Sistemi basati su Blockchain

La Blockchain è una struttura dati con determinate caratteristiche che le consentono di essere applicata soprattutto in sistemi decentralizzati anche detti Peer-To-Peer. Questi sistemi che si contrappongono a quelli centralizzati, prevedono che non ci sia una gerarchia tra le entità(o nodi) che coinvolgono.
Dal punto di vista pratico la Blockchain rappresenta un ledger condiviso da tutte le entità destinato a memorizzare informazioni(più esattamente transazioni) di vario genere secondo un ordine cronologico.
La più importante caratteristica della Blockchain è l'immutabilità secondo cui non è possibile eseguirvi la cancellazione fisica di informazioni che sono state inserite precedentemente.L'unica operazione che può essere effettuata è l'append,cioè l'inserimento di informazioni.
Il fatto che questa sia un registro append-only consente di poter visualizzare l'intera storia delle transazioni memorizzate.

Criptovalute

Una delle applicazioni più importanti è sicuramente quella che riguarda le criptovalute.
La criptovaluta è una moneta digitale utilizzata all'interno di un sistema P2P e che di conseguenza non viene gestita da un'autorità centrale.Contrariamente avviene per la valuta di uno stato gestita dalla Banca Centrale(ad es. l'Euro è gestito dalla BCE).
Nelle criptovalute basate su tecnologia Blockchain vengono memorizzate su quest'ultima tutte le transazioni che avvengono tra gli utenti all'interno del sistema.Qui le transazioni non sono altro che informazioni sullo scambio di una quantità di moneta tra due utenti.
Affinché siano usabili nella pratica le criptovalute devono inglobare alcuni degli attributi più importanti che hanno i sistemi monetari tradizionali.
Fra questi:

- **FIDUCIA DEGLI UTENTI NEL SISTEMA**
  Nei sistemi monetari tradizionali gli utenti si affidano ad una autorità centrale(Banca Centrale).
- **IDENTIFICAZIONE UNIVOCA DELLA MONETA**

Effettivamente le banconote fisiche sono identificate univocamente da un
codice.

- NON FALSIFICABILITÀ

Nelle monete tradizionali si ha la “non falsificabilità” quando il costo di contraffazione di una banconota supera quello della banconota stessa.

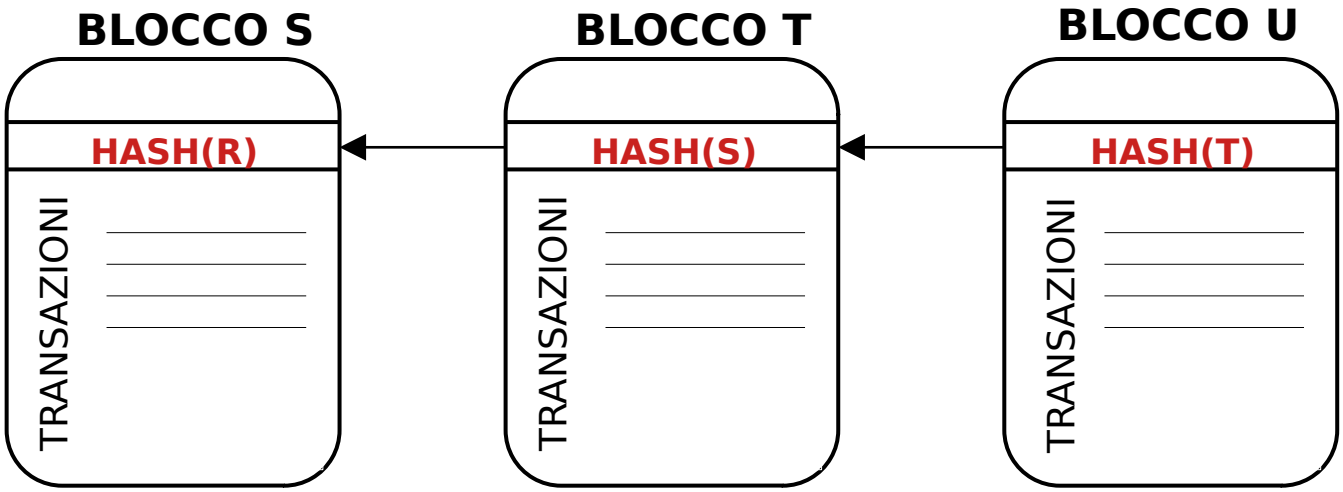
La fiducia degli utenti deve essere ottenuta in tutti i sistemi in cui si effettuano transazioni economiche.Nei sistemi bancari gli estremi di una transazione si affidano alla banca stessa che funge da intermediario.
Nell’elaborare una transazione una banca dovrebbe verificare approfonditamente una serie di aspetti che rendano la transazione sicura.
Ad esempio se un utente Alice versa una quatità di denaro “x” all’utente Bob la Banca si deve assicurare che al termine della transazione Bob abbia già speso la quantità “x” e non possa rispenderla(double spending).Quindi l’utente non visualizza effettivamente tutte le operazioni svolte per elaborare la transazione da parte della banca.
Diversamente avviene nei sistemi basati su Blockchain dove ogni utente compresi coloro che non sono coinvolti direttamente nella transazione possono visualizzare e in un certo senso elaborare i dati.Quest’ultimo aspetto riguarda il fatto che quando avviene una transazione tutti gli utenti dovrebbero dare il consenso per mandarla a buon fine.Quindi molto spesso in questi sistemi sono implementati protocolli distribuiti noti come algoritmi di consenso.
Nelle criptovalute si vogliono prevenire tutte le possibili problematiche che riguardano le transazioni economiche come il double spending o l’appropriazione indebita di quantità di denaro.
Una funzione vitale all’interno di questi sistemi è svolto dalla crittografia e dalle funzioni hash utilizzate per diversi aspetti.

Il sistema Bitcoin

Bitcoin è la prima criptovaluta basata sulla Tecnologia Blockchain ed il suo successo è legato al fatto che ingloba in maniera ottimale gli attributi che caratterizzano una moneta tradizionale.

Struttura della Blockchain

La struttura della Blockchain è la stessa per tutti i sistemi che la implementano mentre a cambiare è il suo contenuto.
Dal punto di vista strutturale può essere considerata come una lista concatenata.
Una lista concatenata è data da un insieme di nodi ognuno dei quali è connesso in qualche modo al precedente.I nodi rappresentano formalmente i blocchi della catena.
A differenza delle liste concatenate però le catene è consentita solo l'operazione di append.Ciò significa che se si vuole aggiungere un blocco questo viene "appeso" in coda alla lista. Quindi questo nuovo blocco verrà collegato a quello aggiunto per ultimo.
Il collegamento tra un blocco e il precedente viene implementato attraverso le funzioni hash.
In particolare se un blocco T è preceduto da un blocco S nel blocco T viene memorizzato l'hash del blocco precedente.
Tutti i blocchi memorizzano lo stesso tipo di informazioni ad esclusione del primo blocco.Il primo blocco detto "Blocco Genesi" è un blocco atipico in quanto privo di predecessore.
Gli altri blocchi memorizzano in una parte iniziale detta header dei metadati tra cui l'hash del predecessore.Nella seconda parte detta body sono invece memorizzate le transazioni effettuate all'interno del sistema.



Mining

L'obbiettivo di un nodo della rete è quello di validare un blocco da lui stesso creato. L'operazione inizia cercando di costruire un blocco valido tramite la risoluzione di un puzzle crittografico. Il processo di validazione è chiamato mining.
La seconda parte del blocco, il body, contenente le transazioni è di dimensione variabile, mentre l'header ha una dimensione fissata.
I campi dell'header sono 6:

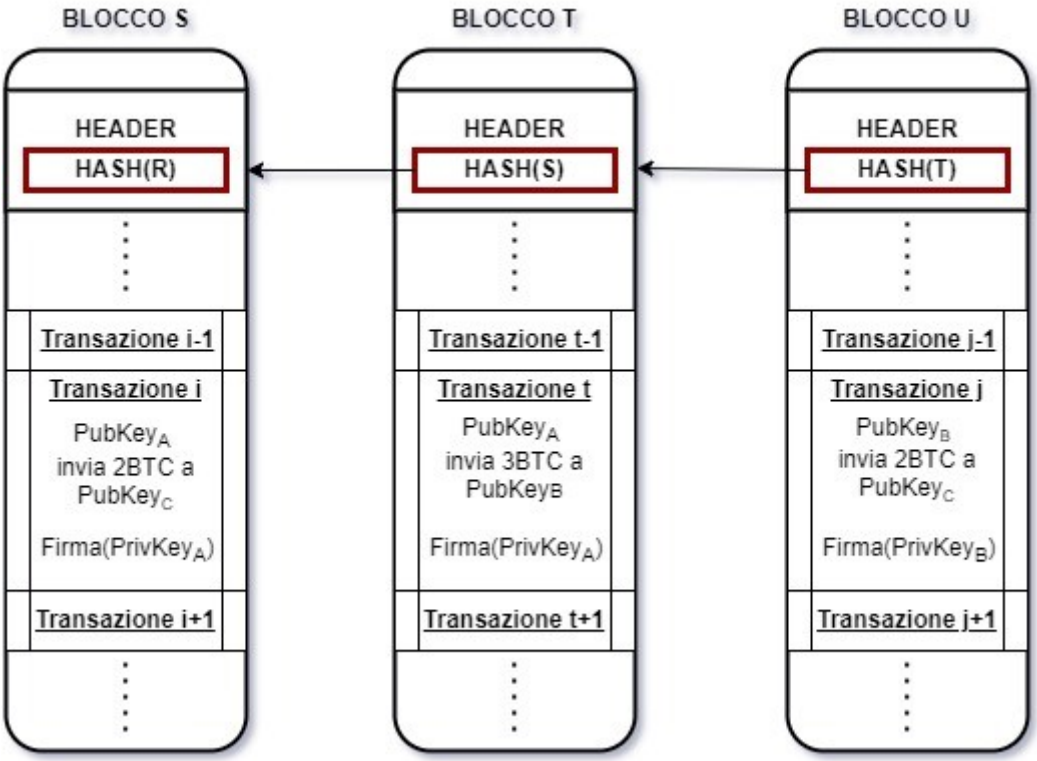
- version: il tipo di regole di validazione che il blocco deve seguire,
- previous block header hash: l'hash (un doppio SHA256) dell'header del blocco precedente,
- merkle root hash: l'hash (un doppio SHA256) di una struttura dati omonima che identifica le transazioni del blocco,
- time: il timestamp dell'ultima transazione,
- nBits: soglia target per cui l'hash del blocco deve essere minore o uguale,
- nonce: numero arbitrario atto a raggiungere il target.

Dopo aver verificato l'intera blockchain, un nodo sceglie tra le transazioni pendenti quelle che preferisce (il miner prende una fee sulle transizioni che inserisce nel blocco), crea una merkle root, e inizia a modificare il nonce in modo da ottenere un hash minore o uguale della soglia target risolvendo quindi il blocco.
Trovato un hash adatto, il blocco viene trasmesso alla rete.
L'aggiunta di un blocco “conia”(crea) BTC.
Inizialmente il premio per il mining di un blocco era di 50 BTC, ma la quantità è stata programmata per decrescere secondo una progressione geometrica che dimezza il premio ogni 210 000 blocchi.
Il volume totale dei BTC è quindi limitato a 21 milioni, che dovrebbero essere creati in circa 130 anni.

Transazioni

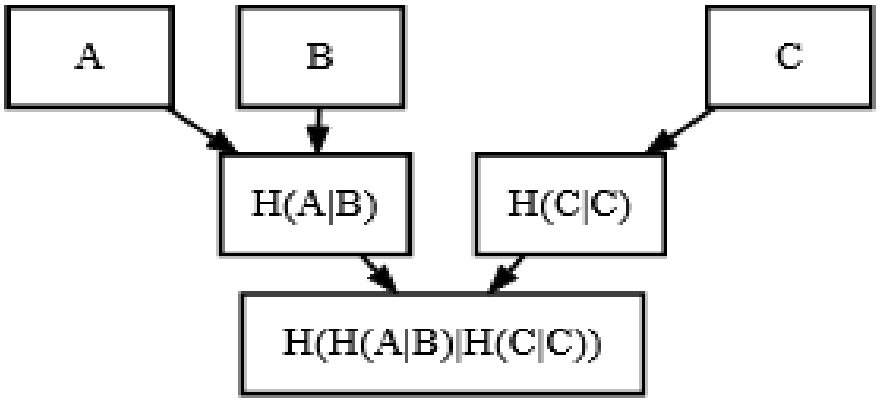
Si differenziano due tipi di utenti Bitcoin: i nodi della rete (che hanno scaricato il software di Bitcoin, possiedono una copia della Blockchain e tentano di creare nuovi blocchi, e quelli che la utilizzano solo per effettuare transazioni sulla catena.

Un utente del secondo tipo può quindi effettuare una transazione in cui trasferisce un certo quantitativo di BTC ad un altro utente.
Formalmente la transazione ha come estremi due chiavi pubbliche in modo da mantenere l'anonimato degli utenti.
Quando ci si interfaccia all'utilizzo dei BTC, si crea una coppia di chiavi (K_{pub}, K_{priv}) chiamata "wallet", che verranno legate tramite le transazioni ad un certo quantitativo di BTC.
Ad esempio se Alice(K_{pub_A}, K_{priv_A}) vuole dare una quantità n di BTC a Bob(K_{pub_B}, K_{priv_B}), allora Alice rinuncia alla sua proprietà aggiungendo la chiave pubblica di Bob K_{pub_B} alla quantità n e firma la transazione con la sua chiave privata K_{priv_A}.
A questo punto viene inviato un messaggio con la transazione tramite la rete P2P di Bitcoin in modo tale che possano essere validati dagli altri nodi.
Sulla catena appaiono solo delle chiavi pubbliche, una del mittente e una del destinatario, e mai dei riferimenti ai "proprietari" di tali chiavi, mentre la transazione in sé è visibile a tutti usando la chiave pubblica mittente.
Ogni utente di Bitcon può possedere un numero arbitrario di coppie (K_{pub}, K_{priv}) , e quindi di wallet senza avere alcun contatto con altri nodi della rete, migliorando così l'anonimato.
Nella situazione sopra descritta, si può definire un proprietario di una certa quantità di BTC come l'utente che possiede la chiave privata associata alla chiave pubblica legata a quel BTC, indipendentemente da chi quelle chiavi le abbia generate.



Merkle Tree

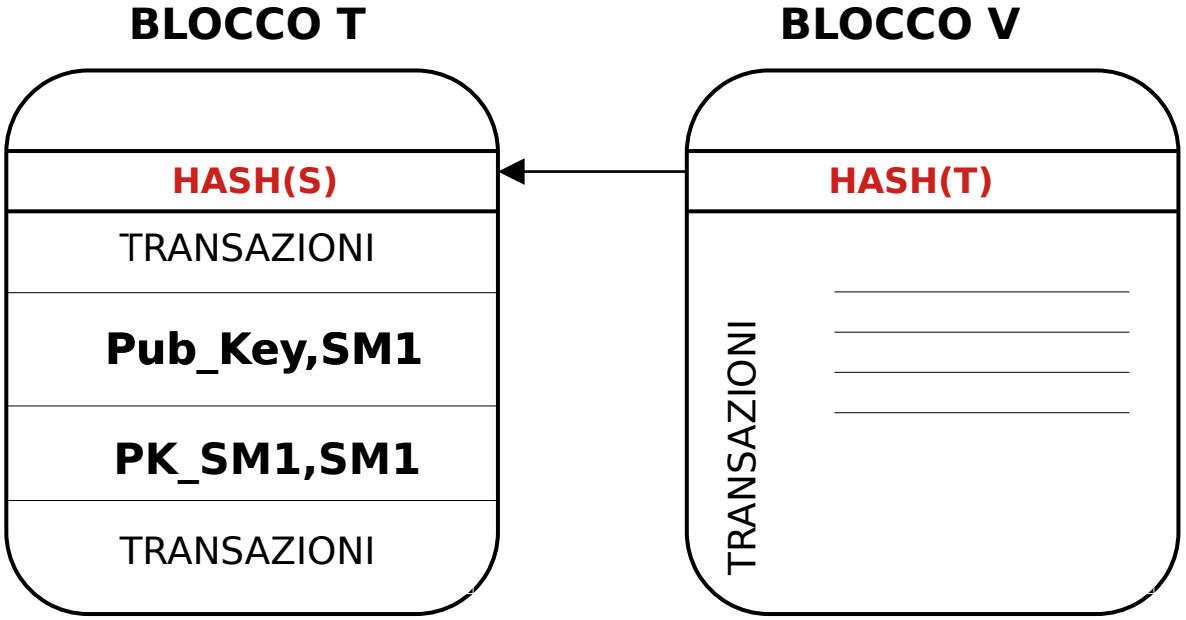
Per minare un blocco è necessario fare l'hashing delle transazioni che dovrebbe contenere, invece di effettuare l'hash di ogni transazione ad ogni cambio di nonce.
Un Merkle Tree permette di effettuare un solo hash e lavorare poi solo sull'header invece che su tutto il blocco.
L'albero si crea utilizzando l'hash di ogni transazione come foglia, e i nodi interni si creano effettuando l'hash dei nodi a 2 a 2, fino ad arrivare ad un unico hash che chiameremo Merkle root.



Questo processo permette di controllare se le transazioni all'interno di un blocco siano state manomesse.Infatti non si può trovare un header del blocco valido e in seguito modificare la lista di transazioni, perché facendolo cambierebbe la Merkle root. Quando il blocco viene inviato ad altri nodi, questi calcolano la root dalla lista di transazioni. Se non corrisponde a quella nell'header, rifiutano il blocco perché è stato manomesso.

Ethereum

Ethereum è un altro sistema basato su Blockchain e viene definito come ambiente decentralizzato per l'esecuzione di vari servizi.Tali servizi includono lo scambio tra utenti di quantità di una moneta digitale analoga a Bitcoin nota come ETH,la realizzazione e la messa in atto di smart contract e Dapp(applicazioni decentralizzate).
In pratica in Ethereum la blockchain non memorizza solamente transazioni di ETH come si limita a fare Bitcoin con la sua valuta,ma consente di implementare ed eseguire dei programmi(smart contracts).
Gli smart contracts rappresentati da un insieme di linee di codice scritti in un linguaggio supportato dall'ambiente vengono tradotti in bytecode dalla macchina virtuale EMV(Ethereum Virtual Machine).
Le istruzioni contenute nel bytecode sono eseguite dagli utenti del sistema.Senza scendere troppo nel dettaglio implementativo uno smart contract è l'istanziazione di una classe Solidity(linguaggio usato per la OOP la cui sintassi è un ibrido tra JS e C++).
Consideriamo il caso in cui un utente che interagisce nel sistema in anonimato tramite la pub_key decide di depositare uno smart contract SM1.Per effettuare questa transazione un utente dovrà pagare una certa quantità di ETH definita dal GAS(è un costo determinato in funzione delle operazioni da effettuare).Ad appendere la transazione alla blockchain saranno dei nodi della rete analoghi ai miners in Bitcoin.Quindi dei particolari nodi che vogliono risolvere problemi di hashing effettueranno l'append della transazione guadagneranno quantità di ETH relative al GAS necessario per l'append della transazione.
Successivamente all'append della transazione "Pub_key ha depositato SM1" viene effettuata una seconda transazione nel quale viene associata una chiave pubblica allo smart contract.



Una classe, come avviene generalmente nella programmazione a oggetti, è caratterizzata da “variabili d’istanza” cioè che appartengono esclusivamente ad un’istanza particolare della classe e “variabili di classe”(membri statici) cioè che essendo caratterizzanti per la classe saranno si presenteranno in tutte le istanze.Quest’ultimo tipo di variabili(static) possono essere modificate se si modifica tutta la classe.Quando avviene la modifica di una variabile di classe si sta effettua una transazione nella blockchain di Ethereum.
Consideriamo un esempio emblematico della differenza che c’è tra Ethereum e Bitcoin.
Supponiamo di creare uno smart contract che memorizza delle quantità di moneta arbitraria posseduta da un insieme di utenti all’interno di un sistema economico.
Una variabile statica del sistema potrebbe essere per l’appunto la lista delle coppie (utente,quantità di moneta detenuta).
Se un utente di tale sistema desse una quantità di moneta ad un altro utente del sistema effettivamente bisogna modificare la lista che è una variabile di classe.Modificando la variabile di classe si va ad effettuare una nuova transazione che verrà aggiunta da un miner alla blockchain di Ethereum(come sempre un miner riceve degli ETH per aver impiegato un certo GAS).
Ciò che è stato creato è effettivamente una criptovaluta simile per certi versi a Bitcoin la cui blockchain è una sotto-blockchain di ethereum.
Pertanto Ethereum si discosta da Blockchain in quanto è un ambiente distribuito che permette di memorizzare transazioni il cui contenuto può essere talmente complesso tanto da essere effettivamente un programma eseguibile.

Gas

Come in Bitcoin anche in Ethereum per effettuare una transazione un utente deve pagare una commissione(fee).
In Bitcoin questa commissione,pagata in Satoshi(unita più piccola di BTC), ha fondamentalmente 2 scopi:

- Il primo scopo è quello di incentivare un miner a validarla. Dunque un miner per validare una transazione avrà un certo dispendio energetico dovuto alla Proof-Of-Work.Pertanto maggiore sarà la commissione pagata da un utente e più il miner si sentirà incentivato a validarla.
- Il secondo scopo è legato alla sicurezza.Se un utente volesse in qualche modo “corrompere” il sistema richiedendo l’esecuzione di molteplici transazioni che vadano in conflitto dovrebbe pagare una

fee piuttosto elevata affiché queste vengano validate all'interno di un blocco in tempi ravvicinati.
In Ethereum si hanno scenari molto simili a Bitcoin ma leggermente più complessi.Questo perché mentre Bitcoin è un sistema che consente agli utenti di effettuare semplici transazioni economiche,Ethereum è un ambiente sul quale si possono depositare programmi eseguibili(smart contract) ma anche impiegare servizi(app decentralizzate).
Dunque un possibile attacco all'ambiente Ethereum che potrebbe generare Denial Of Service potrebbe consistere nel depositare uno Smart Contract il cui contenuto rappresenti un programma in Solidity che una volta mandato in esecuzioni non termina mai(si pensi all'Halting Problem).L'esecuzione di questo Smart Contract seppur pagata bene da un utente potrebbe in qualche modo compromettere l'intero sistema.
Quest'ultimo e altri aspetti sono affrontati con il concetto di Gas.
In Ethereum il Gas è un modo per misurare il lavoro svolto da un miner per effettuare operazioni.Tali operazioni possono essere legate ad una transazione,all'esecuzione di uno smart contract o altro.
Nel caso degli smart contract,programmi scritti in solidity, in base al Bytecode generato dall'EVM, il sistema stabilisce quante unità di gas servono al più per l'esecuzione di una serie di operazioni.
Se un utente vuole usufruire di tale smart contract dovrà pagare il gas di cui il miner avrà bisogno per eseguirlo.Analogamente accade nel caso delle transazioni per le quali servirà una certa quantità di unità di gas necessario alla validazione.
Quindi l'esecuzione di una determinata serie di operazioni richiede una quantità fissata di unità di Gas che è sempre la stessa.
Un utente può anche scegliere di pagare più unità di Gas necessarie all'esecuzione di un insieme di operazioni per richiamare l'attenzione di un miner.Tale quantità di unità non deve però superare un certo limite Gas Limit stabilito dal sistema per quell'insieme di operazioni.
Ad esempio per una transazione un miner non può impiegare più di 21k unità di Gas.Diversamente per uno smart contract che richiede generalmente più operazioni, non si possono impiegare più di circa 140k unità di Gas.
D'altro canto una transazione per cui viene impiegato meno gas del necessario viene terminata e le spese relative vanno perdute.
Dunque se un utente vuole essere tranquillo che la sua transazione vada a buon fine(non si ha sempre la certezza matematica) dovrà pagare almeno per la quantità di Gas necessario alla sua esecuzione.

Tornando all’esempio dello Smart Contract che implementa l’Halting Problem, il sistema forza la sua terminazione quando vengono impiegate per esso più di circa 140k unità di Gas.
Lo scopo del Gas non è solo quello di evitare attacchi come quest’ultimo ma anche quello di bilanciare l’uso delle risorse attribuendo un costo computazionale consono(che si rispecchia nel Gas) al tipo di operazioni eseguite.
Mentre in Bitcoin i miners ricevevano quantità di Satoshi come commissioni in Ethereum vengono ricompensati delle unità di Gas impiegato per le operazioni svolte.
Una singola unità di Gas ha un certo costo che si esprime in GWEY(un miliardesimo di ETH) che dipende fondamentalmente da quanto lavoro stanno svolgendo i miners in un certo momento.
Supponiamo che un utente che vuole effettuare una transazione base per cui il Gas Limit è 21k.

- Considerando il Gas Price pari a 43.17 GWEY(4.3*10^-8 CIOÈ circa 4.3 miliardesimi di ETH) in data 5/12/2022:
  per essere sicuro che la transazione venga eseguita l’utente dovrà dichiarare di essere disposto a pagare almeno:
  (4.3*10^-8)*(2,1*10^4)=8,4*10^-3=0.0084 ETH
- Considerando il valore di ETH pari a 1195 euro in data 5/12/2022 per effettuare la transazione l’utente pagherebbe circa 10.03 euro.

Nel caso in cui l’utente pagasse una per certa quantità di una di Gas.Sia questa quantità X,allora pagherebbe circa:
(4.3*10^-8)*X ETH.
Supponiamo che per la transazione occorressero esattamente 21k unità di Gas:

- se X fosse minore di 21K e per la transazione occorressero esattamente 21k unità di Gas questa non verrebbe validata e l’utente perderebbe gli ETH.
- se invece X fosse superiore a 21k e allora la transazione verrebbe validata e l’utente pagherebbe esattamente 0.0084 ETH(cioè il costo di 21k mila unità di Gas) e ciò che è avanzato gli viene restituito.