WHITEPAPER

Contents

RelateID

RelateID

A Secure, Scalable, Trust-Based Personal Data Blockchain For the World

RelateID is an agnostic global identity-centric blockchain aimed at self-sovereign ownership of one's identity, including all data, documentation, and entities associated with an individual. It returns ownership of an individual's personal information to the individual by facilitating the access, control, and use of the information through a multi-layered permission system.

RelateID's blockchain technology and permissioning system underpin the security, distribution, and storage of personal data to create a decentralized, highly scalable data store for individuals. This technology also enables independent organizations and companies to integrate their bespoke distributed applications with RelateID and interact with self-sovereign individuals in a mutually beneficial way to gain access to personal information on the individual's terms. RelateID allows individuals to choose the data that is allowed to be viewed by any distributed application with an interest in using the personal data.

RelateID deploys tried and tested open-source solutions. When combined these solutions deliver a scalable self-sovereign identity network for personal data that leverages speed, throughput, reliability, high-security, and manageability of personal data with blockchain immutability.

INTRODUCTION

Overview

Using a codependent distributed application individual users will manage their own interactions with the outside world much like social networks do by employing an Identity verification algorithm that utilizes their social connections in the RelateID network to validate their identity. This will create an identity profile score that will indicate the confidence in their realness as an individual. The realness of their other connections in their social network, will serve as a decentralized identity validation system. RelateID will also use trusted third party ID verification to improve the realness scores of individuals within the system.

Why

Privacy is one of the main concerns in today's digital economy. It has become clear through the many data breaches over the last decade, that personal information is vulnerable to theft from third party databases. Individuals have no control over the security of their information and little recourse once a breach has occurred. RelateID will allow individuals to manage their personal data and the documentation associated with their identity, including their social networks, educational credentials, work history, property ownership, equipment and vehicle ownership, creative works, health history, financial history, legal documents, shopping history,digital asset ownership, and much more. It will reverse the concentration of power that has developed around Internet services today by building a truly decentralized system that is open and allows other decentralized applications to develop around it. RelateID removes the honey pots from the table, allowing greater security through the use of individualized data ownership. Self-sovereign identity offers individuals the ability to decide what the world can and cannot know about them.

How

RelateID will facilitate the self-sovereign ownership of personal data using a built in distributed database, featuring enterprise-hardened,scalable, decentralized NoSQL database technology, with built in NoSQL query capability and a robust permission system to manage data access. To answer the questions, "are you a human," "is this your sole identity," "can you be trusted," RelateID will employ a novel proof of trust / proof of realness / proof of stake consensus algorithm. The algorithm will incorporate not only the relative trust in a person's realness, but also the stake they hold in the well-being of the network in terms of the base asset ReID of the RelateID network Permission to validate blocks will be determined by an individual's trustworthiness, and a violation of trust on the network will result in not being able to participate in the validation of future blocks. Creation of the base asset ReID will follow a protocol that will create new ReID's for each newly tagged or verified identity that reaches a defined realness threshold, as well as for each block created. Rewards will be given to identity validators if (1) they validate by tagging a new person's identity and the validators identity realness score is over a predefined score, and (2) the new identity they are tagging reaches a minimum realness score. Validation tagging and asset creation processes will encourage individuals to improve their ID verification score, and thus their ability to tag new individuals. This approach will also ensure maximum decentralization of the identity verification system, as only first connections will be able to tag other individuals on the blockchain, and only by invitation or request of the new individual who is creating their profile. The RelateID application will ask new users if they would like to pull names from their social networks to locate identity validators on the system, as well as invite others to sign on. Node operators, will receive ReID's for each new verified block.

Who

RelateID will initially support a number of base identity types listed below. As new types of entities would like to interact with individuals on the network, the open source development community will ensure that the number of base identity types can be increased.

Self-Sovereign Identity

The first and most fundamental entity on the blockchain will be a self-sovereign Identity of an individual. This self-sovereign Identity will serve as the basis for the trust network that will become evident as individuals interact with each other.

Entities, representing Companies and Organizations

The second type of entity will be an entity that can be created by an individual with a sufficient trust score, and can then have additional members each assigned votes by the initial creator.Individuals who own Voting shares, can then create custom second-level digital assets or tokens within these entities, as long as enough votes support the creation.

1. Votes
   Votes will take the form of a second level digital asset which will be created initially when a new entity, representing a new company, or organization is created by an individual, and will be analogous to shares in a company or organization. The more votes assigned to an individual with a RelateID identity the more voting power they have in the decision making processes of the companies or organizations.

2. Tokens
   Additional custom digital assets can then be created and assigned arbitrarily to any address on the blockchain by transferring them to the address, these will be analogous to ERC20 tokens on the Ethereum network, that is, it would represent a custom currency for the new company, or organization.

ASSETS

Complex Immutable Non-Fungible Transferable Assets

In addition to Votes, and Tokens, there will be many other assets in the real world that will need to be associated with a self sovereign identity, or with entities performing the role of an organization or company.

Property

A standard form of property asset will be created to represent the analog of a title deed, and will be transferable.

Movable Assets

Any movable asset that has a unique identity assigned to it either by a manufacturer, or through government licensing oversight, such as vehicles or X-Ray machines, for example, which would benefit from transfer of ownership on a public register can be tracked by creating the asset in the blockchain, and using its unique ID numbers during the creation of the asset. The blockchain will not allow duplicates for these assets, and these assets can be created and issued by governments and organizations as complex second level digital assets.

Intellectual Property

Many forms of digital intellectual property can be created and managed by individuals, or entities. Music,art, and literature, are examples of this. Files can be stored on the distributed global file system known as Inter Planetary File System or IPFS, and a unique asset can be created to track ownership of the asset. The location of the asset in IPFS can be encrypted into the asset token, and transferred to anyone on the network. The location can only be acquired by the owner of the token, and each time the token is transferred, the location of the file on the IPFS known as the hash of the file can be changed, and the old hash removed to ensure that no one is able to access the files without permission. Any copies will be revealed as copies and not the original, based on the original time and date of creation of the original file.

Personal Records

Self-sovereign record keeping will form a key part of how individuals store, manage, and share their private information. Financial reports, medical history, privileged correspondence, and on-line commercial activities can be retained securely for access and sharing ONLY on an individual's own terms and ONLY with authorized parties. Third party entities will be able to develop their distributed applications to allow for the retention of data by their customers, and will use incentives based on second level digital assets underpinning their own business models.

Education Certification

RelateID can store self-sovereign education records. These can be issued by educational institutions as a second-level asset, signed, and sent to an individual or entity to prove the authenticity of the certification based on trust in the issuer.

Contract Signing and Management

Human readable contracts can be stored alongside recorded transactions, allowing immutable records of agreements between parties.

Financial Assets

Ownership of financial assets belonging to entities and individuals can be tracked securely without the need for third parties through distributed exchanges, which can run as distributed applications associated with RelateID.

Smart Contracts

Smart contracts can be stored and validated for the use of third-party distributed applications, which interact directly with individuals and associated entities.

BACKGROUND

Distributed Databases

Our modern world of Google search, YouTube, NetFlix, and Facebook would not be possible were it not for the massively distributed system they run on. To deliver such services at the scale of billions of users, every one of these ubiquitous services needs to have a robust and reliable distributed database underpinning every piece of data traversing the system. These databases use many of the features that are present in blockchains, including consensus algorithms to determine which data is most up to date and must take precedence over data that may be old or no longer relevant. The infrastructure that runs our Internet services has in fact been using key features that make blockchains important for a lot longer than most people actually think. In fact, the first consensus-driven distributed databases implemented solutions to the Byzantine Generals' problem in which a system could withstand arbitrary faults with random nodes, including deliberate tampering with data on the nodes, were developed as far back as the early 1980s. These first implementations were eventually replaced by morerobust solutions, the most successful of which was Paxos by Lamport introduced in 1998. Every major company now implements some version of Paxos, including Google, IBM, Microsoft, and many more, proving its relevance and reliability for almost two decades.

Conclusion by Fischer, Lynch and Patterson, 1985

In 1985, Fischer, Lynch and Paterson published a paper, which concluded that no asynchronous consensus system could tolerate benign faults without failing completely. This led to the development of systems that needed some synchronicity built into the core protocol to ensure that at some point in the future thesystem could reach consensus. Bitcoin successfully implemented this form of consensus and that has led to the accepted practice of waiting for at least three confirmations of the state of any transaction before releasing funds. This is because there is still a small but significant possibility that diminishes over time that Bitcoin network could revert to another set of transactions propagating in a side chain on some part of the network's nodes.

Solutions by Castro and Liskov, 1999

Castro and Liskov solved the problem in a paper titled "Practical Byzantine Fault Tolerance 1999," in which they describe methods to improve the speed and reliability of distributed databases and allow for faster consensus across the network of nodes.

Modern Implementations of Distributed Databases

With the development of faster and more reliable distributed databases with better consensus mechanisms underpinning the integrity of the data, it became possible to split the full data set across multiple nodes instead of every node having a full copy of all the data. In a calculated trade off between network failure probability, partial replication of the data allowed for gains in storage capacity, minimization of network traffic, and lowering of the average hardware requirements of each node. This Replication Factor, allowed more cost-effective distribution of large global databases, with very high reliability and massive gains in data delivery speeds from multiple nodes simultaneously. Ironically, Bitcoin implements no Replication Factor. Future blockchain implementations will have few, if any, of the scalability concerns experienced by the current most popular blockchains, such as Ethereum and Bitcoin. The new consensus mechanisms, along with replication factors and higher speed database driven blockchains like RelateID will become the dominant blockchain implementations in the coming years.

PERFORMANCE

Robust Performance and Fault Tolerance

RelateID implements scalable, fast, decentralized trust using consensus, with robust fault tolerance in the following scenarios:

Sybil Attack

Utilizing an intrinsic Realness Score for each individual running nodes in a federation of voting nodes, the possibility of a Sybil Attack is minimized due to the benefits of the block reward out waying the high cost associated with the loss of trust and the reputation of the node operator. Node operators are associated with a real identity, and thus attacking the system will lead to that individual being excluded from the option to run a full node which will exclude the individual from further block rewards for voting on blocks.

Benign Fault Tolerance

The underlying fault tolerance mechanism is based on Paxos, and thus can withstand the loss of half the nodes and still maintain consensus in the order of transactions.

Byzantine Faults

By only allowing individuals with highly validated trust to operate full nodes, there is a high degree of trust associated with each full node. This trust minimizes the possibility of a Byzantine Attack on the system by minimizing collusion and punishing any individual by excluding them from further participation in voting of blocks or the operation of a full node. In addition, trusted nodes will increase over time, based on the elimination of potential attacking nodes.

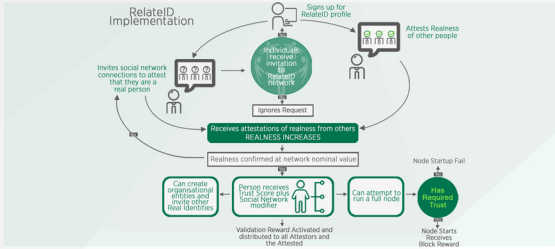
INCENTIVE MODEL

Native Asset RelD, and Incentive Model

The native asset RelD, will form the basis of the incentive model which is based on proof of stake, proof of realness and proof of trust.

Proof of Stake

Individuals will be able to stake ReID to claim a share of the block rewards. The higher the stake, the higher the block reward. In order to discourage individuals from running multiple nodes, a slight incentive will need to be added that makes it more profitable to stake more on a single node associated with a single identity. In addition, the higher your trust score the greater portion of the reward will be assigned to thatindividuals node.



Realness and Trust

Realness - This is a number that will indicate the likelihood that an individual with an identity on the RelateID network is an actual person, and that their relationships with other individuals are likely to be real as well.

Trust - Trust is a number that will indicate whether the individual associated with the identity on the RelateID network is trustworthy with respect to their conduct toward the security on the network.

An individual will start off with a Trust Score directly related to the Realness Score generated when verifying their identity through the process of being tagged by others.

Proof of Trust

RelateID will employ a proof of realness algorithm, which will build a score that is essentially the system's confidence that you are a real person based on the use of validation and social networking. It may eventually be possible to use Zero Knowledge Proofs, so the system will be able to perform validation between individuals without them revealing their public key to the network. The result will be a score associated with every address on the network, which will indicate the realness and trustworthiness of the individuals associated with these addresses. Individuals will beable to share their actual ID with others, but only if they wish to. Identities will by default not be known outside of the permissions granted by an individual within their validation network.

To incentivize individuals to validate others by tagging their' IDs, the system will also reward all individuals who vouch for someone by tagging them. Once their realness score reaches a level in which the system deems them real enough by consensus of the vouching individuals. The incentive will be divided based on your own realness and trust score, so it incentives individuals to maintain their own realness score.

Over time, a person's Trust Score can change, and may be lowered based on an individual's conduct on the network. In the case of double-spend attempts for instance, an individual's trust score can be lowered, even though they still have a high likelihood of being a real person and will be able to continue using the network, but with less power to affect it in a negative way. They will thus be incentivized to recruit more validators, in order to improve their trust scores thus improving the reach of the network. If a person intends to run a full node their Trust Score will have to be sufficiently high before they will be able to run the node. The higher the individual's Trust Scores in their validation network, the greater their trust and realness score on the RelateID network. This will incentivize individuals to choose wisely, and only include other individuals whom they themselves feel are trustworthy, and thus will not negatively affect their own trust score. This should lead to a logarithmic distribution within the users of the ID system with regard to trust, and thus bad actors will find it extremely hard to include high trust individuals in their validation network. It will be possible to withdraw yourtag vote from any individual at any time.

SCALABILITY

Second-level Native Assets

RelateID supports the arbitrary creation of second-level digital assets, as well as double-spend protection for all second-level assets. Some traditional blockchain implementations do not have native double-spend protection for overlay assets that do not form part of the underlying consensus and validation mechanism. Furthermore, other blockchain implementations cannot pay transaction fees second-level assets. This serves to remove friction for higher level distributed applications that need to transact in their own native asset with individuals.

IDENTITY

Creating a Decentralized Identity Blockchain Benefiting the Individual

The use of blockchains has spurred a vibrant new economy of decentralized applications, from virtual currencies to new social networks to decentralized virtual asset exchanges. When Satoshi wrote his seminal white-paper almost ten years ago about how to create a decentralized currency, it could never have been predicted that the underlying blockchain would have such a ubiquitous use-case in our modern society. Almost every human interaction involving some modicum of trust is a potential opportunity for someone with enough understanding of the problem presenting itself and how to decentralize the trust required for the interaction between parties. As blockchain technologies evolve, a clear differentiation between different levels of abstraction is taking place.

1. At the base is new implementation of decentralized consensus using novel approaches to reaching some minimal level of trust in the network state.
2. Next we see the implementation of smart contracts, which allow programmatic control of the transfer of assets and the modification of state based on a set of rules set out by the interacting parties.
3. An additional layer has emerged on which decentralized resources such as databases, file systems, computing power, and communications are being managed and paid for using smart contracts to settle payment and allocation of resources.
4. We then see an application layer, which can run on either centralized or decentralized infrastructure and allow organizations and companies to solve real world business problems. These applications address the complex needs of business with records and value stored and exchanged using blockchains as the trusted repository of truth.

Traditional Blockchain Scalability Constraints

In traditional blockchain implementations there are a number of factors that affect scalability and thus mass adoption.

Resource Waste

A prime waste of resource can be observed in the first blockchain implementation of Bitcoin, in which processing resources were leveraged to ensure network stability and consensus. This has led to an arms race in ASIC development to deliver the fastest hashing capability per dollar invested. Consuming huge amounts of electricity worldwide, and requiring large-scale mining farms situated in centralized positions near the power stations offering the best prices per Kilowatt Hour (KwH).

Transaction Rate Limitations

Current consensus mechanisms result in limited transaction processing capability by constraining the amount of data that can be included in each block. This has resulted in the Bitcoin blockchain being limited to a peak of about seven transactions per second utilizing the current block size.

Transaction Speed Limitations

Block time is another limiting factor. The speed at which a transaction can be verified is limited by the confidence in the immutability of the current state of the ledger. As a result, it is possible for a blockchain to experience short-term consensus breaks and reach full consensus only after a number of additional blocks have been confirmed. This has led to a requirement by participating parties engaging in virtual asset transactions to wait for multiple confirmations before assets are made available to the receiver.

Network Bandwidth as an Overarching Limitation to Adoption

Apart from the above mentioned limitations, the biggest limitation is internet connection bandwidth requirements for a traditional blockchain to operate at scale. If Bitcoin or Ethereum were to be able to scale to on-chain transaction capability of millions of transactions per second, the nodes would not be able to handle the bandwidth -- which would require half a terabyte of additional storage per day, and the corresponding bandwidth to maintain it.

SCALABILITY

Traditional Blockchain Scalability Requirements

Informed decisions must be made to address scalability constraints in a deliberate attempt to avoid the issues facing the mass adoption of blockchain technologies in everyday life. The following approaches have been identified as strong candidates for the next generation of blockchains and the associated distributed applications.

No Centralization of the Underlying Consensus Mechanism

By ensuring that only highly trusted individuals can run verifying nodes, the RelateID blockchain system ensures maximum decentralization of the consensus nodes.

Elimination of Wasted Computing and Energy Resources

The use of proof of trust, proof of realness, and proof of stake mitigates the need for massive processing power in a proof of work consensus paradigm.

Transaction Throughput that Scales with the Network

By employing pipelined block generation using pooled voting, blocks can be verified continuously in a real time directed acyclic graph of transactions which can reach consensus as fast as the network can propagate globally. Tests have shown that 96 nodes can deliver as many as 2 million transactions per second when housed in the same data centre. This was limited only by the Input / Output capabilities of the hardware used in these experiments. This points to a global network with latencies of 150ms being able to reach transaction speeds between 20000 and 200000 transactions per second, several orders of magnitude higher than other traditional blockchain solutions.

Minimum Global Confirmation Times Can be as Low as 1.3 Seconds

Unlike Bitcoin, which has a block time on average of 11 minutes, confirmation times on the RelateID blockchain are only limited by the latency between the blockchain's nodes, which can be as low as 1.3 seconds on a global network with latencies of over 150ms.

Minimizing Bandwidth and Storage Resource Requirements for Full Nodes

RelateID uses a distributed database that continuously updates every node with new transactions and the metadata associated with block consensus votes. By using a replication factor that will be increased in proportion to the number of nodes, the RelateID blockchain will be able to maintain minimal bandwidth and storage requirements relative to other blockchain implementations, especially when the network includes thousands of nodes.

Efficient Storage and Replication of the Blockchain

Because the RelateID blockchain will use a replication factor that will not require full duplication of all data on every node, RelateID will lower the storage requirements for the overall blockchain and its associated data. As the network grows, the probability of consensus failure declines. This allows us to increase the replication factor as the network grows, and thus the network becomes more efficient, as well as faster, and more fault tolerant as it grows.

RelateID

A Secure, Scalable ID Blockchain - for the World

RelateID deploys tried and tested open-source solutions. When combined these solutions deliver a scalable self-sovereign identity network for personal data that leverages speed, throughput, reliability, high-security, and manage-ability of personal data with blockchain immutability.

Performance

The RelateID blockchain has been experimentally verified and tested on standard hardware and network platforms with impressive results with respect to block time, and transaction and storage capacity.

1. Block Time

   Using the power of a new blockchain pipeline and the node voting mechanism delivered by the underlying consensus algorithms, which are based on derivatives of the PAXOS, consensus algorithm, RelateID is able to reach consensus at speeds limited only by the average latency between nodes. Given a global network and a node to node latency of 150ms, RelateID can reach full consensus in less than 1.5 seconds, compared with 11 minutes for Bitcoin (requiring 3 confirmations) and 20 seconds for Ethereum (requiring 12 confirmations). And unlike Bitcoin or Ethereum, no additional confirmations will be required after consensus has been reached, as no future uncertainty exists on the state of the RelateID blockchain.

2. Transaction Capacity

   Transaction throughput has been experimentally verified to result in linear scaling of transaction capacity that results in over 20,000 transactions per second per node in the network of 96 collocated nodes. Thus a network of 96 collocated nodes was be able to support over 2 million transactions per second.

3. Storage Capacity

Unlike other blockchains, RelateID allows for storage of arbitrary data related to blockchain addresses allowing individuals to store their own data in the blockchain itself, instead of an external referenced location. RelateID scales linearly with respect to the total amount of data residing in the underlying database on the blockchain. Depending on the replication factor, the blockchain can store and manage assets and data in the petabyte range with a relatively small number of nodes with large storage capacity.

Profile Security

RelateID uses private keys to manage access to personal data, and will use a distributed application to facilitate the management of a user's private keys. RelateID plans to implement a novel multi-signature key management system, which will use an individual's validation network to recover access to one's own profile in the event of loss of the primary private key by distributing the recovery process in a way that allows for partial key storage on validators data store. This protocol will allow a person to use their validation network to re-verify their identity, using stored biometrics, and other forms of verification to unlock the profile using a reconstructed private key. Third party ID verification companies will also be a key component in this process. Using this approach it will be possible to recover a lost private key in much the same way people currently recover lost passwords.

Public Naming System

The use of a public naming system will allow individuals or entities to publish a human readable name associated with their RelateID identities. In some cases, individuals and their associated entities will want to have their names publicly associated with a specific blockchain address, while retaining anonymity on other addresses. RelateID would offer the ability for addresses to be linked to a name translation service so that individuals can link their addresses to specific labels, and make these publicly available as destinations for the purpose of receiving funds, and or other digital assets.

Distributed Application Architecture

RelateID will be deployed as a blockchain alongside a distributed application (DAPP) which will extend the base interface of the system to the individual via a web wallet which will function as the interface for profile management, the validation network, DAPP subscription, and permission management console. The system will use a code base that will be managed through an open source project allowing it to be upgraded and improved over time by the community to include additional Application Program Interfaces (APIs) as well as functionality within the profile management console.

1. DAPP Nodes

Distributed applications will run off of RelateID nodes, which will also be distributed web services platforms, allowing for the serving and running of the primary DAPP supporting the RelateID blockchain. APIs will allow independent DAPPs running on their own systems to interact with the database to store and retrieve data with permission from individual's data stores. Permission allocation via DAPP subscription by the individual will ensure that each DAPP is isolated from the private data belonging to the individual, which may be related to other DAPPs he individual has subscribed to.

2. Distributed Cloud Architecture

RelateID will provide standardized Docker containers which will be able to run on any docker enabled platform, including a personal computer, Google cloud, Amazon AWS, and decentralized clouds, such as iex.ec. Essentially anyone will be able to run a node anywhere.

3. Docker Containers

A docker container is a virtualized operating environment which facilitates the standardization of all application packages intended to fulfill a specific software need. A container can take customized configurations initiated at startup, which in RelateID nodes will include all configuration variables required to run the node as a specific individual, and will require an individual's private key to initialize fully.

Glossary
of Terms

Blockchain
An immutable decentralized ledger with no third party trust required to validate authenticity of transactions involving the transfer of assets.

Bitcoin
The first practical implementation of a blockchain.

SQL Databases
Databases that store data in tabular format and use a query language that addresses entries based on table names and column headers.

NoSQL Databases
A database that stores data in a format consisting of key-value pairs or a graph.

Distributed Databases
A database that makes use of more than one hardware / software system on a network which can be geographically separated, to store and retrieve data that is replicated across the network, and may be split into subsets of the full dataset to achieve efficiency gains.

Replication factor
This refers to the degree to which data in a distributed database is split into smaller subsets of the full database in order to achieve efficiency gains in storage, bandwidth utilization, data delivery and processing capability.

Immutable
This refers to the property of a record when it can not be changed after it is created.

Zero Knowledge Proofs
A method by which one individual can exchange proof of knowledge or lack thereof, of a certain secret, without revealing the secret to the the asking party, and without an observer being able to determine if the holder of the secret is in fact in possession of the secret.

Node
A node is a reference to a single computer system which is a supporting system in a network of similar computer systems which work as a distributed solution for the delivery of a network service such as processing, storage, databases, network traffic routing. In general Nodes are agnostic and stateless as far as the network is concerned, and can be removed without affecting the service delivery of the system.

Consensus
Consensus describes a group dynamic in which decisions are decided by a majority within a group with the intention of resolving group disputes without the use of a third party mediator. Usually all parties in the group agree prior to reaching consensus that once consensus is reached that all parties in the group accept the outcome of the consensus decision even if they did not agree with the decision individually.

ASIC
An ASIC is a specialized electronic device whose circuits are designed to do one specific operation very efficiently, as opposed to a general purpose processor which is programmable and can solve many problems by being directed at run time by the software which is loaded at the time of execution.

Sybil Attack
In computer security is an attack wherein a reputation system is subverted by forging identities in peer-to-peer networks. It is named after the subject of the book Sybil, a case study of a woman diagnosed with dissociative identity disorder.

Byzantine Fault
Any fault presenting different symptoms to different observers.[3] A Byzantine failure is the loss of a system service due to a Byzantine fault in systems that require consensus.[4] In fault-tolerant computer systems, and in particular distributed computing systems, Byzantine fault tolerance (BFT) is the characteristic of a system that tolerates the class of failures known as the Byzantine Generals' Problem,[1] which is a generalized version of the Two Generals' Problem – for which here is an unsolvability proof. The phrases interactive consistency or source congruence have been used to refer to Byzantine fault tolerance, particularly among the members of some early implementation teams.[2] It is also referred to as error avalanche, Byzantine agreement problem, Byzantine generals problem and Byzantine failure

Zero-knowledge proof/Zero-knowledge protocol
In crytography, it is a method by which one party (the prover) can prove to another party (the verifier) that a given statement is true, without conveying any information apart from the fact that the statement is indeed true. Another way of understanding this would be: Interactive zero-knowledge proofs require interaction between the individual (or computer system) proving their knowledge and the individual validating the proof.[1]

Double-spending
This a potential flaw in a digital cash scheme in which the same single digital token can be spent more than once. This is possible because a digital token consists of a digital file that can be duplicated or falsified.[1] As with counterfeit money, such double-spending leads to inflation by creating a new amount of fraudulent currency that did not previously exist. This devalues the currency relative to other monetary units, and diminishes user trust as well as the circulation and retention of the currency. Fundamental cryptographic techniques to prevent double-spending while preserving anonymity in a transaction are blind signatures and particularly in offline systems, secret splitting.[1]

Hashing
The transformation of a string of characters into a usually shorter fixed-length value or key that represents the original string. Hashing is used to index and retrieve items in a database because it is faster to find the item using the shorter hashed key than to find it using the original value. It is also used in many encryption algorithms.