviluppi Futuri

Il progetto è stato sviluppato in modo da potere essere utilizzabile da dispositivi con esigue risorse di calcolo, come la maggior parte di quei dispositivi che compongono l'*IoT*[1]. Per questo uno degli sviluppi futuri di questo lavoro è poter inglobale questo sistema per la gestione delle

[1] IoT = Internet of Things

aste in un ambiente digitale come una *Smart City* o uno *Smart Building* al fine di poter far si che in caso di un guasto o di una necessità di manutenzione, i dispositivi stessi, essendo connessi a questo sistema, attraverso un sistema di percezione autonomo, possano richiedere e gestire un'asta per la loro riparazione o manutenzione, in maniera totalmente autonoma [5].
Un altro possibile sviluppo potrebbe essere quello dell'introduzione di questo sistema, adattato per il contesto, nella gestione dell'assegnazione dei lavori nelle pubbliche amministrazioni.
Le *Smart Grid* rappresentano un ambito interessante dove poter applicare un sistema di gestione delle aste, per esempio per il trading energetico [42]. Interessante sarebbe anche investigare la possibilità di poter implementare dei metodi di cifratura proprio con le istruzioni base fornite da TEAL, ed effettuare una comparazione sui costi di una soluzione *on-chain* rispetto a quella implementata in questo lavoro.

Appendice A

Appendice A - Algoritmi di Sicurezza

In questa appendice vengono descritti i principali algoritmi di sicurezza utilizzati per la realizzazione del protocollo per la privacy. La maggior parte delle immagini sono tratte del libro *Cryptography and network security: principles and practice* [41]

A.1 AES - Advanced Encryption Standard

L'AES è un algoritmo di cifratura a blocchi simmetrico che sfrutta l'aritmetica nel campo finito $GF(28)$[1] con il polinomio irriducibile $m(x) = x^8 + x^4 + x^3 + x + 1$. AES opera su blocchi di un byte. L'addizione tra due byte è definita come lo XOR bit a bit, mentre la moltiplicazione è nel $GF(28)$ con il polinomio irriducibile. Il cifrario accetta in input un blocco di testo in chiaro di 128 bit o 16 byte, rappresentato da una matrice 4x4 byte, mentre consente di utilizzare tre differenti lunghezze di chiavi 128, 192 o 256 bit e l'algoritmo prende il nome rispettivamente di AES-128, AES-192 e AES-256. Il numero di round presenti è dipendente dalla lunghezza della chiave, quindi, per una chiave di 128 bit vi sono 10 round, per una di 192 bit vi sono 12 round e per una di 256 bit vi sono 14 round. Tutti i round, tranne l'ultimo, sono costituiti da quattro diverse operazioni:L'AES è un algoritmo di cifratura a blocchi simmetrico che sfrutta l'aritmetica nel campo finito$GF(28)$ con il polinomio irriducibile $m(x) = x^8 + x^4 + x^3 + x + 1$. AES opera su blocchi di un byte. L'addizione tra due byte è definita come lo XOR bit a bit, mentre la moltiplicazione è nel $GF(28)$ con il polinomio irriducibile. Il cifrario accetta in input un blocco di testo in chiaro di 128 bit o 16 byte, rappresentato da una matrice 4x4 byte,

[1] GF = Galois Field

mentre consente di utilizzare tre differenti lunghezze di chiavi 128, 192 o 256 bit e l'algoritmo prende il nome rispettivamente di AES-128, AES-192 e AES-256. Il numero di round presenti è dipendente dalla lunghezza della chiave, quindi, per una chiave di 128 bit vi sono 10 round, per una di 192 bit vi sono 12 round e per una di 256 bit vi sono 14 round. Tutti i round, tranne l'ultimo, sono costituiti da quattro diverse operazioni:

- ***SubstitutionBytes***: effettua la sostituzione di un byte con un altro grazie alla S-box.
- ***ShiftRows***: semplice permutazione.
- ***MixColumns***: sostituzione che utilizza l'aritmetica polinomiale definita in $GF(28)$.
- ***AddRoundKey***: XOR bit a bit del blocco corrente con una parte della chiave espansa, è l'unica fase dove i bit della matrice di stato vengono combinati con i bit della chiave.

Nell'ultimo round manca l'operazione MixColumns. La chiave viene rappresentata come una matrice quadrata di byte e successivamente viene espansa in un array di parole (word), dove ogni parola è formata da quattro byte. Una chiave di 128 bit viene espansa in 44 parole, una di 192 bit in 52 parole ed una di 256 bit in 60 parole. L'AddRoundKey viene fatta sia all'inizio che alla fine del processo di cifratura, perché se non ci fosse una funzione di aggiunta immediata dei bit della chiave, qualsiasi altra funzione che fosse fatta al suo posto sarebbe totalmente predicibile. Il fatto di aggiungere in input ed output la chiave garantisce sicurezza sia in input che in output. L'operazione AddRoundKey è l'unica che utilizza la chiave, per tale motivo entrambi i processi di cifratura e decifratura iniziano con essa. Le altre tre operazioni, non facendo uso della chiave, sono invertibili e non aggiungono sicurezza, per questo la decifratura si effettua seguendo la cifratura, mentre l'AddRoundKey, essendo uno XOR, per "invertirlo" basta riproporlo.

A.2 RSA

L'algoritmo RSA appartiene alla famiglia degli algoritmi a chiave pubblica, è un cifrario a blocchi in cui il testo in chiaro ed il testo cifrato sono interi compresi tra 0 ed n-1 per un dato n, che solitamente ha una dimensione di 1024 bit cioè di 309 cifre decimali, nelle applicazioni moderne viene consigliato un n codificato su 2048 bit. Si basa sulla difficoltà di fattorizzare un numero nei suoi fattori primi, la forza dell'RSA sta proprio in questo concetto, dati i numeri primi è facile trovare il numero n, ma dato n è difficile calcolare la sua fattorizzazione. Il valore

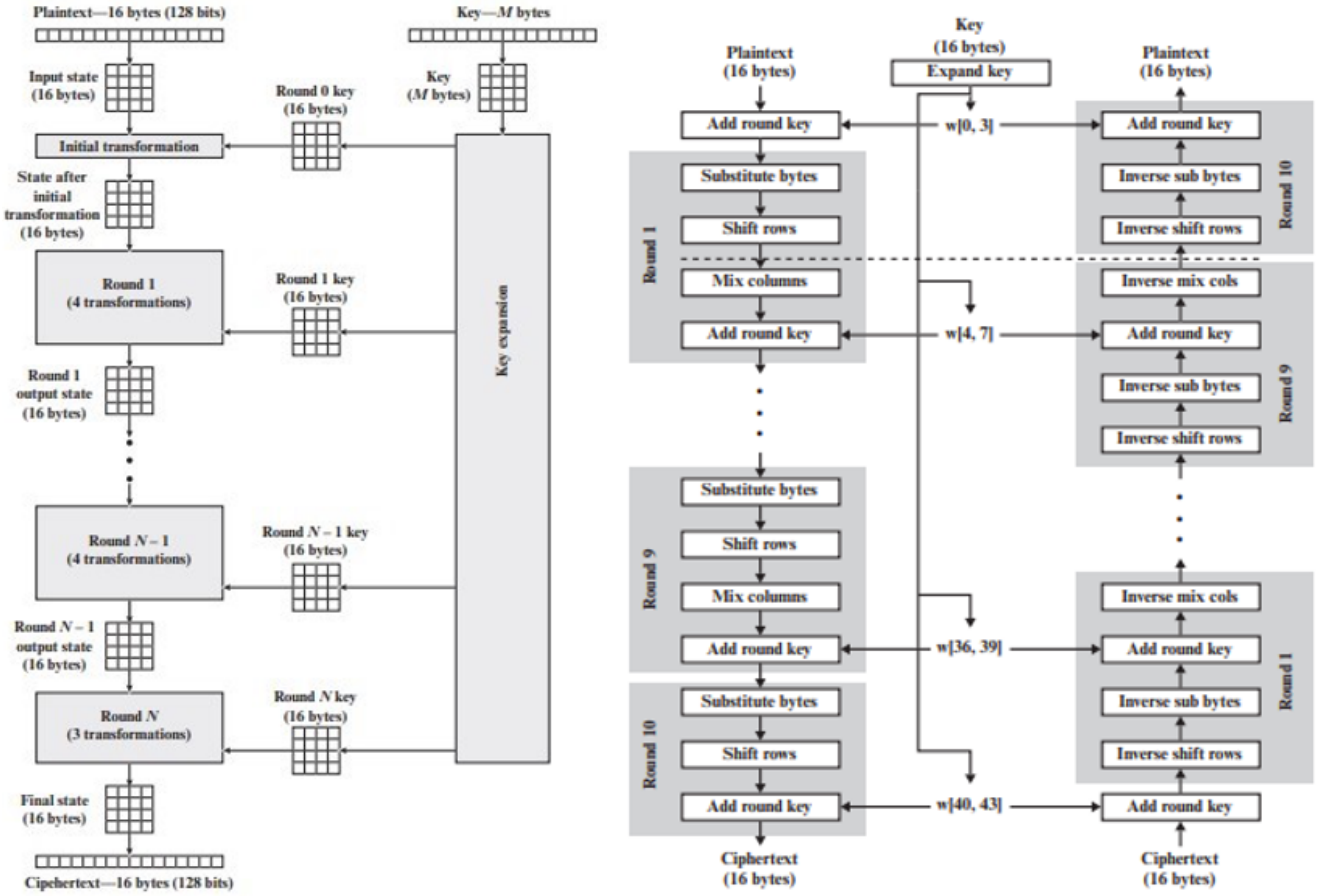

Figura A.1: Diagramma delle operazioni di cifratura(sx) del AES con chiave a 16 byte e lo schema di cifratura e decifratura a 16 byte completo (dx)

di n deve essere noto sia al mittente che al destinatario. Per codificare un blocco di testo in chiaro$M$ nel corrispondente blocco di testo cifrato C:

$$C = M^e \mod n \tag{A.1}$$

Mentre per la decodifica:

$$M = C^d \mod n = (M^e)^d \mod n = M^{ed} \mod n \tag{A.2}$$

Il mittente conosce il valore di e, mentre solo il destinatario conosce il valore di d. Quindi, la chiave pubblica è costituita da $PU = \{e,n\}$, mentre la chiave privata $PR = \{d,n\}$. L'RSA deve

soddisfare tre requisiti:

1. Deve essere possibile trovare i valori di *e, d, n* tali che $(M^e d \mod n = M)$, $\forall M < n$.
2. Deve essere relativamente semplice calcolare $(M^e \mod n)$ e $(C^d \mod n)$, $\forall M < n$.
3. Deve essere computazionalmente impossibile calcolare d, noti n ed e, cioè deve essere impossibile determinare la chiave privata a partire dalla chiave pubblica.

Il funzionamento dell'algoritmo RSA consiste in:

1. Scegliere due numeri primi p e q molto grandi, di 2048 bit, entrambi primi.
2. Calcolare il prodotto $n = p \cdot q$ e la funzione toziente di Eulero $\phi(n) = (p-1)(q-1)$. La funzione toziente di Eulero di un numero intero n rappresenta la cardinalità dell'insieme dei residui ridotto di n, questo contiene solo l'insieme dei residui che sono primi relativi con n.
3. Si sceglie un numero e che sia primo relativo con $\phi(n)$, cioè $gcd(\phi(n), e) = 1$ e $1 < e < \phi(n)$.
4. Si calcola il numero d tale che sia inverso moltiplicativo di $e \mod \phi(n)$, cioè $ed \mod \phi(n) = 1$, quindi $d = e - 1 \mod \phi(n)$.
5. Viene scelta la chiave pubblica $PU = \{e, n\}$ e la chiave privata $PR = \{d, n\}$.

Se si usasse una chiave pubblica moto piccola come e=3, allora RSA sarebbe vulnerabile ad attacchi semplici, allo stesso modo un valore di d molto piccolo esporrebbe RSA ad attacchi a forza bruta ed a varie forme di crittoanalisi.

A.3 Diffie-Hellman

L'algoritmo di Diffie-Hellman è usato per lo scambio delle chiavi e si basa sui logaritmi discreti. Il suo funzionamento è come un algoritmo di cifratura asimmetrico, ma serve per scambiare chiavi simmetriche. Lo scopo è quello di permettere a due entità di scambiarsi le chiavi simmetriche in modo sicuro. La sicurezza di questo algoritmo sta nella difficoltà di calcolare i logaritmi discreti. Supponiamo che due utenti vogliono creare una chiave condivisa.

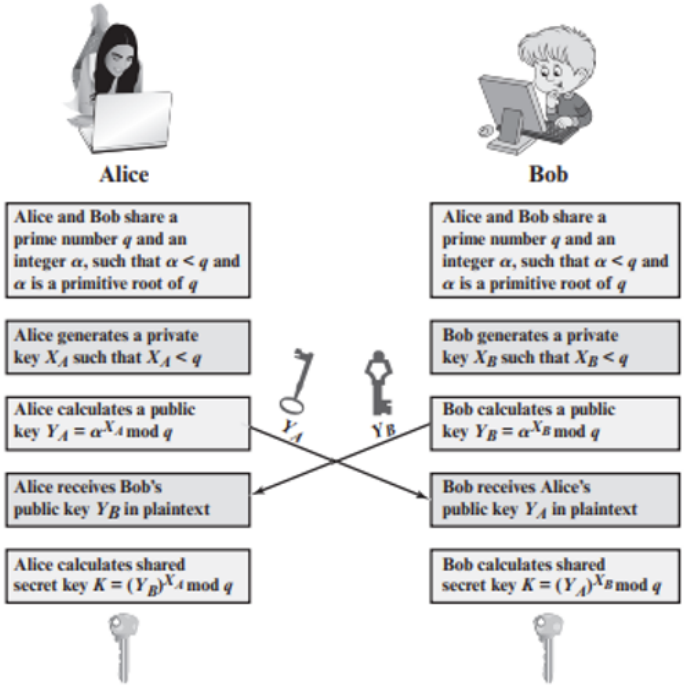

Figura A.2: Processo di scambio delle chiavi Diffie-Hellman

Inizialmente entrambi conoscono un numero primo q ed un numero intero α, tale che sia una radice primitiva di q. Se α è una radice primitiva di q, allora i numeri α mod q, $\alpha^2 \mod q$, ... , $\alpha^{q-1} \mod q$, sono tutti distinti e sono una permutazione di tutti i numeri in $[1, q-1]$. Successivamente l'utente A sceglie un intero casuale$X_A < q$, che è la sua chiave privata, e calcola la chiave pubblica $Y_A = \alpha^{(X_A)}$ mod q e la trasmette a B. Allo stesso modo l'utente B sceglie la sua chiave privata $X_B < q$ e calcola $Y_B = \alpha^{(X_B)} mod q$, che è la sua chiave pubblica, e la trasmette ad A. L'utente A, ricevuta la chiave pubblica di B, Y_B, calcola la chiave segreta condivisa$K = (Y_B)^{(X_A)} \mod q$. Anche l'utente B riceve la chiave pubblica di A, Y_A, e calcola la chiave segreta condivisa $K = (Y_A)^{(X_B)} \mod q$. Data la chiave pubblica Y_A è impossibile risalire alla chiave privata X_A , in quanto non è possibile calcolare il logaritmo discreto, cioè che dato un numero intero b ed una radice primitiva α di un numero primo p, si può trovare un esponente i unico tale che $b \equiv \alpha^i \mod p$ con $0 \leqslant i \leqslant p-1$. L'esponente i è il logaritmo discreto in base α di b modulo p, in formula:

$$i = d\log_\alpha b \mod p. \tag{A.3}$$

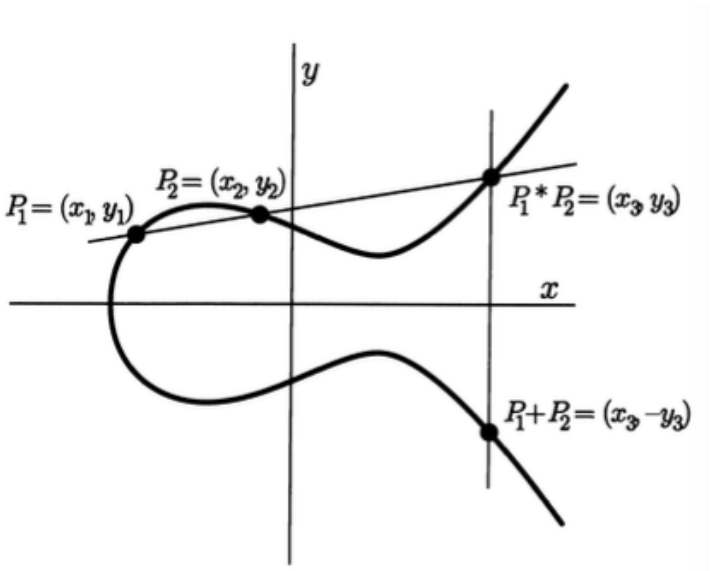

Figura A.3: Grafico curva ellittica generica

A.4 Crittografia delle curve ellittiche

La lunghezza della chiave per rendere sicuro RSA è aumentata negli anni e questo rende le applicazioni pesanti. Per ovviare a questo problema si è pensato di utilizzare la crittografia delle curve ellittiche (Elliptic Curve Cryptography – ECC), che utilizza le proprietà aritmetiche delle curve ellittiche per produrre sistemi crittografici a chiave pubblica. Questa crittografia utilizza curve ellittiche nelle quali le variabili ed i coefficienti sono ristretti agli elementi di un campo finito, in particolare vengono usate due famiglie di curve ellittiche, le curve prime su Z_p e le curve binarie su $GF(2^m)$. Le curve ellittiche crittografiche nascono per garantire la robustezza con chiavi di lunghezza minore, infatti il loro punto forte, rispetto all'RSA, è che offrono maggiore sicurezza ma con una chiave più piccola. Come tutta la crittografia a chiave pubblica , ECC si basa su funzioni matematiche semplici da calcolare in un verso, ma molto difficili da invertire, anche in questo caso la difficoltà risiede nell'impossibilità di calcolare il logaritmo discreto rispetto ad un punto base noto.

A.5 ECDH - Elliptic Curve Diffie-Hellman

Per effettuare lo scambio delle chiavi tramite curve ellittiche, si sceglie un intero grande q, che sia o un numero primo p o un intero della forma 2^m, ed i parametri a e b per l'equazione di una curva ellittica rispettivamente in Z_p o in $GF(2^m)$. Questo definisce il gruppo ellittico di punti $E_q(a,b)$ che è noto a tutti i partecipanti. Successivamente si sceglie un punto base $G = (x_1, y_1)$, noto a tutti, $\in E_p(a,b)$, il cui ordine è un intero molto grande n, dove per ordine n di un punto G su una curva ellittica si intende il più piccolo intero positivo n tale che $nG = 0$. Uno scambio di chiavi tra gli utenti A e B può avvenire nel seguente modo:

1. A sceglie un intero n_A minore di n (l'ordine di G), dove n_A è la chiave privata di A. Poi, A genera una chiave pubblica $P_A = n_A \times G$.

2. Allo stesso modo, B sceglie una chiave privata n_B e calcola una chiave pubblica $P_B = n_B \times G$.

3. A genera la chiave segreta $K = n_A \times P_B$ e B genera la chiave segreta $K = n_B \times P_A$.

I due calcoli della chiave segreta condivisa producono gli stessi risultati in quanto

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A \tag{A.4}$$

Per rompere questo schema, un attaccante dovrebbe essere in grado di calcolare k dati G e kG, cosa che si assume sia difficile.
Invece, per cifrare il messaggio in chiaro m in un messaggio P_m per essere inviato come un punto (x,y), sarà il punto P_m che verrà cifrato e poi decifrato. Come nel sistema per lo scambio di chiavi, un sistema di codifica e decodifica richiede un punto G ed un gruppo ellittico $E_q(a,b)$ come parametri. Ogni utente A sceglie una chiave privata n_A e genera una chiave pubblica $P_A = n_A \times G$. Per cifrare e inviare un messaggio P_m a B, A sceglie un intero positivo casuale k e produce il testo cifrato C_m formato dalla coppia di punti:

$$C_m = \{kG, P_m + kP_B\} \tag{A.5}$$

A sta utilizzando la chiave pubblica di B, P_B. Per decifrare il testo cifrato, B moltiplica il primo punto della coppia per la sua chiave privata n_B e sottrae il risultato al secondo punto:

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m \tag{A.6}$$

A ha mascherato il messaggio P_m aggiungendovi kP_B. Nessuno a parte A conosce il valore di k, quindi anche se P_B è una chiave pubblica, nessuno può rimuovere la maschera kP_B.

A.6 SHA-256

SHA-256 fa parte della famiglia di funzioni hash Secure Hash Algorithm 2 (SHA-2). SHA è stato sviluppato dal National Institute of Standards and Technology (NIST) ed è stato aggiornato alla versione 3 nel 2015. Ancora oggi viene utilizzata la versione 2 perché considerata ancora crittograficamente sicura. Di base le funzioni hash prevedono:

- Un input suddiviso in blocchi di dimensione b.
- Un valore iniziale IV .
- Una funzione tipicamente non lineare chiamata funzione di compressione, ripetuta più volte a formare dei round.

All'input delle funzioni hash, solitamente, viene applicato un padding di un multiplo intero di lunghezza prefissata, come 1024 bit, che include anche il valore di lunghezza del messaggio originale in bit, questo è una misura di sicurezza necessaria ad evitare di produrre un messaggio alternativo con lo stesso valore di hash.
I requisiti di sicurezza delle funzioni hash crittografiche sono:

1. ***Dimensione dell'input variabile***: la funzione hash H può essere applicata a un blocco di dati di dimensioni qualsiasi;
2. ***Dimensione dell'output fissa***: la funzione hash produrrà in output un digest di dimensione fissa;
3. ***Efficienza***: non deve essere computazionalmente oneroso calcolare la funzione hash di un elemento, quindi, $H(x)$ è facile da calcolare per un dato x;

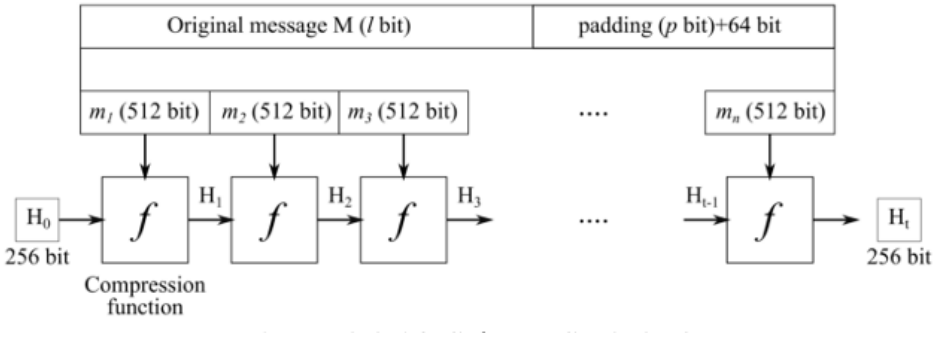

Figura A.4: Schema SHA-256.

4. ***Resistenza alla preimmagine (proprietà one-way)***: fornisce la proprietà di monodirezionalità, cioè dato il digest è impossibile risalire al messaggio originale, cioè per un dato digest h, deve essere computazionalmente impossibile trovare un y tale che $H(y) = h$;

5. ***Resistenza alla seconda preimmagine (resistenza debole alle collisioni)***: garantisce che sia impossibile trovare un messaggio alternativo con lo stesso valore hash di un messaggio dato, per un dato blocco x, è impossibile trovare un $y \neq x$ tale che $H(x) = H(y)$;

6. ***Resistenza alle collisioni (resistenza forte alle collisioni)***: è computazionalmente impossibile trovare una coppia (x, y) con $x \neq y$ tale che "H(x) = H(y)";

7. ***Pseudorandomicità***: l'output di H soddisfa i test di randomicità.

Una funzione hash che soddisfa le prime cinque proprietà è detta funzione hash debole, se viene soddisfatto anche il sesto requisito di resistenza alle collisioni, allora si parla di funzione hash forte. SHA-256 può essere utilizzato per eseguire l'hashing di un messaggio M di L bit con $0 \leqslant L \leqslant 264$ e restituisce in output un digest di 256 bit. Se il blocco è minore di 512 bit viene effettuato il padding formato da un bit posto ad 1 seguito dal numero necessario di 0. Questo garantisce che l'algoritmo elabori blocchi di messaggi di 512 bit. L'estensione di 64 bit oltre al padding rappresenta la lunghezza iniziale del messaggio.
Il cuore dell'algoritmo SHA-256 è il modulo della funzione di compressione formato da 64 round. Ogni round è costituito da 8 buffer da 32 bit ciascuno e prende in input una costante additiva di 32 bit. Le costanti utilizzate sono 64 e sono le radici cubiche dei primi 64 numeri primi. Vengono eseguite 64 round di funzione di compressione, in cui i valori hash vengono ruotati secondo uno schema specifico e vengono aggiunti ulteriori dati. Nell'ultimo round viene

prodotto un valore hash finale a 256 bit che è il digest.
Nel caso in cui due utenti volessero comunicare, SHA-256 garantisce l'integrità dei dati in modo che entrambe le parti possano essere sicure che la comunicazione provenga effettivamente dalla persona corretta. Il dispositivo destinatario crea un hash del messaggio originale e lo confronta con il valore hash inviato dal mittente. Se entrambi i valori hash sono uguali, il messaggio non è stato manomesso durante il transito.
SHA-256 può essere utilizzato anche per le firme digitali. Una firma digitale è un modo per firmare documenti digitali, codici o software verificabili dal destinatario o dagli utenti. In questo modo, si è sicuri su chi è il creatore o se un determinato file è stato manomesso. Una firma digitale viene creata applicando un hash al messaggio ed applicando a questo un algoritmo di cifratura che prende in input la chiave privata del mittente, mentre la chiave pubblica viene utilizzata dal destinatario per decifrare il messaggio. La verifica può essere effettuata applicando l'algoritmo SHA-256. L'hashing garantisce che la firma digitale non sia stata modificata da quando è stata firmata. Il sistema del destinatario esegue l'algoritmo di hashing e utilizza la chiave pubblica per decifrare il messaggio, se corrisponde, sa che i dati sono inalterati e autentici. SHA 256 è uno degli algoritmi più affidabili per l'autenticazione e la verifica dell'integrità dei messaggi.
Viene utilizzato con molti diversi protocolli e processi di autenticazione e crittografia, tra cui:

- ***SSL/TLS*** : SSL (Secure Socket Layer) e Transport Layer Security (TLS)sono protocolli di crittografia che mantengono l'integrità e la riservatezza dei dati mentre sono in transito;
- ***SSH*** : Il protocollo Secure Shell (SSH) crea un canale sicuro tra due dispositivi per il trasferimento dei dati;
- ***IPsec***: Internet Protocol Security è una raccolta di protocolli progettati per proteggere il trasferimento dei dati tra diverse reti IP;
- ***S/MIME*** : Secure/Multipurpose Internet Mail Extensions è un algoritmo per proteggere l'integrità e la riservatezza delle e-mail.

Ad oggi nessuno è riuscito ad attaccare SHA-256, idealmente ci vorrebbero risorse e tempo illimitati. Questo è il motivo per cui i certificati TLS utilizzano questo algoritmo per verificare la loro integrità, così come la rete Bitcoin.

**** OMISSIS ****

Elenco delle figure

2.1 Ciclo Gartner Blockchain[1] 8
2.2 Blockchain in generale[2] 11
2.3 Caratteristiche Smart Contracts 14

3.1 Schema generale di un sistema per la gestione di aste elettroniche basato su cloud 19
3.2 Schema a quattro fasi del sistema basato su cloud e LKH 22
3.3 Generazione delle chiavi con LKH 24
3.4 Schema generale sistema di prove a conoscenza zero 28
3.5 Schema generale NIZKPs con uso di Blockchain 29

4.1 Generazione coppia di chiavi Pubblica e Privata in Algorand 33
4.2 Inizializzazione Account Algorand 33
4.3 Trasformazione chiave pubblica in indirizzo Algorand 34
4.4 Trasformazione chiave privata in codifica base64 34
4.5 Trasformazione chiave privata in mnemonico 25 parole 35
4.6 Fase Block Proposal[3] 40
4.7 Fase Soft Vote [4] 41
4.8 Fase Certify Vote[5] 42

5.1 Operazioni di un asta a busta chiusa non informatizzata 48
5.2 Stack della piattaforma 49
5.3 Schema Modulare della piattaforma 50
5.4 Componenti del protocollo per il mantenimento della privacy. 55
5.5 Schema di flusso protocollo per la privacy con RSA 56
5.6 Generazione chiave AES in caso di RSA. 57

5.7 Schema di flusso protocollo per la privacy con ECC e KDF 59
5.8 Diagramma di sequenza fase 1, richiesta di creazione di una asta da parte di un utente[6] 63
5.9 Diagramma di sequenza fase 2, flusso dei messaggi all'interno della *bidding window* 64
5.10 Diagramma di sequenza fase 3 flusso dei messaggi all'interno della *key transmission window* 65
5.11 Diagramma di sequenza fase 4 illustrazione delle tre sottofasi 66
5.12 Home page della web app. 68
5.13 Modello composizione Web Application 69
5.14 Media del costo della commissione di una transazione in dollari americani dal 22/05/2023 al 22/06/2023[7] 72

A.1 Diagramma delle operazioni di cifratura(sx) del AES con chiave a 16 byte e lo schema di cifratura e decifratura a 16 byte completo (dx) 80
A.2 Processo di scambio delle chiavi Diffie-Hellman 82
A.3 Grafico curva ellittica generica 83
A.4 Schema SHA-256. 86

B.1 Login page della web app. 88
B.2 Register page della web app. 89
B.3 Login page della web app. 89
B.4 Home page del Manager. 90
B.5 Dettaglio Asta. 91
B.6 **a.** Home Page Client Mobile **b.** Partecipa Asta Mobile **c.** Offerta Mobile . . . 92

Elenco delle tabelle

5.1 Stato Globale 52
5.2 Stato Locale 52
5.3 Costi transazioni e saldo minimo 73
5.4 Transazioni utenti 73
5.5 Simulazione costi creatore 74

Bibliografia

[1] S. Ames, C. Hazay, Y. Ishai e M. Venkitasubramaniam. «Ligero: Lightweight sublinear arguments without a trusted setup». In: *Proceedings of the 2017 acm sigsac conference on computer and communications security*. 2017, pp. 2087–2104.

[2] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich et al. «Hyperledger fabric: a distributed opera- ting system for permissioned blockchains». In: *Proceedings of the thirteenth EuroSys conference*. 2018, pp. 1–15.

[3] A. O. Bada, A. Damianou, C. M. Angelopoulos e V. Katos. «Towards a Green Blockchain: A Review of Consensus Mechanisms and their Energy Consumption». In: *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2021, pp. 503–511.

[4] D. J. Bernstein. «Curve25519: new Diffie-Hellman speed records». In: *Public Key Cryptography-PKC 2006: 9th International Conference on Theory and Practice in Public- Key Cryptography, New York, NY, USA, April 24-26, 2006. Proceedings 9*. Springer. 2006, pp. 207–228.

[5] A. Bordonaro, A. De Paola, G. Lo Re e M. Morana. «Smart Auctions for Autonomic Ambient Intelligence Systems». In: *2020 IEEE International Conference on Smart Computing (SMARTCOMP)*. 2020, pp. 180–187.

[6] R. G. Brown, J. Carlyle, I. Grigg e M. Hearn. «Corda: an introduction». In: *R3 CEV, August* 1.15 (2016), p. 14.

[7] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille e G. Maxwell. «Bulletproofs: Short proofs for confidential transactions and more». In: *2018 IEEE symposium on security and privacy (SP)*. IEEE. 2018, pp. 315–334.

[8] C. Burch. «Django, a web framework using python: Tutorial presentation». In: *Journal of Computing Sciences in Colleges* 25.5 (2010), pp. 154–155.

[9] V. Buterin et al. «A next-generation smart contract and decentralized application plat- form». In: *white paper* 3.37 (2014), pp. 2–1.

[10] J. Chen e S. Micali. *Algorand*. 2017. arXiv: 1607.01341 [cs.CR].

[11]J. Dai e M. A. Vasarhelyi. «Toward blockchain-based accounting and assurance». In: *Journal of information systems* 31.3 (2017), pp. 5–21.

[12] H. Eenmaa-Dimitrieva e M. J. Schmidt-Kessen. «Creating markets in no-trust environments: The law and economics of smart contracts». In: *Computer law & security review* 35.1 (2019), pp. 69–88.

[13] C. Esposito e C. Choi. «Design and Implementation of a Blockchain-based e-Voting system by using the Algorand platform». In: *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. 2023, pp. 715–723.

[14] C. Fontaine e F. Galand. «A survey of homomorphic encryption for nonspecialists». In: *EURASIP Journal on Information Security* 2007 (2007), pp. 1–10.

[15] H. S. Galal e A. M. Youssef. «Verifiable sealed-bid auction on the ethereum blockchain». In: *Financial Cryptography and Data Security: FC 2018 International Workshops, BIT- COIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers 22*. Springer. 2019, pp. 265–278.

[16] K. Gilani, F. Ghaffari, E. Bertin e N. Crespi. «Self-sovereign Identity Management Framework using Smart Contracts». In: *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE. 2022, pp. 1–7.

[17] R. Haakegaard e J. Lang. «The elliptic curve diffie-hellman (ecdh)». In: *Online at https://koclab. cs. ucsb. edu/teaching/ecc/project/2015Projects/Haakegaard+ Lang. pdf* (2015).

[18] S. Hohl e F. Rezabek. «Seminar Innovative Internet Technologies: Zero Knowledge Proofs». In: *Network* 39 (2022).

[19] F. Idelberger e P. Mezei. «Non-fungible tokens». In: *Internet Policy Review* 11.2 (2022).

[20] M. Islam, M. Chetty, S. Lim, M. Chadhar e S. Islam. «Blockchain Based Smart Auction Mechanism for Distributed Peer-to-Peer Energy Trading». In: *Proceedings of the 55th Hawaii International Conference on System Sciences*. 2022.

[21] V. Kapoor, V. S. Abraham e R. Singh. «Elliptic curve cryptography». In: *Ubiquity* 2008.May (2008), pp. 1–8.

[22] I. Kotsiuba, A. Velykzhanin, O. Biloborodov, I. Skarga-Bandurova, T. Biloborodova, Y. Yanovich e V. Zhygulin. «Blockchain evolution: from bitcoin to forensic in smart grids». In: *2018 IEEE international conference on big data (big data)*. IEEE. 2018, pp. 3100– 3106.

[23] H. Krawczyk e P. Eronen. *HMAC-based extract-and-expand key derivation function (HKDF)*. Rapp. tecn. 2010.

[24] R. La Rocca, R. Mancini, M. Benedetti, M. Caruso, S. Cossu, G. Galano, S. Mancini, G. Marcelli, P. Martella, M. Nardelli et al. *Integrating DLTs with market infrastructures: analysis and proof-of-concept for secure DvP between TIPS and DLT platforms*. Rapp. tecn. Bank of Italy, Directorate General for Markets e Payment System, 2022.

[25] L. Lamport, R. Shostak e M. Pease. «The Byzantine generals problem». In: *Concurrency: the works of leslie lamport*. 2019, pp. 203–226.

[26] H. Li e W. Xue. «A blockchain-based sealed-bid e-auction scheme with smart contract and zero-knowledge proof». In: *Security and Communication Networks* 2021 (2021), pp. 1–10.

[27] Y. Lindell. *Secure Multiparty Computation (MPC)*. Cryptology ePrint Archive, Paper 2020/300. https://eprint.iacr.org/2020/300. 2020. U R L: https://eprint. iacr.org/2020/300.

[28] T. Lorünser, F. Wohner e S. Krenn. «A Privacy-Preserving Auction Platform with Public Verifiability for Smart Manufacturing.» In: *ICISSP* 2022 (2022), pp. 637–647.

[29] M. Maller, S. Bowe, M. Kohlweiss e S. Meiklejohn. «Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings». In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2019, pp. 2111–2128.

[30] F. Martín-Fernández, P. Caballero-Gil e C. Caballero-Gil. «Authentication based on non- interactive zero-knowledge proofs for the internet of things». In: *Sensors* 16.1 (2016), p. 75.

[31] S. Micali, M. Rabin e S. Vadhan. «Verifiable random functions». In: *40th annual symposium on foundations of computer science (cat. No. 99CB37039)*. IEEE. 1999, pp. 120–130.

[32] E. Milanov. «The RSA algorithm». In: *RSA laboratories* (2009), pp. 1–11.

[33] F. Mogavero, I. Visconti, A. Vitaletti e M. Zecchini. «The Blockchain Quadrilem- ma: When Also Computational Effectiveness Matters». In: *2021 IEEE Symposium on Computers and Communications (ISCC)*. 2021, pp. 1–6.

[34] S. Nakamoto. «Bitcoin whitepaper». In: *URL: https://bitcoin. org/bitcoin. pdf-(: 17.07. 2019)* (2008).

[35] S. Nakamoto. «Bitcoin: A peer-to-peer electronic cash system». In: *Decentralized business review* (2008), p. 21260.

[36] I. A. Omar, H. R. Hasan, R. Jayaraman, K. Salah e M. Omar. «Implementing decentralized auctions using blockchain smart contracts». In: *Technological Forecasting and Social Change* 168 (2021), p. 120786.

[37] P. Paillier. «Public-key cryptosystems based on composite degree residuosity classes». In: *Advances in Cryptology—EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings 18*. Springer. 1999, pp. 223–238.

[38] H. Qusa, J. Tarazi e V. Akre. «Secure e-auction system using blockchain: UAE ca- se study». In: *2020 Advances in Science and Engineering Technology International Conferences (ASET)*. IEEE. 2020, pp. 1–5.

[39] N. Rajkumar, N. Mohanasuganthi, M. Gokul. e K. Satheeskumar. «LKH Model based Enhancing Security in E - Auction System for Cloud Communication». In: *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. 2023, pp. 1414–1419.

[40] R. Rivera, J. G. Robledo, V. M. Larios e J. M. Avalos. «How digital identity on blockchain can contribute in a smart city environment». In: *2017 International smart cities conference (ISC2)*. IEEE. 2017, pp. 1–4.

[41] W. Stallings e M. P. Tahiliani. «Cryptography and network security: principles and practice, vol. 6». In: *editor: Pearson London* (2014).

[42] M. Stübs, W. Posdorfer e S. Momeni. «Blockchain-Based Multi-Tier Double Auctions for Smart Energy Distribution Grids». In: *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2020, pp. 1–6.

[43] S. C. Tan e S. H. Heng. «Secure Cryptographic E-Auction System». In: *International Journal of Technology* 13.6 (2022), p. 1222.

[44] M. N. Temte. «Blockchain challenges traditional contract law: Just how smart are smart contracts». In: *Wyo. L. Rev.* 19 (2019), p. 87.

[45] G. Varavallo, G. Caragnano, F. Bertone, L. Vernetti-Prot e O. Terzo. «Traceability platform based on green blockchain: an application case study in dairy supply chain». In: *Sustainability* 14.6 (2022), p. 3321.

[46] S. Vincent. «Primer on NIZK Proofs for Secure Computation». In: (2018).

[47] X. Wang e Y. Guan. «A Hierarchy Byzantine Fault Tolerance Consensus Protocol Based on Node Reputation». In: *Sensors* 22.15 (2022), p. 5887.

[48] S. Wu, Y. Chen, Q. Wang, M. Li, C. Wang e X. Luo. «CReam: A smart contract enabled collusion-resistant e-auction». In: *IEEE Transactions on Information Forensics and Security* 14.7 (2018), pp. 1687–1701.

[49] A. De Paola, P. Ferraro, G. Lo Re, M. Morana, M. Ortolani. A fog-based hybrid intelligent system for energy saving in smart buildings. In Journal of Ambient Intelligence and Humanized Computing (JAIHC), 2019, issn 1868-5145, doi 10.1007/s12652-019-01375-2;

[50] V. Agate, P. Ferraro, S. Gaglio. A Cognitive Architecture for Ambient Intelligence Systems. In Proceedings of the 6th International Workshop on Artificial Intelligence and Cognition (AIC 2018), 2018;

[51] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, M. Morana, M. Ortolani, D. Peri. A Context-aware System for Ambient Assisted Living. In Proceedings of the 11th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2017), 2017;

[52] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, M. Morana, M. Ortolani, D. Peri. An Ambient Intelligence System for Assisted Living. In Proceedings of the International Annual Conference of AEIT (2017);

[53] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, S. Das. An Adaptive Bayesian System for Context-Aware Data Fusion in Smart Environments. In IEEE Transactions on Mobile Computing, vol. PP, n. 99, doi: 10.1109/TMC.2016.2599158, ISSN 1536-1233, 2016;

[54] A. De Paola, G. Lo Re, M. Morana, M. Ortolani. SmartBuildings: an AmI System for Energy Efficiency. In Proceedings of the 4th International Conference on Sustainable Internet and ICT for Sustainability, 2015, pp. 1-7, doi: 10.1109/SustainIT.2015.7101372;

[55] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re. Autonomic Behaviors in an Ambient Intelligence System. In Proceedings of the 2014 IEEE Symposium on Computational Intelligence for Human-like Intelligence (IEEE SSCI 2014);

[56] A. De Paola, M. Ortolani, G. Lo Re, G. Anastasi, S.K. Das. Intelligent Management Systems for Energy Efficiency in Buildings: A Survey. ACM Computing Surveys, 2014, doi http://dx.doi.org/10.1145/2611779;

[57] A. De Paola, G. Lo Re, M. Morana, M. Ortolani. An intelligent system for energy efficiency in a complex of buildings. In Proceedings of the 2nd International Conference on Sustainable Internet and ICT for Sustainability, 2012, pp. 1-5;