DISCLAIMER

Il materiale contenuto nel drive è stato raccolto e richiesto tramite autorizzazione ai ragazzi frequentanti il corso di studi di Informatica dell'Università degli Studi di Salerno. Gli appunti e gli esercizi nascono da un uso e consumo degli autori che li hanno creati e risistemati per tanto non ci assumiamo la responsabilità di eventuali mancanze o difetti all'interno del materiale pubblicato.

Il materiale sarà modificato aggiungendo il logo dell'associazione, in tal caso questo possa recare problemi ad alcuni autori di materiale pubblicato, tale persona può contattarci in privato ed elimineremo o modificheremo il materiale in base alle sue preferenze.

Ringraziamo eventuali segnalazioni di errori così da poter modificare e fornire il miglior materiale possibile a supporto degli studenti.

CoScienze
Associazione

Esposito

1. Consenso di Nakamoto:

- Ogni nuova transazione viene inviata in broadcast a tutti i nodi
- Ogni nodo colleziona le nuove transazioni in un blocco
- Ogni nodo cerca una PoW difficile per il proprio blocco. La PoW è il calcolo di un valore tale che un hash di blocco+nonce inizi con uno stabilito numero di zeri.
- Quando un nodo trova una PoW, invia in broadcast a tutti i nodi il blocco contenente le transazioni e la PoW.
- Un nodo accetta il blocco se: tutte le transazioni sono valide, le transazioni non sono relative a un bene già speso, la PoW è valida e a tale scopo il nodo calcola a sua volta l'hash del blocco e verifica abbia lo stesso numero di zeri iniziali.
- I nodi manifestano l'accettazione del blocco aggiungendolo alla blockchain all'atto della creazione del blocco successivo, utilizzando l'hash del blocco accettato per creare il prossimo blocco.

Le ***problematiche*** del consenso di Nakamoto sono che due nodi possono fare concorrentemente il broadcast di blocchi differenti, nodi diversi possono ricevere i due broadcast in ordine diverso, ciascun nodo lavora sul primo blocca che riceve, ma conserva l'altro ramo. L'indecisione si risolve quando un ramo diventa più lungo ed i nodi che lavorano sul ramo meno lungo lo abbandonano e proseguono a lavorare sulla catena più lunga.

1.1 Rispetto al teorema CAP quali proprietà ha il consenso il consenso di Nakamoto ?

Il consenso di Nakamoto è di tipo **AP** cioè garantisce l'**availability (A)**(disponibilità dei dati, ottengo una risposta da una richiesta fatta) ed è **Tolerance to network partition (P)**

2. Qual' è la differenza tra hard fork e soft fork ?

Un fork rappresenta uno stato incoerente che può essere sfruttato dagli avversari per causare confusione, transazioni fraudolente e sfiducia all'interno della rete. I fork possono essere creati involontariamente a causa di malfunzionamenti o incompatibilità negli aggiornamenti. I fork intenzionali possono essere

- Soft fork : si ha quando la modifica è retro compatibile.
- Hard fork : si ha quando i nuovi blocchi che la rete accetta sembrano non validi per i nodi pre-fork. Gli hard fork possono portare a una divisione della cryptovaluta.

3. Che cos'è l' attacco 51%

Un attacco del 51% avviene un singolo aggressore o un gruppo di nodi Sybil, di solito un numero ingente di **miner,** raggiunge la maggioranza dell'hashrate nella rete (Sybil Attack: quando il nodo finge di essere qualcun altro e porta i nodi onesti a seguire quello che dice il nodo sybil). Di

conseguenza, salta il concetto di rete decentralizzata, dato che il 51% del network è sotto il controllo del medesimo gruppo di **miner**, che potranno quindi manipolare la blockchain impedendo la verifica di transazioni o blocchi, invertire transazioni per consentire il double spending, creare biforcazioni, impedire ad altri miners di validare blocchi.

Double spending, ovvero spendere più volte lo stesso bene, tuttavia la parte attaccante non può modificare le informazioni nei blocchi già creati o generare nuove monete.

Le criptovalute il cui algoritmo di consenso si basa sul **PoW (Proof-of-Work)**, per proporre un nuovo blocco da aggiungere al ledger, un nodo deve dimostrare di aver usato abbastanza risorse di calcolo per risolvere un enigma matematico, il cui risultato viene messo all'interno del blocco stesso. La PoW implica la computazione di un valore di hash con un determinato numero di bit iniziali pari a zero. Ogni **miner**, cerca di trovare un nonce tale da soddisfare il vincolo sul numero di zero prodotti dalla funzione hash, e questo rende l'attacco del 51% molto oneroso.

Le criptovalute il cui algoritmo di consenso si basa sul **PoS (Proof-of-Stake),** sono difficilmente soggette ad attacchi del 51%,in tale algoritmo, i validatori contribuiscono a mantere le capacità operative del network in base alla quantità di criptovalute che possiedono **(stake)**. Non viene considerata la potenza di calcolo, di conseguenza ogni attacco risulta poco redditizio. Nella maggioranza dei casi, l'attacco del 51% viene eseguito su criptovalute più recenti, in quanto non è necessario possedere potenze di mining estremamente elevate.

Per esempio, l'attacco del 51% su Bitcoin o Ethereum, richiederebbe troppe risorse, risultando sconveniente e quindi praticamente impossibile.

5. Che cos'è la proof of stake (PoS)?

Proof of stake (PoS) è un'alternativa efficiente dal punto di vista energetico al PoW. Nel PoW la possibilità di proporre un blocco è proporzionale alla sua potenza di calcolo, nel PoS è proporzionale al valore del suo stake, o puntata si riferisce alle monete possedute. PoS evita la competizione di hashing bruteforce che si verificherebbe se fosse stato usato PoW ottenendo così una significativa riduzione del consumo energetico. Questo implica l'assenza del premio per chi inserisce il blocco giusto nella blockchain, e del mining, in quanto non vengono create nuove unità di criptovaluta con la creazione di ogni blocco. I validatori sono ricompensati con una commissione per le transazioni validate. Il DPoS (delegate proof of stake) consente ai nodi che detengono lo stake maggiore, stakeholders, di votare per eleggere i verificatori di blocchi. Questo fa si che i detentori di stake concedono il diritto di creare blocchi ai delegati che sostengono invece di creare blocchi stessi, riducendo così il loro consumo di potenza computazionale a 0. PoS ha lo svantaggio di accentrare le ricchezze nelle mani di pochi e nel caso di una fork i validatori saranno incentivati ad perare su entrambe le catene

6. Che cos'è la proof of work (PoW)?

Per proporre un nuovo blocco da aggiungere al ledger, un nodo deve dimostrare di aver utilizzato abbastanza risorse di calcolo per risolvere un enigma matematico, il cui risultato inserito nel blocco stesso. La PoW implica la computazione di un valore di hash con un determinato numero di bit

iniziali pari a zero. Ogni nodo (miner) cerca di trovare un nonce tale da soddisfare il vincolo sul numero di zero prodotti dalla funzione di hash.

7. Parliamo dell' attacco eclisse

Un gruppo di nodi dannosi isola i nodi vicini, compromettendo il traffico in entrata ed in uscita su di esse. Con un numero sufficiente di nodi compromessi, in un cluster, si possono isolare i nodi onesti, così che essi abbiano una visione sbagliata dello stato della blockchain.
Se un potente attacco riesce a dominare la comunicazione in entrata/uscita tra un miner vittima e la rete principale, allora la vittima non sarà più in grado di contribuire all'estensione della catena principale. L'attacco eclisse sfrutta la connettività debole, per risolvere bisogna aumentare la connettività e la diversità geografica delle connessioni peer-to-peer.

8. Che cos'è il selfish mining ?

Alcuni minatori tentano di aumentare le loro ricompense mantenendo privati i loro blocchi e determinando una block race tra la catena pubblica degli onesti e quella privata degli egoisti.
I miners onesti estendono la catena con blocchi validi, il miner egoista continua a estendere il suo ramo segreto fino a quando la catena pubblica è ad un passo indietro e pubblica la sua. Poiché la catena segreta è più lunga, le altre parti la considerano la catena principale. Se il numero di miner disonesti è più grande di 1/3 possono portare a termine l'attacco. Il selfish è possibile in blockchain piccole.

9. Che cos'è il finney attak ?

E' una variante del double spending. Il miner genera una transazione e non ritrasmette il blocco computato. Genera poi un dublicato della transazione precedente e lo invia ad un destinatario. Quando quest'ultimo accetta la transazione, consegna il prodotto, ed è qui che il miner pubblica il blocco precedente con la transazione originale, cosi facendo la transazione precedente viene invalidata, cosi il minatore spende il doppio.

10. Merkle-Tree che cos'è e a cosa serve:

Un ***Merkle-Tree*** è una struttura dati suddivisa in diversi livelli il cui scopo è metterli in relazione nodo con un'unica radice associata ad essi. Per ottenere ciò, ogni nodo deve essere identificato con un identificatore univoco (hash). Questi nodi iniziali, chiamati nodi figli (foglie), sono quindi associati a un nodo non superiore chiamato nodo genitore (ramo). Il nodo padre avrà un identificatore univoco risultante dall'hash dei suoi nodi figlio. Questa struttura si ripete fino a raggiungere il nodo radice o radice di Merkle, la cui impronta è associata a tutti i nodi dell'albero. Grazie a questa struttura unica, i Merkle-Tree consentono di mettere in relazione una grande quantità di dati in un unico punto il ***Merkle-Root***. In questo modo, la verifica e la validazione di questi dati può diventare molto efficiente, dovendo verificare solo la Merkle-Root invece dell'intera struttura. Quindi si va ad accelerare il processo di verifica di grandi quantità di dati.

10.1 Cosa serve il Merkle-Tree per la blockchain?

Un Merkle-Tree è una struttura che mette in relazione tutte le transazioni e le raggruppa tra coppie per ottenere un root hash che è correlato a tutti gli hash dell'albero. Selezionare tutte le transazioni di una rete sarebbe estremamente lento e inefficiente. Per questo motivo è stato implementato questo sistema, poiché, se un hash viene modificato, tutti gli altri cambierebbero fino a raggiungere la radice. Ciò invaliderà l'autenticità delle informazioni per l'intero albero. È proprio questa funzione che consente agli alberi Merkels di fornire l'elevato livello di sicurezza che li caratterizza.

10.2 E per i file come funziona ?

Nel Merkle-Tree abbiamo che un file viene suddiviso in chunck, ovvero delle sottoporzioni, e si calcola l'hash di ogni blocco utilizzando una funzione hash crittografica SHA256. Si combinano gli hash a coppie per creare un albero binario, ogni nodo memorizza l'hash del concat dei suoi figli. Nel momento in cui un utente vuole salvare un file su di un server per poi richiedere un blocco:

- L'utente crea l'MTR radice del Merkle-Tree dei dati del file iniziale D
- L'utente invia i dati del file D al server
- L'utente elimina i dati D, ma memorizza MTR (32 byte)
- L'utente richiede un blocco al server
- L'utente restituisce il blocco x e una breve prova di inclusione della radice π
- L'utente controlla che il blocco x sia incluso in MTR usando la prova della radice π

10.3 Perché non devo fare il semplice hash del blocco ma il Merkle-Tree?

Come detto in precedenza, in Merkle-Tree se un hash viene modificato, tutti gli altri cambierebbero fino a raggiungere la radice. Ciò invaliderà l'autenticità delle informazioni per l'intero albero. È proprio questa funzione che consente agli alberi Merkels di fornire l'elevato livello di sicurezza che li caratterizza.

11. Trust execution environment che cos'è e cosa realizza?

TEE combina parti hardware e software e consentono di dividere il sistema in due ambienti di esecuzione.

- Non-Trusted, chiamato Rich Execution Environment (REE), ovvero un ambiente di esecuzione dove ho la pienezza delle istruzioni.
- Trusted, il TEE dove posso eseguire solo le operazioni sensibili.

Il TEE fornisce un canale di comunicazione sicuro tra il processore e la periferica esterna. Se un'applicazione dannosa può intercettare l'input, può avere accesso a dati sensibili.

È inoltre necessario un processo di avvio sicuro, perché se riesco a cambiare il boot-strap posso manipolare e compromettere il dispositivo.

TEE ha anche una terza modalità chiamata Monitor, utilizzata per eseguire il salvataggio del contesto e il passaggio tra Rich e Secure OS.

La base per la realizzazione a livello hardware di TEE è costituita da **coprocessore**, che sono un processore specializzato a livello hardware e accelera il processo di cifrature/decifratura per una migliore sicurezza dei dati e protezione delle chiavi segrete.

Un processore generico (**GPP**) viene personalizzato e utilizzato per implementare alcuni algoritmi crittografici.

11.1 Come mai ha bisogno di un algoritmo del consenso?

Il TEE ha bisogno di un algoritmo di consenso come PoET che offre una soluzione al problema dei generali bizantini che usa un “ambiente di esecuzione affidabile” per migliorare l’efficienza delle soluzioni attuali come PoW. Poet sceglie in modo casuale i peer per eseguire le richieste e i comportamenti fraudolenti sono evitati da TEE e verifica di identità.

12. Tipo di Consenso poet (?)

PoET, Proof of Elapsed Time, è una soluzione al problema dei generali bizantini dove però l’esecuzione dell’algoritmo è in TEE, perché TEE non rende il codice alterabile esternamente, quindi i nodi non sono bizantini. PoET è un algoritmo crash-Tollerant , ma il fatto che viene eseguito in un ambiente TEE lo rende anche Bizantin-Tollerant.
PoET funziona essenzialmente come segue:

- Ogni validatore richiede un tempo di attesa da un'enclave
- Il validatore con il tempo di attesa più breve per un particolare blocco di transazione viene eletto leader.
- Una funzione, come "CreateTimer", crea un timer per un blocco di transazioni che è garantito essere stato creato dall'enclave.
- Un'altra funzione, come "CheckTimer", verifica che il timer sia stato creato dall'enclave. Se il timer è scaduto, questa funzione crea un'attestazione che può essere utilizzata per verificare che il validatore abbia atteso il tempo assegnato prima di rivendicare il ruolo di leadership.

Vantaggi di poet:

- È un algoritmo di consenso e generazione di blocchi estremamente efficiente e scalabile (molto più di PBFT).
- Il processo è perfetto da applicare a reti blockchain private, infatti è rivolto a spazi aziendali controllati.
- La selezione e l'accettazione dei validatori garantisce che la rete sia resistente agli attacchi esterni e interni.

13. Mi può parlare di paxos?

Paxos è un protocollo che serve a raggiungere il consenso. I nodi che partecipano a Paxos possono avere ruoli diversi, e l'esecuzione del protocollo è suddivisa in turni.

Proprietà del consenso di Paxos:
Liveness:

1. Uno tra i valori proposti prima o poi viene scelto.
2. Se un valore viene scelto, ogni processo prima o poi apprenderà tale scelta.

Safety:

1. Un valore può essere scelto solo tra quelli proposti.

2. Il valore scelto deve essere unico.
3. Un processo non deve mai apprendere che un valore è stato scelto a meno che esso non sia stato effettivamente scelto.

Ruoli dei processi

Abbiamo tre tipi di nodi:

- **Proposer**: sceglie un valore da votare e lo propone a tutti.
- **Acceptor**: riceve la proposta dal proposer, la vota oppure no.
- **Learner**: memorizza il valore votato dalla maggioranza.

Esecuzione del protocollo

Paxos funziona così:

1. Un **proposer** avvia un turno di votazione, inviando a tutti un messaggio che chiamiamo prepare.
2. Gli **acceptor** che lo ricevono, possono rispondere con una promise, impegnandosi quindi a votare per quel turno.
3. Il **proposer** sceglie un valore e lo manda a tutti con un accept.
4. Gli **acceptor** dicono ai **learner** di imparare quel valore con un learn.

Paxos non è usato solo per mantenere consistenza tra le repliche, ma soprattutto per la gestione del ***fault tolerant***.

Infatti, Paxos funziona fino al guasto del 50% degli acceptor. Aumentare il numero di acceptor aumenta i costi (più macchine) a fronte di un aumento di resistenza ai guasti.

14. Algoritmo di comunicazione gossiping

Il gossiping è un algoritmo distribuito per poter garantire la consegna (per questa proprietà lo usiamo in Fabric) di messaggi sebbene fallimenti di link, processo e di rete si possano verificare nel sistema. Realizza una soluzione di flooding selettivo:

A – periodicamente, ogni processo decide di inviare un messaggio a un insieme k di interlocutori scelti in maniera casuale,

B – sulla base dei messaggi scambiati, ritrasmissioni vengono attivate,

C – messaggi persi vengono recuperati. Esistono due

14.1. Caratteristiche del gossiping (?)

Esistono due modalità di interazione tra i partecipanti a un algoritmo di gossiping:

- **Push style**: il gossipier invia l'informazione a uno o più destinatari scelti casualmente;
- **Pull style**: il gossipier interroga uno o più destinatari richiedendo l'invio di un'informazione.

L'informazione che viene scambiata, o digest, può essere di due tipi:

- **Positive**, il messaggio di gossiping contiene gli identificativi dei messaggi correttamente ricevuti;
- **Negative**, il messaggio di gossiping contiene gli identificativi dei messaggi noti per essere stati persi.

Il gossiping può subire attacchi DoS per congestionare una porzione di rete o un determinato nodo. Una soluzione è usare primitive crittografiche, oppure considerare validi i messaggi di gossiping se ricevuti da un numero elevato di peer.

15. Gruppo chiuso e gruppo aperto

Bisogna prima fare una premessa, spiegare innanzitutto il **TEOREMA CAP** che ci dice che ogni sistema in rete che condivide dati può avere in un dato istante al più due delle tre proprietà desiderabili:

- **Consistency (C)** : avere una copia aggiornata dei dati
- **Availability (A)** : disponibilità dei dati
- **Tolerance to network partitions (P)**

Il teorema CAP si applica anche in blockchain dove nelle rali implementazioni della blockchain, non è mai possibile ottenere sia la Consistency che l'Availability, perchè devono affrontare la tolleranza alle partizioni.

- Nei sistemi **permissionless** in un ***gruppo aperto*** di nodi scelgono di privilegiare e quindi garantire l'Availability anziché la Consistency, per poter essere in grado di utilizzare, inviare e ricevere criptovalute. I casi di fork rappresentano i momenti di inconsistenza che sono risolti nel tempo, così da garantire la Eventual Consistency.
- Nei sistemi **permissioned** in un ***gruppo chiuso*** di nodi scelgono di privilegiare e quindi garantire la Consistency anzichè l'Availability, perchè non si è focalizzati sulle criptovalute ma sulla gestione di dati in ambito distribuito

16. Certification autority

Una **Certificate Authority (CA)** fornisce una serie di servizi di certificazione agli utenti di una blockchain. Nello specifico emette i certificati per consentire alle organizzazioni di autenticarsi sulla rete; alle applicazioni client di autenticare le proposte di transazione ; ai peer di approvare le proposte e caricare le transazioni nel libro mastro se valide.

17. Reliable Multicast?

I partecipanti si coordinano per garantire che il messaggio venga consegnato a tutti i destinatari. Soddisfa le proprietà di **integrity** (un processo corretto p riceve il messaggio m al più di una volta), **validity** (se un processo corretto p in via m in multicast, prima o poi riceverà m) e **agreement** (se un processo corretto riceve m, allora tutti i processi corretti in group(m) prima o poi riceveranno m).

L'Uniform Reliable Multicast implementa la proprietà uniform che fa si che la primitiva di comunicazione tiene conto anche dei fallimenti dei processi.

17.1 Eventual Reliable Multicast ?

Con l'Eventual Reliable Multicast abbiamo che l'eventuale consegna di tutti i messaggi a tutti i membri del gruppo avviene senza imporre alcun ordine di consegna particolare .

17.2 ATOMIC MULTICAST che cos' è ?

Atomic multicast è una primitiva di comunicazione che consegna messaggi a più gruppi di processi secondo un **ordinamento totale**, con ogni gruppo che riceve la proiezione dell'ordine totale sui messaggi ad esso indirizzati.

Ordinamento totale: se un processo corretto riceve i messaggi m e poi m' allora ogni altro processo corretto che riceve m' avrà ricevuto m. Non ci interessano i tempi di invio ma i tempi di ricezione.

18. Parlami del PBFT

Un algoritmo che tollera invece i guasti bizantini è il PBFT (Practical Byzantine Fault Tolerance). Il leader è scelto per ogni round dell'algoritmo. I nodi sono ordinati in base al proprio identificativo

- Tutti i risultati inviati al client devono essere uguali, altrimenti il client decide a maggioranza. Il client invia una richiesta al nodo primario (leader), che la trasmette a tutti i nodi secondari (backup) assegnando un numero di sequenza.
- I nodi secondari e il leader si accordano sull'ordinamento delle richieste, quando l'ordinamento è stato approvato, la richiesta viene eseguita e un risultato restituito al client.
- Se il client riceve f+1 risposte identiche, si raggiunge consenso.

18.1 In che modo PBFT riesce a tollerare fallimenti bizantini ?

L'algoritmo tollera f guasti anche di natura bizantina, con N maggiore o uguale di 3f+1, ovvero i nodi bizantini non riescono a far deviare il consenso raggiunto.

18.2 Rispetto al teorema CAP chi utilizza la PBFT quali proprietà garantisce?

La PBFT è un algoritmo CP ovvero il sistema Garantisce consistenza **(C)** (avere una copia dei dati aggiornati) ed è **tolerance to network partitions (P**).

18.3 Si può forkare una PBFT ?

Non è possibile forkare una PBFT in quanto questo algoritmo è CP ovvero il sistema è **tolerance to network partitions (P**). L'algoritmo infatti arriva sempre ad una soluzione consistente.

18.4 Quali sono le limitazioni del PBFT?

Il principale problema con PBFT è che richiede che i nodi verifichino la validità dei messaggi degli altri e che il numero di nodi attivi in un dato momento sia sempre noto. La sicurezza si basa su MAC (Message Authentication Code) e questo le rende inutilizzabile con oltre 1000 nodi

19. In bitcoin è possibile avere smartcontract ?

Essendo stateless rende complesso la realizzazione di applicazione come smart contract, quindi possiamo avere solo degli script.

20. Che cos'è un programma deterministico ?

Un algoritmo si dirà deterministico se per ogni istruzione esiste, a parità di dati d'ingresso, un solo passo successivo; in pratica esiste uno e un solo possibile percorso dell'algoritmo e quindi a fronte degli stessi dati di partenza produrrà gli stessi risultati.

21. Caratteristica del non ripudio

Chi effettua un'azione non può più tirarsi indietro, quindi chi invia e chi riceve i dati è strettamente responsabile delle sue azioni. Per realizzare questa caratteristica si usa la firma digitale, quindi chi compie un'azione non può dire di non averla fatta in quanto per inviare o ricevere cose usa la firma digitale.

22. Che cos'è lo State machine replication (SMR)?

State Machine Replication è un meccanismo a cui si ispirano le blockchain:

- Tutti i server iniziano con lo stesso stato iniziale;
- tutti i server ricevono la stessa sequenza di richieste secondo l'ordine di generazione dai client;
- Tutti i server che ricevono la stessa richiesta emetteranno gli stessi risultati di esecuzione e termineranno nello stesso stato.

SMR è spesso realizzato in maniera leader-based, con un server primario che riceve le richieste dai client e inizia la procedura di broadcast, mentre gli altri ricevono le stesse richieste e aggiornano il proprio stato locale in modo che corrisponda a quello del leader.

23. Che rapporto c'è tra blockchain e GDPR ?

Nel contesto della privacy e del GDPR, le blockchain presentano varie sfide:

- Come garantire i diritti del titolare dei dati immessi nella catena, fino al diritto di cancellazione se i dati sono immutabili?
- Chi risponde del trattamento dei dati in una rete di registri replicati?
- Il trattamento dei dati è tecnicamente solo quello che si conclude con la chiusura del primo blocco o è continuo?

la non modificabilità e l'impossibilità di cancellazione ed il fatto di garantire l'integrità dei dati e aumentare la fiducia nella rete , risulta essere in forte contrasto con requisiti legali come il "diritto di rettifica " e il "diritto all'oblio" sanciti nell'artt. 16 e 17.

- Soluzioni al diritto di oblio potrebbero essere: la crittografia dei dati personali e la successiva eliminazione delle corrispettive chiavi, lasciando su blockchain solo i dati indecifrabili;
- mediante l'uso dei cosiddetti modelli di memoria "fuori catena", registrare su blockchain solo un "collegamento", lasciando il dato vero e proprio al di fuori del libro mastro;
- garantire il diritto all'oblio mediante una corretta anonimizzazione.

Il modello di governance decentralizzato di Blockchain e la molteplicità degli attori coinvolti rendono più ostica la definizione dei ruoli.

I partecipanti, che scrivono sul canale e inviano dati alla convalida dei miners, possono essere considerati i responsabili. Qualora un gruppo di partecipanti decida di attuare un trattamento con

uno scopo comune, il responsabile va identificato tra loro. In caso contrario, tutti i partecipanti dovrebbero essere considerati come contitolari del trattamento. NON SONO DEFINIBILI TITOLARI I MINER E LE PERSONE FISICHE CHE IMMETTONO DATI PERSONALI NELLA BLOCKCHAIN IL RESPONSABILE DEL TRATTAMENTO VA RICERCATO TRA GLI SVILUPPATORI DEGLI S.C. E I MINATORI.

Per quanto rigurada l'identificabilità, la blockchain ha il vantaggio che ogni partecipante ha un identificativo costituito da una serie di caratteri alfanumerici apparentemente casuali e non identificabili con reali identità. Quanto ai dati aggiuntivi memorizzati sulla blockchain, nel caso in cui si tratti di dati personali, essi dovrebbero essere registrati preferibilmente in modalità crittografata.

È necessario determinare le finalità del trattamento e la valutazione d'impatto (DPIA), al fine di identificare come perseguibile l'uso di blockchain pubbliche, private, con o senza cifratura del contenuto dei blocchi, così da dimostrare i rischi residui di una scelta tecnologica e la loro accettabilità.

24. Qual'è la differenza tra privacy e sicurezza?

Il termine "protezione dei dati" è un termine fuorviante in quanto la privacy non protegge i dati. La privacy si riferisce all'utilizzo e all'utilizzo improprio delle informazioni da parte di utenti autorizzati. La sicurezza garantisce il controllo accessi alle risorse o alle informazioni di un'organizzazione. Avere un sistema sicuro non garantisce la privacy e avere un sistema che rispetta la privacy non vuol dire avere un sistema sicuro.

25. Che cos'è sistem design, seurity by design del GDPR?

Con l'espressione **security by design** si intende un approccio per lo sviluppo software e hardware che cerca di mettere i sistemi in sicurezza rispetto ad attacchi e vulnerabilità impreviste, attraverso misure come il monitoraggio continuo, l'utilizzo di credenziali e l'aderenza a pratiche di programmazione migliori.Nel GDPR, infatti, non solo si individua nel Titolare del trattamento il soggetto responsabile di garantire il rispetto dei principi applicabili al trattamento di dati personali ma si stabilisce che il medesimo debba essere altresì "in grado di comprovare". Sarà necessario, per esempio, essere in grado di documentare il processo che ha portato alla definizione del registro dei trattamenti, alla valutazione di un determinato rischio in materia di sicurezza, alla decisione di notificare o meno agli interessati una violazione dei dati personali, di aver attuato in relazione ad un nuovo trattamento le necessarie valutazioni legate alla **privacy «by design»**. Il principio della **privacy by design** richiede che la tutela dei diritti e delle libertà degli interessati con riguardo al trattamento dei dati personali comporti l'attuazione di adeguate misure tecniche e organizzative al momento sia della progettazione che dell'esecuzione del trattamento stesso.

26. Che cos'è la pseudo-anonimizzazione?

La pseudo-anonimizzazione è intesa come un particolare trattamento dei dati personali realizzato in modo tale che i dati stessi non potranno più essere attribuiti direttamente ed automaticamente ad un interessato specifico. Infatti, tali dati potranno essere ricondotti all'interessato cui si riferiscono solo attraverso l'impiego di altre informazioni aggiuntive, che dovranno essere, a tal fine, conservate separatamente e con l'impiego di precauzioni tecniche e organizzative adeguate

27. Che cos' è la sincronizzazione del tempo in un sistema distribuito?

La sincronizzazione del tempo fisico richiede la sincronizzazione dei clock delle cpu coinvolte nel sistema distribuito. Garantire che i diversi clock di diversi nodi siano uguali è impossibile. Di conseguenza, gradualmente i clock derivano uno rispetto l'altro. La differenza di valori tra due diversi orologi è detta clock drift rate. Pertanto possiamo distinguere i sistemi distribuiti in sistema sincrono, sistema asincrono o sistema parzialmente sincrono.

SISTEMA DISTRIBUITO SINCRONO: per natura sono sistemi piccoli, è definito tale se esistono e sono noti i limiti inferiori e superiori al tempo di esecuzione di ogni passo di elaborazione, il limite superiore al tempo di consegna di un messaggio, il limite superiore al tasso di deviazione di ciascun orologio locale (clock drift rate). Se non è possibile garantire tali valori limite. Vantaggi: è possibile definire algoritmi distribuiti basati sull'individuazione dei fallimenti tramite time-out. Svantaggi: è difficile assicurare tali proprietà in un sistema su grande scala e nel tempo. Tipicamente i sistemi reali sono asincroni e si tenta di trasformarli in sistemi parzialmente sincroni.

SISTEMA DISTRIBUTO ASINCRONO: un sistema è asincrono se non esistono limiti alla velocità di esecuzione de processi, al ritardo di trasmissione dei messaggi, o alla deviazione degli orologi. Non è possibile formulare ipotesi temporali relativamente all'elaborazione, allo scambio messaggi e alla sincronizzazione. Alcuni problemi non hanno soluzione nei sistemi asincroni.

28. Parliamo del BGP hijacking

BGP (Border Gateway Protocol) è il protocollo con cui si scambiano informazioni sulle rotte che i pacchetti dati devono seguire per raggiungere le varie destinazioni. Le tabelle di routing vengono scambiate tra i router usando proprio BGP. La pratica del **BGP hijacking** mira a modificare le tabelle di routing per fare in modo che soggetti non autorizzati possano ricevere il traffico dati originariamente destinato ai server legittimi.

29. È sempre possibile raggiungere il consenso in un sistema distribuito ?

E' possibile raggiungere consenso in un sistema distribuito solo nel caso in cui sia sincrono e siano soddisfatti i requisiti di:

- TERMINAZIONE (TERMINATION): prima o poi ogni processo corretto prende una decisione;
- ACCORDO (AGREEMENT): due qualsiasi processi corretti non decidono diversamente;
- INTEGRITÀ (INTEGRITY): se tutti i processi corretti propongono lo stesso valore, la decisione finale di ogni processo corretto corrisponde a quel valore.

Non esiste nessun algoritmo deterministico in grado di garantire il raggiungimento del consenso in un sistema asincrono a scambio di messaggi.

De santis

1. Paradosso del compleanno

Il paradosso del compleanno afferma che la probabilità che almeno due persone in un gruppo compiano gli anni lo stesso giorno è, in un gruppo di almeno 23 individui scelti a caso, è almeno ***½*** nello specifico ***0,51 (51%).***

Supponiamo che ***h: X → Z*** sia la nostra funzione hash, che ***il valore assoluto di X è uguale a m*** (***|X| = m***) ***e il valore assoluto di Z uguale a n*** (***|Z|= n***), con **n** ed **m** finiti e tali che ***m >= 2n***. Banalmente il numero delle collisioni è almeno **n** . Un approccio ingenuo alla ricerca di collisioni consiste nello scegliere **k** valori casuali distinti, $x_1 \ldots x_k$ appartenenti a ***X***, calcolare z di i uguale all'hash di x di i ($z_i = h(x_i)$) per **i *compresa tra 1 e k (1 <= i <= k)*** e verificare se c'è una collisione. Usando questo metodo possiamo calcolare un limite inferiore alla probabilità di trovare collisioni. Questo limite inferiore dipenderà da **k**, da **n** ma non da **m**. Dal momento che siamo interessati al limite inferiore supponiamo che ***h alla meno 1 di z è approssivatimavente uguale a m fratto n*** ($h^{-1}(z) \approx m/n$) ***per ogni z appartenente a Z*** **(z ∈ Z),** ossia, che la dimensione delle immagini inverse di h siano alquanto bilanciate. Sotto questa ipotesi infatti la probabilità che ci siano collisioni è più bassa rispetto al caso in cui le immagini inverse non siano bilanciate.

2. Funzione di hash e paradosso del compleanno

La funzione hash restituisce una stringa di poche centinaia di bit a partire da una sequenza di bit di lunghezza arbitraria. Un blocco contiene l'hash del blocco precedente e le funzioni di hash crittografiche cercano di rendere computazionalmente inammissibile trovare due sequenze che producono la stessa impronta. La possibilità che 2 sequenze di bit sottomesse alla stessa funzione di hash dia origine allo stesso valore (collisione) è definita "**Paradosso del compleanno**".

Nell'ambito della crittografia si utilizza il paradosso del compleanno per indicare che le funzioni hash crittografiche abbiano la proprietà di "resistenza forte alle collisioni".

3. Che cos'è una collisione?

Una collisione è una situazione che avviene quando due diversi input tramite una funzione hash producono lo stesso output.

4. Firma digitale

La **firma digitale** è un metodo teso a dimostrare l'autenticità di un messaggio o di un documento digitale inviato tra mittente e destinatario attraverso un canale di comunicazione non sicuro, garantendo al destinatario che:

- il mittente del messaggio sia chi dice di essere (autenticazione),
- il mittente non possa negare di averlo inviato (non ripudio)
- il messaggio non sia stato alterato lungo il percorso dal mittente al destinatario (integrità).

La firma digitale garantisce 3 proprietà:

- Facile da produrre
- Difficile da falsificare
- Facile da verificare

Si parla di 4 firme diverse:

- **elettronica (semplice o debole) FES o FEC**: dati in forma elettronica, acclusi oppure connessi tramite associazione logica ad altri dati elettronici e utilizzati dal firmatario per firmare". Si riferisce a password, pin, tecniche biometriche, una scansione della firma olografa e più che una firma vera e propria rappresenta un metodo di autenticazione.
- **elettronica avanzata FEA**:
    - è connessa unicamente al firmatario;
    - è idonea a identificare il firmatario;
    - è creata mediante dati che il firmatario può utilizzare sotto il proprio esclusivo controllo;
    - è collegata ai dati per identificare ogni successiva modifica.
- **elettronica qualificata FEQ** : una firma elettronica avanzata creata da un dispositivo per la creazione di una firma elettronica qualificata e basata su un certificato qualificato per firme elettroniche
- **firma digitale FD:** un particolare tipo di firma qualificata basata su un sistema di chiavi crittografiche, una pubblica e una privata, correlate tra loro, che consente al titolare di firma elettronica tramite la chiave privata e a un soggetto terzo tramite la chiave pubblica, rispettivamente, di rendere manifesta e di verificare la provenienza e l'integrità di un documento informatico o di un insieme di documenti informatici.

5. Se ho un documento di X byte, come funziona la firma digitale?

Tecnicamente inserire una firma digitale in un documento elettronico significa:

- **Generare l'impronta digitale:** Nella fase iniziale, viene generata l'impronta digitale usando la funzione di hash, che è un algoritmo che produce una stringa di lunghezza piccola e costante (generalmente 120 o 160 bit) ***indipendentemente dalle dimensioni dell'originale***.
  L'impronta per ogni documento è unica e non invertibile: questo significa che modificando anche un solo carattere del testo si otterrà un impronta diversa.
- **Generare la firma:** viene applicata la cifratura con chiave privata dell'impronta digitale precedentemente generata: la firma sarà quindi legata, da un lato (tramite la chiave privata usata per la generazione) al soggetto sottoscrittore, e dall'altro (tramite l'impronta) al testo sottoscritto. L'impronta, e non l'intero documento, viene in questa fase criptata con la chiave privata del mittente ottenendo la generazione della firma digitale.
- **Apporre la firma:** viene aggiunta la firma del sottoscrittore in una posizione predefinita, generalmente alla fine del documento.

Al destinatario del documento verranno spediti:

- il documento firmato secondo la procedura tecnica descritta
- il certificato che deve essere rilasciato dall'ente di certificazione a garanzia della titolarità della chiave pubblica necessaria a decriptare la firma digitale

6. Come si firma l'hash

Il valore hash di un messaggio è una rappresentazione non ambigua e non falsificabile di un messaggio. Le caratteristiche della funzione hash sono:

- comprime
- è efficiente
- sicurezza forte: è difficile trovare 2 messaggi diversi che hanno lo stesso valore hash
- one way: facile trovare il valore hash associato ad un messaggio ma è difficile trovare il messaggio a cui è associato il valore hash

Dato che una funzione hash comprime allora si prende il messaggio di grandi dimensioni e si calcola la funzione hash ottenendo un messaggio di dimensioni ridotte su cui si può invocare l'algoritmo di firma. Nel caso specifico della firma digitale:

Il digest, ossia la rappresentazione unica e compatta delle informazioni originali contenute nel documento firmato, nonché del documento stesso quindi l'output dell'applicazione dell'algoritmo di hash, rappresenta l'impronta informatica del documento sottoposto a firma digitale e rappresenta anche l'impronta informatica dell'intero archivio digitale.Il codice di hash è usato per implementare la marcatura temporale: in questo modo, sarà sempre possibile attestare che la firma digitale è stata apposta nel periodo di validità del documento in questione. DA CANCELLARE

7. Differenza tra firma digitale e MAC

I **MAC** differiscono dalle firme digitali in quanto sono sia generati che verificati utilizzando la stessa chiave segreta. Questo implica che il mittente e il destinatario del messaggio devono scambiarsi la chiave prima di iniziare le comunicazioni. La firma digitale invece si basa sulla tecnica della **crittografia asimmetrica** o **a doppia chiave** (una privata e una pubblica): ***la chiave privata*** servirà al sottoscrittore per codificare il documento elettronico da firmare e ***quella pubblica*** servirà al destinatario per decodificare il messaggio ricevuto. Dato che la chiave privata è nota solo al suo possessore, una firma digitale prova che un documento è stato firmato esattamente dal suo proprietario e da nessun altro

8. Che vuol dire autenticazione? (user autentication e message autentication MAC)

L'autenticazione è l'atto di confermare la verità di un attributo di una singola parte di dato o di una informazione sostenuta vera da un'entità. In contrasto con l'identificazione, che si riferisce all'atto di confermare l'identità di una persona o qualcosa, l'autenticazione è il processo di confermare davvero l'identità. Quindi l'autenticazione è il processo tramite il quale un sistema informatico, un computer, un software o un utente verifica la corretta, o almeno presunta, identità di un altro computer, software o utente che vuole comunicare attraverso una connessione, autorizzandolo ad

usufruire dei relativi servizi associati. È il sistema che verifica, effettivamente, che un individuo è chi sostiene di essere. L'autenticazione è diversa dall'identificazione (la determinazione che un individuo sia conosciuto o meno dal sistema) e dall'autorizzazione (il conferimento ad un utente del diritto ad accedere a specifiche risorse del sistema, sulla base della sua identità).

9. Che tipo di crittografia simmetrica o asimmetrica è meglio usare ?

Una ragione per cui la crittografia asimmetrica è spesso considerata più sicura della crittografia simmetrica è che la prima, a differenza della sua controparte, non richiede lo scambio della stessa chiave di cifratura-decifratura tra due o più parti.

10. Che cos'è DES

DES è un algoritmo a chiave comune orientato alla codifica di blocchi di bit di input di lunghezza prestabilita. Il messaggio M da codificare deve essere preliminarmente suddiviso in blocchi da 64 bit. Nell'algoritmo DES tradizionale è previsto l'utilizzo di una chiave comune di lunghezza pari a 56 bit. L'uscita del modulo di codifica è costituita a sua volta da blocchi da 64 bit , per cui sia il messaggio di entrata che quello in uscita (cifrato) risultano della stessa lunghezza.

11. DES è sicuro o è stato superato?

E' l'algoritmo di cifratura simmetrico più utilizzato fino a poco tempo fa. Attualmente DES è considerato insicuro per moltissime applicazioni. La sua insicurezza deriva dalla chiave utilizzata per cifrare i messaggi, che è di soli 56 bit. L'algoritmo è ritenuto sicuro reiterandolo tre volte nel Triple DES, anche se in teoria così è esposto ad alcuni attacchi. Negli ultimi anni DES è ***stato sostituito dal'AES*** che elimina molti dei problemi del DES.

12. Nel des qual' è la lunghezza del blocco e quale quella della chiave?

Nel DES la **dimensione del blocco** è di 64 bit. Il DES usa inoltre una chiave per modificare la trasformazione in modo che l'operazione di decifratura possa essere effettuata solo conoscendo la chiave stessa. La **chiave è lunga 64 bit ma solo 56** di questi **sono effettivamente utilizzati** dall'algoritmo.

13. Come funziona l' implementazione dell' algoritmo DES?

DES è un algoritmo a chiave comune orientato alla codifica di blocchi di bit di input di lunghezza prestabilita. Il messaggio M da codificare deve essere preliminarmente suddiviso in blocchi da 64 bit. Nell'algoritmo DES tradizionale è previsto l'utilizzo di una chiave comune di lunghezza pari a 56 bit. L'uscita del modulo di codifica è costituita a sua volta da blocchi da 64 bit , per cui sia il messaggio di entrata che quello in uscita (cifrato) risultano della stessa lunghezza.

Analizzando la struttura interna dell'algoritmo DES, è possibile individuare tre elementi di base:

- un modulo utilizzato per effettuare una permutazione iniziale (Diffusion)

- 16 round, ciascuno dei quali rappresenta una rete di Feistel (algoritmo di cifratura a blocchi), per effettuare le specifiche trasformazioni previste dall'algoritmo (ciascuna iterazione introduce sia Diffusion che Confusion)
- un modulo per effettuare una permutazione finale (Diffusion)

Ciascuna iterazione utilizza una sottochiave diversa {K1, K2, ...K16} a 48 bit. Le sottochiavi sono ottenute tramite successive operazioni di permutazione e shift circolare a partire dalla chiave iniziale. La chiave iniziale a 56 bit viene artificialmente portata a 64 bit, tramite l'aggiunta di un bit di parità ogni 7 bit. L'operazione di decodifica del DES, segue essenzialmente la stessa serie di passi descritti per l'operazione di codifica. In tal caso i dati in ingresso sono costituiti da un blocco di 64 bit di testo cifrato. Le sottochiavi vengono adesso utilizzate in ordine inverso {k16, k15, .k1). L'uscita dal modulo di decodifica coincide con i 64 bit di messaggio originario.

Lo standard DES utilizza una funzione F composta dai seguenti blocchi:

- un E-BOX che realizza una permutazione e espansione di R_i da 32 a 48 bit;
- una operazione di XOR con la chiave K_i ;
- 8 distinti S-BOX paralleli che realizzano una sostituzione e compressione da 6 bit a 4 bit;
- un P-BOX che realizza un'ulteriore permutazione.

Ciascun S-BOX implementa una diversa operazione di sostituzione. Poiché non sono mai stati resi noti i dettagli che illustrano come e perché si è arrivati a realizzare simili blocchi, il livello di sicurezza dello standard è stato molte volte messo in discussione.

14. Che cos'è AES ? Come faccio a scegliere una lunghezza rispetto ad un'altra ?

AES, è un algoritmo di cifratura a blocchi a chiave simmetrica. L'AES include tre cifrari a blocchi e ciascuno di questi cifrari a blocchi ha un numero diverso di possibili combinazioni di tasti, come segue:

- AES-128: lunghezza chiave a 128 bit
- AES-192: lunghezza chiave a 192 bit
- AES-256: lunghezza chiave a 256 bit

Anche se esistono tre cifrari a blocchi, ognuno di essi crittografa e decrittografa i dati in 128 blocchi bit utilizzando chiavi di lunghezza diversa (come specificato sopra). Quindi è sicuro dire che anche se la lunghezza delle chiavi può essere diversa, la dimensione del blocco è sempre la stessa (128 bit o 16 byte, sono la stessa cosa).

La dimensione della chiave è un elemento critico, dato che una chiave troppo corta sarebbe esposta al rischio di "forzatura" tramite un attacco a forza bruta. Quindi scegliere una chiave troppo corta non è l'ideale per questo tipo di algoritmo, possiamo dire però che AES 128 bit è sicuro contro la tecnologia moderna e che AES 256 è sicuro contro ogni probabile tecnologia futura.

14. Che cosa sono i cifrari a flusso (stream cipher) (non li ha spiegati)?

In crittografia un cifrario a flusso (detto anche cifrario a caratteri) è un cifrario simmetrico nel quale i simboli (i bit) che codificano il testo in chiaro sono cifrati indipendentemente l'uno dall'altro e nel quale la trasformazione dei simboli successivi varia con il procedere della cifratura.

15. Funzione di HASHING

Una funzione HASH è una funzione che prende in input una lunghezza arbitraria di bit e da in output una lunghezza compressa di tali bit. In particolare l'idea è che dato un messaggio il valore HASH ad esso associato è un valore non falsificabile e non ambiguo.

Le funzioni HASH possono essere utilizzate in diversi ambiti:

- ***FIRMA DIGITALE***
- ***INTEGRITA' DEI DATI:*** quando salvo un documento memorizzo il suo valore hash, quando lo riapro calcolo il valore hash del documento aperto e vedo se tale valore è uguale a quello memorizzato

Una funzione HASH ha ***tre*** caratteristiche:

1. ***SICUREZZA DEBOLE***: dato un messaggio M è difficile trovare un altro messaggio M con lo stesso valore hash cioè tale che **h(M) = h(M')**
2. ***SICUREZZA FORTE:*** è difficile trovare due messaggi M e M' con lo stesso valore hash (l'hash deve essere di almeno 2^{160})
3. ***ONE WAY***: facile da calcolare difficile da invertire

15.1 La confidenzialità può essere garantita con l'hashing?

Gli hash non possono essere utilizzati per scoprire il contenuto del messaggio originale o una qualsiasi delle sue altre caratteristiche, ma possono essere utilizzati per determinare se il messaggio è cambiato. In questo modo, gli hash forniscono confidenzialità, ma non integrità.
Dove per confidenzialità si intende che l'accesso alle informazioni è possibile solo alle persone autorizzate (ottenuta tramite la cifratura del documento, quindi una volta cifrato solo chi avrà la chiave di cifratura corretta potrà accedervi).