Editors: Peter Gutmann, pgut001@cs.auckland.ac.nz | David Naccache, david.naccache@ens.fr | Charles C. Palmer, ccpalmer@us.ibm.com

On the Origins and Variations of Blockchain Technologies

Alan T. Sherman, Farid Javani, Haibin Zhang, and Enis Golaszewski | University of Maryland, Baltimore County

We explore the origins of blockchain technologies to better understand the enduring needs they address. We identify the five key elements of a blockchain, show the embodiments of these elements, and examine how these elements come together to yield important properties in selected systems. To facilitate comparing the many variations of blockchains, we also describe the four crucial roles of common blockchain participants. Our historical exploration highlights the 1979 work of David Chaum, whose vault system embodies many of the elements of blockchains.

Understanding Blockchains

With myriad blockchain distributed ledger systems in existence, more than 550 associated patent applications under review, and much associated hype, it can be difficult to make sense of these systems, their properties, and how they compare. Through exploring the origins of these technologies, including David Chaum's 1979 vault system, we provide insights and a clear and useful way to think about blockchains. Our historical perspective distills important ideas, identifies enduring needs, and shows how changing technologies can satisfy those needs. This perspective will help people understand where blockchains came from, whether they are important, and if they will persist. (For a complete list of references, see A. Sherman et al.)[1]

Digital Object Identifier 10.1109/MSEC.2019.2893730
Date of publication: 20 March 2019

Elements of Blockchains

Blockchains provide a mechanism through which mutually distrustful remote parties (nodes) can reach consensus on the state of a ledger of information. To trace the origins of these technologies, we start by identifying their essential elements informally. A blockchain is a distributed ledger comprising blocks (records) of information, including information about transactions between two or more parties. The blocks are cryptographically *linked* to create an immutable ledger. Nodes may append information to the ledger through invoking transactions. An *access policy* determines who may read the information. A *control policy* determines who may participate in the evolution of the blockchain and how new blocks may potentially be appended to the blockchain. A *consensus policy* determines which state of the blockchain is valid, resolving disputes should conflicting possible continuations appear.

As explained by Cachin and Vukolic,[2] a range of control policies is possible, including permissioned, consortium, private, and permissionless blockchains. In a permissioned blockchain, a body identifies and controls who may update state and issue transactions. A private blockchain is a permissioned blockchain controlled by one organization. A consortium blockchain is a permissioned blockchain involving a group of organizations. In a permissionless blockchain, anyone may potentially append new blocks, with the consensus policy (e.g., a majority of participants) determining which continuation is valid.

Blockchains achieve consensus and control (and, in particular, prevent double spending) in part through applying protocols and establishing high costs (both economic and computational) to modify the ledger. Typically, permissioned systems run faster than permissionless systems do because their control and consensus strategies depend on faster fault-tolerant protocols[3] rather than on time-consuming cryptographic proofs of work (PoWs), and they usually involve fewer nodes. Gencer et al. show that permissionless blockchains (such as Bitcoin and Ethereum) are much more centralized than many people assume: 20 mining pools control 90% of the computing power.

Some blockchains additionally support the idea of smart contracts, which execute terms of agreements between parties, possibly without human intervention. These agreements might be embodied as arbitrary computer programs including conditional statements.

 Copublished by the IEEE Computer and Reliability Societies

Embodiments of the Elements

Although the seminal paper on Bitcoin appeared in 2008 (with the mysterious author Satoshi Nakamoto),[4] most of the underlying technological ideas had arisen many years earlier. A blockchain is a type of distributed database, an idea that goes back to at least the 1970s (e.g., Wong[11]). More generally, the idea of record keeping goes back millennia, including to ancient Mesopotamia. Kanare describes proper methods for scientific logging, including the idea of preserving all transaction records, in addition to the history of any modifications to the collected data—ideas that are found in many systems (e.g., Hyperledger Fabric).

The idea of immutably chaining blocks of information with a cryptographic hash function appears in the 1979 dissertation of Ralph Merkle at Stanford, in which Merkle explains how information can be linked in a tree structure now known as a *Merkle hash tree*. A linear chain is a special case of a tree, and a tree provides a more efficient way of chaining information than does a linear chain. Subsequently, in 1990, Haber and Stornetta applied these ideas to time-stamp documents, creating the company Surety in 1994. These prior works, however, do not include other elements and techniques of blockchain.

To prevent an adversary from unduly influencing the consensus process, many permissionless systems require that new blocks include a proof of computational work. Nakamoto's paper cites Back's[5] 2002 effective construction from Hashcash. In 1992, Dwork and Naor proposed proof of computation to combat junk mail. The idea and a construction underlying PoW, however, may be seen in an initial form in 1974 in Merkle's puzzles,[6] which Merkle proposed to implement public-key cryptography. Bitcoin was the first to use PoW for both mining and achieving consensus.

PoW aims, in part, to defend against Sybil attacks, in which adversaries attempt to forge multiple identities and use those forged identities to influence the consensus process. With PoW, however, a node's influence on the consensus process is proportional to its computational power: forging multiple identities that share the adversary's given computational power does not help. To adapt to varying amounts of available computational resources, PoW systems dynamically throttle the difficulty of the PoW problem to achieve a certain target rate at which the problems are solved.

Permissioned blockchains can be modeled using the concept of (Byzantine fault-tolerant) state machine replication, a notion proposed in 1978 by Lamport and, later, concisely formalized by Schneider. State machine replication specifies what are the transactions and in what order they are processed, even in the presence of (Byzantine) faults and unreliable communications.[3] Thereby, to achieve a strong form of transaction consensus, many permissioned systems build on the ideas from the 1998 Paxos protocol of Lamport[7] (which deals only with crash failures) and from the 2002 Practical Byzantine Fault Tolerance protocol of Castro and Liskov. Nakamoto observed that the permissionless Bitcoin system realizes Byzantine agreement in open networks.

Arguably, many of the elements of blockchains are embodied in David Chaum's 1979 vault system,[8] described in his 1982 dissertation[9] at Berkeley, including detailed specifications. Chaum describes the design of a distributed computer system that can be established, maintained, and trusted by mutually suspicious groups. It is a public record-keeping system with group membership consistency and private transaction computations that protects individual privacy through physical security. The building blocks of this system include physically secure vaults, existing cryptographic primitives (symmetric and asymmetric encryption, cryptographic hash functions, and digital signatures), and a new primitive introduced by Chaum—threshold secret sharing.[8] Chaum's 1982 work went largely unnoticed, apparently because he never made any effort to publish it in a conference or journal, instead pursuing different approaches to achieving individual privacy.

In Chaum's system, each vault signs, records, and broadcasts each transaction it processes. Chaum states, "Because the aggregate includes COMPRESSED_HISTORY, the [cryptographic] checksum is actually 'chained' through the entire history of consensus states."[9] He further says, "Nodes remember and will provide all messages they have output—each vault saves all it has signed, up to some limit, and will supply any saved thing on request; only dead vaults can cause loss of recently signed things."[9]

Chaum's system embodies a mechanism for achieving membership consistency: "Among other things, the algorithms must provide a kind of synchronization and agreement among nodes about allowing new nodes into the network, removing nodes from the network, and the status of nodes once in the network."[9] The system also embodies a weak form of transaction consensus, albeit vaguely described and apparently not supporting concurrent client requests: "If the output of one particular processor module is used as the output for the entire vault, the other processors must be able to compare their output to its output, and have time to stop the output on its way through the isolation devices."[9] The consensus algorithm involves majority vote of nodes based on observed

signed messages entering and leaving vaults.

Chaum created his vaults system before the emergence of the terms *permissioned* and *permissionless* blockchains, and his system does not neatly fall into either of these discrete categories. In Chaum's system, each node identifies itself uniquely by posting a public key, authenticated by level 2 trustees. For this reason, some people may consider Chaum's system a permissioned blockchain.

This narrow view, however, diminishes the fact that each node can be authorized in a public ceremony independently from any trustee. During this ceremony, vaults are assembled from bins of parts, which the public (not necessarily nodes) can inspect and test—a procedure that inspired Chaum to coin the more limited phrase *cut and choose*. Regardless of whether one views some configurations of Chaum's vaults as permissionless systems, the trust bestowed through the public ceremony creates a system whose trust model is the antithesis of that of a private (permissioned) blockchain. For these reasons, we consider Chaum's system publicly permissioned.

Chaum assumes, essentially, a best-effort broadcast model, and he does not provide mechanisms for achieving consensus with unreliable communications—technologies that subsequently have been developed and applied in modern permissioned systems. Chaum's dissertation does not include the ideas of PoW, dynamic throttling of work difficulty, and explicit smart contracts (though Chaum's vaults support arbitrary distributed private computation).

Unlike in most blockchain systems, nodes in Chaum's system hold secret values, which necessitates a more complex mechanism for restarting after failures. Using what Chaum calls *partial keys*, any vault can back up its state securely by encrypting it with a key and then escrowing this key using what we now call *threshold secret sharing*. After reading Chaum's February 1979 technical report[8] that describes partial keys, Adi Shamir published an elegant alternate method for secret sharing in November 1979.

Chaum also notes that pseudonyms can play an important role in effecting anonymity: "Another use allows an individual to correspond with a record keeping organization under a unique pseudonym which appears in a roster of acceptable clients."[9] To enable private transactions for blockchains, engineers are exploring the application of trusted execution environments, continuing an approach fundamental in Chaum's vaults.

In 1994, Szabo[10] coined the term *smart contract*, but the idea of systematically applying rules to execute the terms of an agreement has a long history in trading systems. For example, in 1949, with a system involving ticker tapes and humans applying rules, Future, Inc. generated buy and sell orders for commodities. Recently, so-called hybrid blockchains have emerged, which combine Byzantine fault-tolerant state machine replication with defenses against Sybil attacks—for example, PeerCensus, ByzCoin, Solidus, Hybrid Consensus, Elastico, OmniLedger, and RapidChain.

Also, Hyperledger (an umbrella project involving Fabric, a system for permissioned blockchains) and Ethereum (a platform for public blockchains) have joined forces. Recently, researchers have applied game theory to model and analyze the behaviors of players and mining pools in blockchain-based digital currencies (see Dhamal and Lewenberg). Table 1 chronicles some of the important cryptographic discoveries underlying blockchain technologies. For example, in 2018, the European Patent Office issued the first patent on blockchain—a method for enforcing smart contracts.

Table 1. A timeline of selected discoveries in cryptography and blockchain technology.

1970	James Ellis, public-key cryptography discovered at Government Communications Headquarters (GCHQ) in secret
1973	Clifford Cocks, RSA cryptosystem discovered at GCHQ in secret
1974	Ralph Merkle, cryptographic puzzles (paper published in 1978)
1976	Diffie and Hellman, public-key cryptography discovered at Stanford
1977	Rivest, Shamir, and Adleman, RSA cryptosystem invented at the Massachusetts Institute of Technology
1979	David Chaum, vaults and secret sharing (dissertation in 1982)
1982	Lamport, Shostak, and Pease, Byzantine Generals Problem
1992	Dwork and Naor, combating junk mail
2002	Adam Bach, Hashcash
2008	Satoshi Nakamoto, Bitcoin
2017	Wright and Savanah, nChain European patent application (issued in 2018)

Comparison of Selected Blockchain Systems

To illustrate how the elements come together in actual blockchain systems, we compare a few selected systems, including Chaum's vaults, Bitcoin, Dash, Corda, and Hyperledger Fabric, chosen for diversity. Table 2 describes how each of these systems carries out the four crucial

participant roles of any blockchain defined ahead. For more context, Table 3 characterizes a few important properties of these systems and of one additional system—Ethereum.

In his vault system, Chaum[9] identifies four crucial participant roles of any blockchain, which we call *watchers*, *doers*, *executives*, and *czars*. The watchers passively observe and check the state of the ledger. The doers (level 1 trustees) carry out actions, including serving state. The executives (level 2 trustees) sign (or otherwise attest to) the blocks. The czars (level 3 trustees) change the executives and their policies. Chaum refers to these participants as *bodies*,[9] leaving it unclear whether they could be algorithms.

Although most systems do not explicitly specify these roles, all systems embody them, though with varying nuances. For example, many people naively think of Bitcoin as a fully distributed system free of any centralized control, but, in fact, Bitcoin's core developers—as is true for all distributed systems—carry out the role of czars, changing the underlying software that implements policy. Despite these significant powers, the control structure is still more distributed (anyone can potentially become a core developer) than for a permissioned system controlled entirely by a prespecified entity. In Bitcoin, in each round, the winning miner (a doer) becomes an executive for that round. It is instructive to understand how each blockchain system allocates the four participant roles.

Table 3 illustrates some of the possible variations of blockchains, including varying control and consensus policies as well as different types of smart contracts. Whereas most blockchain systems maintain a single chain, Corda supports multiple independent chains, per node or among subsets of nodes. Similarly, Chaum's system also supports multiple chains. While most blockchains require each node to maintain the same state, Corda's and Chaum's systems do not.

Conflicts and Challenges

Because blockchain technologies address enduring needs for permanent, indelible, and trusted ledgers, they will likely be around in various forms for a long time. There are, however, some troubling fundamental conflicts that have not been solved. These conflicts include tensions between the following pairs of potentially dissonant concerns: privacy and indelibility, anonymity and accountability, stability and alternative future continuations, and current engineering choices and long-term security. For example, recent European privacy laws grant individuals the right to demand that their personal data be erased from most repositories (the right to be forgotten). Satisfying this erasure requirement is highly problematic for indelible blockchains, especially for ones whose nodes lack physical security.

An attraction of blockchains is their promise of stability enforced through consensus, yet sometimes the nodes cannot agree, resulting in a fork and associated possible splits in the continuations of the chain. In a hard fork, level 3 trustees issue a significant change in the rules that is incompatible with the old rules. In a

Table 2. Alignment of participant roles across five blockchain systems.

Role	**Chaum, 1982** *A flexible system based on vaults*	**Bitcoin, 2008** *A permissionless system using PoW*	**Dash, 2014** *A system that speeds up Bitcoin with a masternode network*	**Corda, 2016** *A permissioned system with smart contracts*	**Hyperledger Fabric, 2016** *A permissioned system with smart contracts*
Watchers Passively check state	Any computer online[9]	Nodes (distinct from full nodes)	Any computer online	Nodes	Peers
Doers Carry out actions, including serving state	Level 1 trustee	Full nodes	Miners	Nodes	Peers
Executives Sign blocks (or otherwise attest to them)	Level 2 trustee (promoted from level 1 by czars)[9]	Winning miner (promoted from doers each round)	Winning masternode (promoted by an algorithm from the masternode network, which anyone may join for 1,000 Dash)	Nodes (each node is an executive for its Corda blocks, called *states*)	Endorsing peers
Czars Change executives and their policies	Level 3 trustee[9]	Core developers	Quorum of masternodes	Permissioning service	Endorsement policies

Table 3. Three properties of several distributed ledger systems.

System	Permissioned?	Basis of Consensus	Smart Contracts
Chaum, 1982	Permissioned, with option for publicly permissioned	Weak consensus; does not handle concurrent client requests	Private arbitrary distributed computation
Bitcoin, 2008	Permissionless	PoW	Conditional payment and limited smart contracts through scripts
Dash, 2014	Combination	Proof of stake	No
Ethereum, 2014	Permissionless	PoW	Yes, nonprivate Turing complete objects
Hyperledger Fabric, 2015	Permissioned	Based on state machine replication	Yes, off-chain
Corda, 2016	Permissioned	Based on state machine replication	Yes (set of functions), including explicit links to human language

soft fork, there is a less severe change in the rules for which the old system recognizes valid blocks created by the new system (but not necessarily vice versa).

Security engineers must commit to particular security parameters, hash functions, and digital signatures methods.

No such choice can remain computationally secure forever in the face of evolving computer technology, including quantum computers and other technologies not yet invented. The hopeful permanence of blockchains is dissonant with the limited-time security of today's engineering choices.

Additional challenges facing blockchains include the huge amounts of energy spent on blockchain computations (especially PoW), the high rates at which ledgers grow, and the associated increases in transaction latency and processing time (Bitcoin's ledger is currently more than 184 GB).

As of September 2018, the hash rate for Bitcoin exceeded 50 million TH/s, consuming more than 73 TWh of power per day, more than the amount consumed by Switzerland. These hashes were attempts to solve cryptographic puzzles of no intrinsic value (finding an input that, when hashed, produces a certain number of leading zeroes), and almost all of these computations went unused. Attempts, such as Primecoin and others, to replace cryptographic hash puzzles with useful work (e.g., finding certain types of prime integers) are challenging because it is very hard to find useful problems that have assured difficulty and whose level of difficulty can be dynamically throttled. Some researchers are exploring alternatives to PoW, such as proof of space, proof of stake, and proof of elapsed time.

To understand blockchain systems, it is helpful to view them in terms of how the watchers, doers, executives, and czars carry out their functions under the guidance of the access, control, and consensus policies. This systematic abstract view helps focus attention on crucial elements and facilitates a balanced comparison of systems. Blockchains address many longstanding inherent needs for indelible ledgers, from financial transactions to property records and supply chains. With powerful existing cryptographic techniques, a wide set of available variations, and a large amount of resources allocated to these technologies, blockchains hold significant potential. ■

Acknowledgments

We thank Dan Lee, Linda Oliva, and Konstantinos Patsourakos for their helpful comments. Alan T. Sherman was supported in part by the National Science Foundation under Scholarship for Service grant 1241576.

References

1. A. Sherman, F. Javani, H. Zhang, and E. Golaszewski, On the origins and variations of blockchain technologies. 2018. [Online]. Available: http://arxiv.org/abs/1810.06130
2. C. Cachin and M Vukolic, "Blockchain consensus protocols in the wild," in *Proc. 31st Int. Symp. Distributed Computing*, 2017, vol. 1, pp. 1–16.
3. L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Trans. Programming Languages Syst.*, vol. 4, no. 3, pp. 382–401, 1982. [Online]. Available: https://dl.acm.org/citation.cfm?doid=357172.357176
4. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Bitcoin, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf
5. A. Back, "Hashcash: A denial of service counter-measure," Hashcash, 2002. [Online]. Available: http://www.hashcash.org/papers/hashcash.pdf

6. R. C. Merkle, "Secure communications over insecure channels," *Commun. ACM*, vol. 21, no. 4, pp. 294–299, 1978. [Online]. Available: https://dl.acm.org/citation.cfm?doid=359460.359473
7. L. Lamport, "The part-time parliament," *ACM Trans. Comput. Syst.*, vol. 16, no. 2, pp. 133–169, 1998. [Online]. Available: https://dl.acm.org/citation.cfm?doid=279227.279229
8. D. L. Chaum, "Computer systems established, maintained, and trusted by mutually suspicious groups," Elect. Eng. Res. Lab., Univ. California, Berkeley, Tech. Memo. UCB/ERL/M79/10, 1979.
9. D. L. Chaum, "Computer systems established, maintained and trusted by mutually suspicious groups," Ph.D. dissertation, Dept. Comput. Sci., Univ. California, Berkeley, 1982.
10. N. Szabo, "Smart contracts," 1994. [Online]. Available: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html
11. E. Wong, "Retrieving dispersed data from SDD-1: A system for distributed databases," in *Proc. 2nd Berkeley Workshop Distributed Data Management and Comput. Networks*, May 1977, pp. 217–235.

Alan T. Sherman is a professor of computer science at the University of Maryland, Baltimore County. His research interests include secure voting, applied cryptography, and cybersecurity education. He is a Senior Member of the IEEE. Contact him at sherman@umbc.edu.

Farid Javani is a Ph.D. student at the University of Maryland, Baltimore County. Contact him at javani1@umbc.edu.

Haibin Zhang is an assistant professor in the Department of Computer Science and Electrical Engineering at the University of Maryland, Baltimore County. Haibin received a Ph.D. from the University of California, Davis, in 2001. His research interests include distributed computing and secure blockchains. Contact him at hbzhang@umbc.edu.

Enis Golaszewski is a Ph.D. student at the University of Maryland, Baltimore County. Contact him at golaszewski@umbc.edu.

Digital Object Identifier 10.1109/MSEC.2019.2900896