JOURNAL OF PEER PRODUCTION
Journal of Peer Production
ISSN: 2213-5316
http://peerproduction.net

The Politics of Cryptography: Bitcoin and The Ordering Machines

Quinn DuPont

Abstract:

This paper explores the cryptographic aspects of Bitcoin. I suggest that cryptography can be reimagined and reconceptualised, putting forth an alternative to the dominant view that cryptography is secrecy. I argue that we can fruitfully view cryptography as a discrete notational system. I describe the specific cryptographic mechanisms as used in Bitcoin, and building on this foundation I offer a description of a full Bitcoin transaction. My method for understanding this technical foundation was to engage in praxis, so I describe the lessons I learned by running a Bitcoin mining machine. In conclusion, by drawing on my reconceptualization of cryptography as a discrete notational system, I suggest that Bitcoin functions as a new weapon in our control society.

Keywords: bitcoin, cryptography, control, code, order

Quinn DuPont

It was April 10th, 2013 and the price of a single Bitcoin surged past 250 USD on the Mt. Gox exchange.i A few months prior I had purchased seven Bitcoins for just under $200, and now they were nearing $2000 in value (Figure 1).ii But, just as fast as the market went up, it came down. Ever the amateur gambler, I panicked and sold too early. But, I was lulled by my humming money machine, permuting cryptographic codes by the millions every second. I was not the only one interested in playing the Bitcoin market, and the increasing price of Bitcoin was due to a number of factors, including a sustained distributed denial-of-service attack on Mt. Gox, and other people like me gambling in the latest crypto-anarchist adventure.

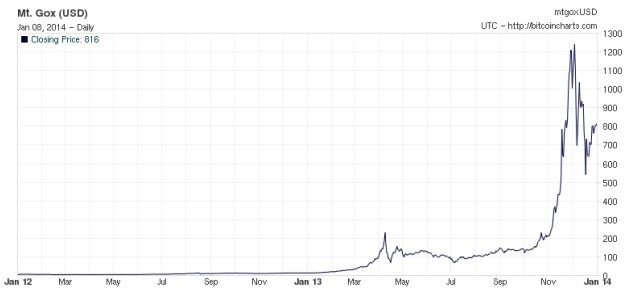


Figure 1. Daily closing price of Bitcoin on Mt. Gox USD market, January 2012 to January 2014. (source: http://bitcoincharts.com/charts/mtgoxUSD#rg730zczsg2012-01-01zeg2013-12-31ztgCzm1g10zm2g25)

Like many other modern currencies, Bitcoin is fiat money. But, unlike traditional fiat money, Bitcoin is cryptographic and electronic. There is no "physical" substrate to Bitcoin; the "coins" exist only as cryptographic representations stored in digital wallets. In the simplest caricature of complex economics, fiat money has no intrinsic value and thus requires people to trust that it will retain value. Usually government backing provides this semblance of trust, but when this trust is eroded (e.g., a weak government), value often plummets. Technical flaws also cripple the trust that sustains fiat money, such as rampant fraud, counterfeiting, or hyperinflation.

Trust in Bitcoin rests on a range of technical advantages supplied by cryptography. According to advocates, cryptography is secure (safe from technical or mathematical error).iii When applied to economic apparatuses, counterfeiting and double-spending is prevented through the use of public key cryptography, and hyperinflation is kept in check because the cryptographically-secure mining protocol ensures the measured production of money (with a maximum number of coins produced). Yet, as with general discussions of cryptography, complicated political issues often transform into a binary of state versus personal power. On the one hand, when cryptography is used for Privacy Enhancing Technologies it is seen as a block against government snooping. On the other hand, these same cryptographic technologies are often used against the people. This debate is particularly important for crypto/cyber-libertarians,iv who often believe that Bitcoin's lack of government backing is a virtue (J.M.P., 2013).

Described generally, cryptography is typically understood as a means to ensure "information security" or "information secrecy" (see e.g. Kohno et al., 2010). Here, security and secrecy are understood in terms of social relations (c.f. Bellman, 1979). Modelled in its simplest formulation, a secret is some information that I possess and you do not, while information security might be described more abstractly as control of information within a relationship. Or, the encyclopaedic definition: "the aim [of cryptography] is secrecy and confidentiality: the practice of keeping secrets, maintaining privacy, or concealing valuables" (Bauer, 2005). Though the conceptual analysis usually stops here, cryptography also functions more deeply, in ways rarely appreciated by those developing the technology. Understanding how cryptography functions at this deeper level is essential to understanding how Bitcoin functions. I argue that cryptography is central to Bitcoin and yet produces a set of *non-secret* powers for its social (and economic) effect.

This paper suggests that cryptography can be reimagined and reconceptualised, putting forth an alternative to the dominant view that cryptography is secrecy. The long history of cryptography is abbreviated to show that cryptography previously functioned in many different ways, but has been systematically black-boxed. By opening up this black box and reconceptualizing cryptography I argue that we can attend to the history of cryptography yet retain analytical rigour by viewing cryptography as a discrete notational system. Then, returning to Bitcoin, I describe the specific cryptographic mechanisms as used in Bitcoin. Building on this foundation I offer a description of a full Bitcoin transaction. My method for understanding this technical foundation was to engage in praxis, and so, returning to my introductory story about my own experiences with Bitcoin I describe the lessons I learned by running a Bitcoin mining machine. Finally, I take up Gilles Deleuze's suggestion to "look for new weapons" in our control society. By drawing on my reconceptualization of cryptography as a discrete notational system I suggest that Bitcoin

functions as a new weapon in a logic of order.

Conceptualizing cryptography

Given the important role that cryptography plays in our lives, it is surprising how little attention it receives outside of the academic worlds of mathematics/engineering and privacy/law. A few authors outside of these fields have explored how cryptography intersects with other domains, but for the most part, cryptography has been a mathematical and legal concern.v Perhaps the most useful of these authors is Kahn (1967) who provides a history of cryptography,vi identifying how the golden age of Western cryptography started with Leon Battista Alberti (1404-1472). Reaching far beyond mathematics and law, Alberti identified cryptography as an intriguing mix of technologies and practices: the "occult arts," "mysteries of nature," "strange characters with unusual meanings," "movable type," and secrecy (Alberti, 2010).

Over the next six centuries these themes played out in interesting ways. In the 17th century, John Wilkins (1614-1672) developed a proposal for an unambiguous, universal/philosophical language (1668) after developing the ideas in *Mercury* (1694), the first English-language cryptography manual.vii Athanasius Kircher (c. 1601-1680), similarly, reworked Johannes Trithemius' (1462-1516) troubled work on cryptography for his universal language scheme (Figure 2).viii Francis Bacon (1561-1626) worked out a method for universally signifying nature with his bilateral cipher before launching his "great reformation" of scientific interpretation (Bacon, 1762). And Gottfried Wilhelm Leibniz (1646-1716) contributed to discrete mathematics after writing his dissertation (1989) on the cryptographic combinatorics of Raymund Lull's (1232-1315) occult and cabalistic science (Gardner, 1958).

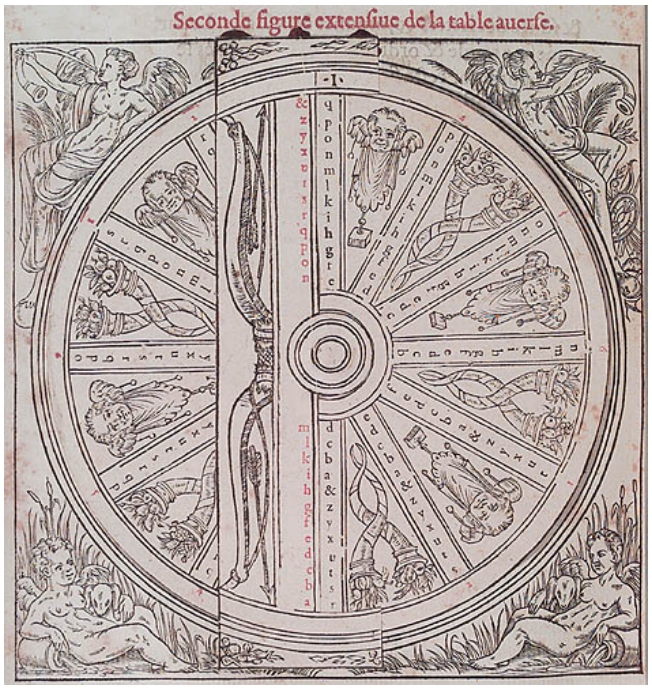


Figure 2. Tritheme's (Trithemius) cipher wheel (1561)

The richness of these historical themes has now, in contemporary cryptography, practically disappeared. Cryptography has become operationalized and nearly univocal—gone are the vibrant connections to language, science, and art. Today, our most influential conceptualization is Claude Shannon's (1945) linked notion of secrecy and information. Prior to his famous Mathematical Theory of Communication (Shannon and Weaver, 1948), Shannon had been working on war-time encryption systems.x Shannon's cryptographic work stemmed from a long line of influences and prior work (e.g., Nyquist, Hartley, and Wiener) (Thomsen, 2009), as well a richer, multivocal, and open conceptual backdrop (Cherry, 1953; Geoghegan, 2008).

Research on cryptography set the stage for Shannon's more general, and more rigorous portrayal of information. It was in working out the coding issues for cryptography that Shannon developed his theory of information.xi In contrast to much of the engineering work being done on information transmission at that time, Shannon focused on discrete rather than continuous signals (Thomsen, 2009). Drawing on Hartley, Shannon bracketed the issue of meaning, and discussed only *how much* information can pass through a channel. This conceptualization, combined with Nyquist's observation that information transmission obeys a logarithmic rule, allowed Shannon to generalize the issue and show that information accords to physical properties of the world (Aspray, 1985; Hayles, 1999).

Despite being central to his study, Shannon left his understanding of "secrecy" implicit. The closest we get to a description of secrecy is that it arises from the "*a priori* probability associated with… choosing that [enciphering] key" (Shannon, 1945). The *a priori* probability is a function of the statistics of the transformation from one (information) space to another. In ideal situations, secrecy becomes a matter of making guesses in the presence of a stochastic phenomenon.

Before Shannon "won" the battle for our history—how we come to think of cryptography—the field of possibility was more open (Cherry, 1953). Competing conceptualizations existed but none were better prepared than Shannon's for the coming changes in cybernetics and informatics (also made possible, in large part, by his work on the Mathematical Theory of Communication). By identifying "secret information" as the endpoint of cryptographic conceptualization, however, we risk teleological explanations that make it difficult to understand cryptography's non-secret role in contemporary society. If we start to reconfigure our understanding of cryptography—open the socio-technical black box—we stand to gain a deeper appreciation, and may be better able to understand the politics of technologies that use cryptography.

In order to open the black box of cryptography our metaphors must also be rethought. The analysis of cryptography rarely penetrates beyond metaphors, so it is especially important to dispel any mistaken notions. There are two closely related classes of metaphor most often used to describe cryptography: the dead, and the missing or hidden. The metaphor of death is most common in literary accounts, if for no other reason than the association of "crypt" and "*crypto*graphy." On this account cryptography "buries" (Derrida, 1998; Turing quoted in Mackenzie, 1996) and even communes with the dead (Poe, 1991; Rosenheim, 1997). The metaphor of being "hidden" is likely due to the perceived semiotic shift that occurs when a text gets encrypted: at one moment it is there and seen by all, and (like a good magic trick) the next moment it is gone (until conjured up again in decryption). Here the language used is "hidden" (Schmeh, 2012), "veiled" (Cooke, 1983), or perhaps text containing a "false bottom" (Glidden, 1987). These metaphors are evocative but lacking. As will become clear below, a more useful metaphor is "discreet," in the sense of being circumspect, but also "discrete" as separate or distinct.xii

Re-conceptualizing cryptography

I argue that cryptography can be re-conceptualized as a generic notational system, quite appropriately sharing the ambiguous name "code." To get to this conclusion, I argue that cryptography is unspeakable in the sense of being a written language without syllables (which perhaps disqualifies it as a language at all). Due to the curious nature of being unspeakable, cryptography shifts from mimetic to algorithmic representation.

Algorithmic representation permits the transposition and combination of its symbolic elements, but only when made of disjoint, articulate, and unambiguous marks. This system of marks is a discrete notational scheme. Additionally, because algorithmic representation permits the rearrangement of its symbolic structure it can be used to order the world, in subtle but powerful ways.

"Literature has nothing more to say," remarks Friedrich Kittler, "because it all ends in cryptograms" (1999). This is only true if we take Kittler very literally, that literature has been silenced and can no longer *speak*. Kittler reminds us that speaking written words is not natural and immediate. First, the Greeks needed to "invent" vowels so as to create a storage system able to capture the wealth of articulable knowledge (Winthrop-Young, 2011). Second, in what Kittler (1990) called the Discourse Network 1800, the Mother's Mouth must teach children how to speak these written marks.

Kittler goes on to accuse those languages lacking written vowels as being unable to perfectly capture the world's information. Written languages that lack vowels are not as expressive, Kittler problematically suggests, and so are not as mimetic (Winthrop-Young, 2011). Kittler remarks, "nobody knows how the heretic king Akhneten called his N-f-r-t-t when they were making children" (Kittler quoted in Winthrop-Young, 2011, p. 91). But, of course, this is a challenge for history, not the Egyptians. The Egyptians were able to speak the name that was recorded as N-f-r-t-t because they added the appropriate vowels, that is, they created syllables in speech.

It is hard to underline this important point in the written form—I suggest you try speaking the letters, but out loud: N-f-r-t-t. The punchline is that in order to be voiced, syllables must be created, making sounds like "nef eff arr tee tee." In the audio book version of E.A. Poe's *The Gold Bug* (2013) this same comedy plays out when the poor narrator is forced to articulate a long series of cryptographic symbols, so that 5 3 ‡ ‡ † 3 0 in the text becomes "five, three, double dagger, double dagger, single dagger, three, zero" in the audio book (and to great comedic effect, this goes on for half a page). While the narrator demonstrated that it is possible to speak a cryptogram in some fashion, it no longer counts as speech in the sense of meaningful language. This is the lesson that the Mother's Mouth must teach her children, separating the grunts of animals from those of human language (Kittler, 1990).

The reason cryptography cannot be spoken is because it has very particular syntactic requirements. These requirements where formalized by Nelson Goodman (1976) in his analysis of notational schemes. Roughly, a notational scheme requires that marks can be interchanged within a class of marks without difference, and that one can in theory determine what character a particular mark belongs to. The result of this analysis is that a number of forms of writing, upon inspection, share a common ancestor. In addition to cryptography, notational schemes include musical notation, Morse code, and binary (e.g., compiled software code).

Although Kittler's accusations that consonantal written languages are less mimetic than their vowelized cousins is problematic, this lack becomes very real when writing transforms into a notational scheme. The smooth contours of the world are not well represented in a notational scheme. In Kittler's media triptych, the gramophone and film camera are highly mimetic in that they record every gradient of nature, whereas the typewriter eradicates the "continuous movement of the hand" in handwriting (Benjamin quoted in Kittler, 1999, p. 196). The representation of likeness or verisimilitude is easy to imagine with a photograph, and yet even a typed, well-crafted story can also be very mimetic, as though the reader is transported into the scene.

So the typewriter can, of course, be used to write a mimetic story, but it excels when it reduces the world to discrete symbols. Putting aside, for the moment, Kittler's (1999) penchant for inaccurate but provocative history, it is a short historical step from typewriter to cryptogram, with the transformation of Nietzsche's Malling typewriter into the German crypto-typewriter, the Enigma machine (Figure 3). Unlike the gramophone and film camera, recording N-f-r-t-t is no trouble for the typewriter. While the typewriter cannot capture voiced syllabification, it perfectly represents the alphabetization of language. And for the typewriter so too for cryptography: when encrypted, N-f-r-t-t readily becomes A-S-E-G-G.

Figure 3. Four rotor German naval Enigma machine on display at Bletchley Park (Manske, 2005)

While notational schemes are poor at mimetic representation, the ease in which parts can be split and rearranged make them ideal for what I call "algorithmic" representation. Algorithmic technologies are powerful because they chop the world up in to discrete pieces and then re-arrange the results.xiii These technologies *order* the world rather than model or recreate it. To underline the point, all cryptography is just the simple substitution cipher, shuffling symbols about in determined, ordered ways. And the same principle is at work in all notational systems, where complex and surprising uses abound. Conceptualized in light of a notational scheme, cryptography surpasses the typical narrow conceptualization as secrecy.

For example, Kittler's barbarian Spartans invented a writing technology called the *skytale*. The *skytale* is a leather strap along which letters are written that, once wrapped around a wooden rod of a particular diameter, reveals a message. The device was once thought to facilitate a kind of primitive encryption, but recent research has shown it to be something else (Kelly, 1998; West, 1988).xiv How this technology was actually used is unknown, and somewhat beside the point: the *skytale* exemplified an early algorithmic machine, made of Kittler's alphabet but impossible to phonetically speak. In fact, there is good evidence that the device may have been used to ensure silence, as a way of guarding against the accidental articulation of bad omens and the like (c.f. Lateiner, 2005; Montiglio, 2000). The *skytale* is, perhaps, the first device to destroy syllabification, and to depart from mimetic media and move towards ordering language and the world.

Another important aspect of cryptography is that it decomposes language into identical, modular pieces. This fact works equally well for other applications of the discrete notational scheme too. A thousand years after the Spartans, Alberti ushered in a new notational method for architecture that replaced the older craft-oriented way of building (Carpo, 2011). This notational system permitted architects to become designers, not makers,

and enabled the construction of identical but modular buildings. So close is the connection between the notational schemes of architecture and cryptography, in fact, that Alberti generalized his architectural advancements and applied the same thinking to language to develop polyalphabetic cryptography (Alberti, 2010; Kahn, 1980).

Similarly, Leibniz employed a notational scheme when he established the groundwork for discrete mathematics, and specifically, the mode of mathematics that we now use to analyze and model cryptography, known as combinatorics (what Leibniz called "complexions"). In Leibniz's early work, *Dissertation on the Art of Combinations* (*Dissertatio de arte combinatoria*) (1989), he develops Raymund Lull's cabalistic method of interrogating language and the natural world, setting it on firm mathematical and logical ground (Figure 4).xv To do so, Leibniz derives his metaphysical analysis of order from the existence of natural parts and wholes (or "unities"). Leibniz writes, "let the whole be ABC; then AB, BC, and AC will be smaller wholes," which "by reason of order" results in "abcd, bcda, cbab, dabc" (calculated to be "arranged in 24 ways") (Leibniz, 1989, p. 78). Leibniz's development of order ultimately leads to modern symbolic logic, and by way of Stanhope, Jevons, and others to the calculating machine (Gardner, 1958).

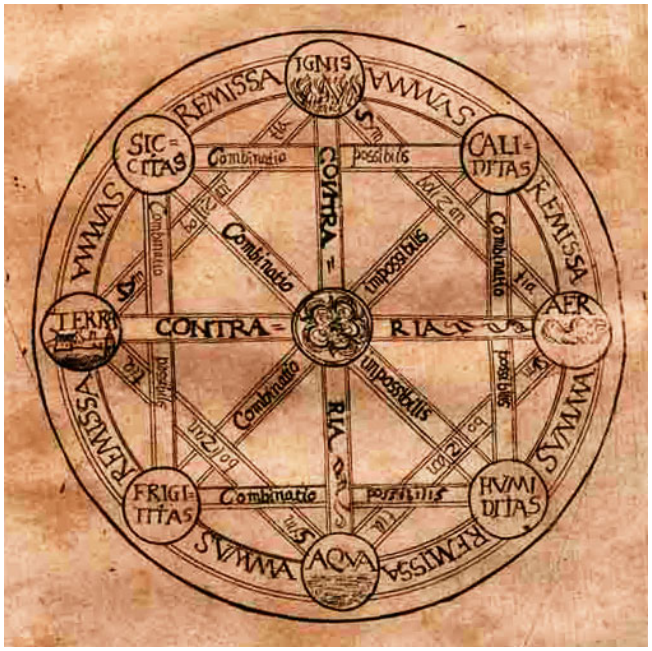
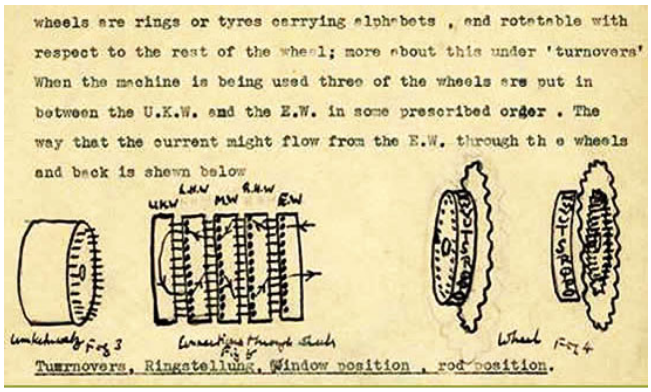


Figure 4. Leibniz's volvelle from his Dissertatio, 7v (1666)

Eventually, it can be surmised, Turing's machine would read a tape of Leibniz's letters and re-order the input, perhaps (after von Neumann) according to a particular programmed algorithm. Although Shannon recognized the importance of discrete symbol processing (as noted above), it was Turing that really advanced this position by seeing the potential in ordering discrete symbols (Figure 5). The link from Alberti to Turing arrives by way of representational machines that employ a discrete notational scheme. When this representation is set in motion, the complex ordering is not limited to language. Cryptography has long been used to investigate the natural world or create art, and is now poised to order the economy.

.


wheels are rings or tyres carrying alphabets , and rotatable with respect to the rest of the wheel; more about this under 'turnovers' When the machine is being used three of the wheels are put in between the U.K.W. and the E.W. in some prescribed order . The way that the current might flow from the E.W. through th e wheels and back is shewn below

Turnovers. Ringstellung. Window position . rod position.

Figure 5. Turing's notes on the Enigma machine (1939)

Bitcoin cryptography

Before showing how Bitcoin orders the economy we must understand how cryptography functions in Bitcoin. The cryptography used in Bitcoin is not unusual or exemplary; in fact, there are no cryptographic innovations in Bitcoin (in computer security terms, this is a virtue of the system). Bitcoin uses a standard SHA-256 hashing algorithm. This hashing algorithm is put to some seemingly strange uses, but nothing unique to the history of cryptography since the development of public key cryptography in the 1970s.

Private and public-key cryptography, hashing functions

For the vast majority of the history of cryptography—its development from substitution ciphers and code systems to polyalphabetic and keyed algorithms—the encrypting mechanism was unitary. With the invention of public (or asymmetric) key cryptography in the 1970s it became possible to create a system that "split" the cryptographic key.xvi Splitting the cryptographic key (explained below) ushered in new uses for cryptography, and was well timed for the coming advance of the Internet and popularization of point-to-point electronic communication.

In simple "code" systems (often called "nomenclators") a letter, word, or entire phrase may be replaced with an alternative, with the substitution presumably kept private between the two communicating parties (Kahn, 1967). The basic principle is captured by so-called substitution ciphers, which exchange one letter for another in a deterministic manner. From this mechanism more complicated forms of cryptography were invented. Polyalphabetic cryptography uses multiple alphabets for the substitution, sometimes jumping from one alphabet to another according to an agreed-upon secret "key."

For much of the history of cryptography no notion of cryptographic "key" existed. In more recent usage the key became analytically separated from the cryptographic algorithm, as a result of "industrial" uses of cryptography that require reusable ciphers. We now generally speak of an immutable (public) algorithm, and a mutable (private) key. In fact, the key is no more important than the set of transformations (and should be characterized as part of the system of transformation). For purposes of cryptanalysis ("codebreaking") a key is only as valuable as knowledge of the corresponding mechanism or set of transformations. Yet, in recent years cryptographic best practice requires keeping the key secret and the mechanism public (working on the assumption that the mechanism will eventually be discovered and reverse-engineered anyway).

Symmetric-key cryptography employs the same key for encryption and decryption, so the shared secret item is identical among parties. The obvious downside is that to maintain secret communications all parties must ensure that the shared key is kept private from any so-called adversaries. Modern forms of symmetric-key cryptography work on digital bits, either encrypting the bits one at a time (or, more realistically, encrypting byte by byte or in prescribed bit-length "words"), or grouping the bits into blocks (and adding padding as needed so that each block includes the same number of bits). Other non-cryptographic features may be present in a modern cryptographic system, such as error detection, compression, and so on.

The symbolic transformations in symmetric-key cryptography are fundamentally the same as that of asymmetric-key cryptography—in fact, symmetric-key primitives can be used to build asymmetric-key systems. The sole (but critically important) difference between symmetric-key and asymmetric-key cryptography is that rather than sharing a single (unitary) secret key, asymmetric-key cryptography uses a binary key, in which the two parts are linked and both are required to decrypt and encrypt. By splitting the key into two linked parts, one part can be kept secret, while the other is made public (the parts are typically linked mathematically—e.g., prime factorization or calculating the discrete logarithm—but any suitable mechanism could be used). The private key should not be easily deducible from the public key, yet the public key should be easily deducible from the private key (using so-called mathematical trap-door functions). A trap-door function such as exponentiation modulo is based on the mathematical belief that it is easy to calculate the remainder when a number is raised to some power, divided by another (the modulus), yet, if given all the information other than the exponent, it is very difficult to solve for the exponent (i.e., it is slow to compute the discrete logarithm). Only when the secret key is possessed is it easy to open the trap door, otherwise the calculation is slow (but certainly tractable).

Asymmetric-key setups offer several interesting possibilities. The key used to encrypt a message is not the same key used to decrypt it, so I may encrypt a message with your public key, in which only you can decrypt (using your linked private key). Configured this way, I can send you a message that only you can read (akin to placing a piece of mail in a publicly-accessible but locked mailbox), and you can do the same for me by using my public key, ensuring confidentiality.xvii Similarly, if I encrypt a message with my private key and send you both my (cleartext) message and the encrypted message (or a "digest" of it), your ability to decrypt the encrypted message with my public key ensures that the message is authentic (non-repudiated, i.e., guaranteed to be mine). These features work to permit secret message communication without ever requiring a secret communication channel, or to ensure that a message has not been changed. In a world where the communication channel is necessarily open to eavesdropping, asymmetrical-key cryptography performs a kind of magic trick: secret messages over public channels without ever requiring the prior transmission of secret information.

A related application of asymmetric-key cryptography is the hash function. A hash function is a set of permutations that transform some data into a fixed-sized output (a digest), which changes considerably given even a slightly different input datum. When a hash function uses cryptographic mechanisms to create the digest it can be used to ensure the message is not repudiated (its integrity can be verified), which is especially useful for creating compact, easily transmittable "fingerprints" of data. Similarly, by computing a hash for a password, the hash may be stored in place of the secret password, and authenticated against the digest representation instead of the actual password (allowing the digest to be captured by an adversary without losing secrecy).

The hash algorithm used in Bitcoin is SHA-256, a protocol for hashing in the Secure Hash Algorithm 2 family that outputs 256 bit digests. SHA-256 is composed of a simple set of logic transformations configured to perform the necessary mathematical trap-door function required by the asymmetric-key cryptography. Thus, described in as many of levels of abstraction as I can muster, from the basics of digital computation to a full ASCII-encoded hash digest, the SHA-256 hash algorithm operates as follows. Electromagnetic flux is discretized on a clock-cycle. Bits are then transformed using logical operations performed by transistors. The binary representation is collected into 64 bit "words" that function as high-level data structures (integers). Mathematical trap-door/one-way functions are constructed from the set of transformations (+, and, or, xor, shr, rot), with the entire algorithmic structure corresponding to the Merkle–Damgård padding and compression scheme (Figure 6). The resulting 256-bit hash digest may then be encoded into an ASCII-encoded character string for portability and human-readability.

In sum, SHA-256 is the process of successively interpreting electric-magnetic flux as a series of different ordering mechanisms or techniques. The primary ordering is the first: from flux to binary, and once this discretization is accepted as real (by fiat) the ordering techniques are limited only by human imagination.

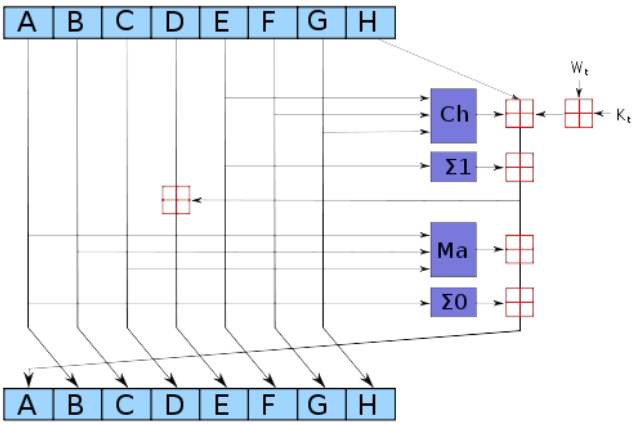


Figure 6. Schematic of SHA-256 algorithm (Kockmeyer, 2007)

Bitcoin specifics

The fundamental cryptographic algorithm used in Bitcoin is SHA-256, however, its conceptual utility draws on a recent history of academic and practical developments. Of the numerous developments, the most significant and relevant are: Ralph Merkle's hash-trees (patent filed in 1979), David Chaum's blind signatures (1982), Adam Back's hashcash proof of work system (1997), Wei Dai's b-money scheme (1998), Nick Szabo's bit gold concept (1999), and Hal Finney's reusable proofs of work (2004).

In addition to having a hand in the invention of public key cryptography, Ralph Merkle developed a system for efficiently verifying large data structures through a tree structure of hash digests (Merkle, 1982). As described above, a hash digest can be used to verify the non-reputability of a datum, but for large data structures it would be extremely time-consuming to perform a hash function on every datum. Merkle realized that by organizing hash digests into a tree structure (where each node is a hash digest) it is possible to compute the hash digest for only the top-most node (while authenticating the left and right nodes) rather than every node, to ensure non-reputability. Hash trees are commonly used to ensure data integrity, but when used with cryptographic hash functions every prior message is checked for authenticity (none of the messages can be faked).

Blind signatures are another result of public key cryptography being used in unexpected ways. In Chaum's original concept for blinded signatures,

payment systems with the anonymity of cash but the security of digital money (like Bitcoin) were the intended target (Chaum, 1982). By using public key cryptography Chaum proposed a system that ensured 1) the inability of third parties to determine information about the payee, 2) the ability of individuals to provide proof of payment, and 3) the ability to stop payments when needed (in cases of theft). Chaum envisioned a digital equivalent of paper envelopes lined with carbon paper. By writing a signature on the outside of the envelope a second "blind" signature is duplicated on the inside. In Chaum's example of authenticated secret voting, the blinded signature is sent to the elector, removed from the envelope, signed by the elector, and mailed back to the voter in a new envelope (thus only the elector views the signature). If a voting dispute arises the signatures can be authenticated against the signatures on the envelopes, but each vote remains anonymous.

While the mutability of binary digits is useful for much computing, a system of electronic cash requires the opposite quality: money needs to be made solid, slow, and non-(token)replicable. Originally proposed and developed by Adam Back (1997) for limiting email spam, hashcash uses two facts of public key cryptography: non-reputability of hash digests, and the computational difficulty of finding a hash "collision." Due to the fact that it is nearly impossible to predict the outcome of a hash function on an arbitrary input (with current knowledge of the mathematical underpinnings of asymmetric-key cryptography used in hash functions), but easy to verify the results, a challenge-response mechanism can be created to require "work" (computational effort). By arbitrarily requiring a specified output of a hash function—such as that the first 20 or more bits of the hash digest must be zeros—the sender can establish a "difficulty" threshold. The only way to find a hash digest with a specified output is to compute the hash of a different input value over and over until the result meets the necessary difficulty, and since the result can be verified easily the receiver of the hash function does not need to repeat the computational work to verify that the sender expended a set amount of work. For its original purpose of limiting email spam, the requirement to perform work when sending email would make sending email slow and computationally expensive, thus, sending a single email would result in a modest slow-down, but sending millions would become nearly impossible (or at least would require many expensive computers). In such an email system any email that was sent without corresponding evidence of computational work would not pass verification by the receiver (which is a quick calculation, in comparison to performing the original hash calculation), and would be discarded as spam.

As part of the discussion of applications to Back's hashcash proposal on the Cypherpunks mailing list, Wei Dai proposed a system of currency generation using Back's mechanism (Dai, 1998).xviii Dai applies Back's hashcash mechanism in an effort to create a world where cryptography functions as the "medium of exchange," and as a way to "enforce contracts" without the intervention of a government (Dai, 1998). Dai's protocol for the creation of bmoney requires a specified amount of computational work (that anyone can perform), which is then verified by the community who update a collective ledger book, awarding the worker the specified funds. In the bmoney proposal, exchange of funds is accomplished by collective bookkeeping (authenticated with cryptographic hashes), and contracts are enforced through the broadcast and signing of transactions with digital signatures (i.e., public key cryptography).

Hal Finney (2004) extended the bmoney and hashcash proposals by suggesting a formalization of the proof of work mechanism, a scheme that permits the reuse and exchange of proof of work tokens (hash digests). With these extensions it became possible for Nick Szabo (Szabo, 2008, 1997) to conceive a system that accurately calculates the "difficulty" of proof of work for the purpose of money generation, and to allow the generated money (hash digests) to be exchanged and reused.

A Bitcoin transaction

When the pseudonymous programmer Satoshi Nakamoto proposed Bitcoin in 2008 it built on the crypto-anarchist developments from the following two decades (Nakamoto, 2008). In terms of invention, Bitcoin introduced a modest change to the bmoney, bit money (and other) proposals already in existence. Rather than require a single collective ledger of transactions, or awkwardly share the ledger among parties, Nakamoto suggested that a "blockchain" contain all transactions (including generated money by "miners" performing cryptographic proofs of work, described in more detail below). The blockchain is a Merkle hash tree of transactions. For each transaction the mining servers verify the hash digests that result from transactions, incentivised to perform computational work by being awarded money for successfully performing proofs of work. The transactions are verified by the miners and grouped into "blocks"; once the top node of each block is verified a specified amount of work is required to seal the block and win the resulting money.

A full round of a transaction requires several steps. Money is stored in a wallet, which is a unique hash digest generated by each user (and any number of wallets are possible). To send you money I first digitally sign the transaction request with my private key (that is, I perform asymmetric-key encryption on the transaction request data). Using my public key, the network can verify my transaction request. The transaction request is sent (via peer-to-peer communication protocols) to the network and then bundled with others into blocks every ten minutes. Each block includes a hash digest of the previous blocks (arranged in a hash tree), a hash digest of the current block, and a "nonce". A nonce is added input that (when hashed) results in a radically different output. Only when an output value meets a certainly "difficulty" (proof of work) will a block be considered authenticated (the difficulty is specified by requiring at least *n* leading zeros in the hash digest output, set by the protocol to regulate the speed of block generation). As each subsequent block is verified the previous blocks fall further down the hash-tree, with the newer hashes contingent on the previous hash digests' value. In this way any fraudulent changes to the blockchain are instantly discovered (and rejected).

It is improbable to create fraudulent transactions because as time goes on, and block upon block is verified by the miners, fraudulent transactions would require changing every subsequent block, at a rate greater than the (legitimate) network of miners. A fraudulent block would only be accepted if the alternative blockchain is longer, which would require performing more proofs of work than the legitimate network.xix

Bitcoin praxis

In many ways the Bitcoin miner is at the heart of Bitcoin cryptography, since it creates money and verifies transactions. Starting in 2013, I engaged in the practical operation of a Bitcoin miner. What follows is a description of this Bitcoin praxis as I learned the details of the protocol.

As described above, the Bitcoin algorithm uses the SHA-256 method of computing hash digests. While any computing mechanism could in theory calculate a SHA-256 hash, there are certain reasons why conventional Central Processing Unit (CPU) mining is now rarely used. With even the fastest modern CPUs running software designed to take advantage of multithreaded computation the number of hashes computed per second is low compared to other technologies, and because the ability to "win" the awarded money for a successfully verified block is in competition with other miners, an arms race is always at hand.

CPUs are usually designed to manage and switch computational tasks, and take care of a variety of sub-processes, which makes a CPU ideal for general computing but inefficient for performing the same type of simple calculation repeatedly. Commercially available Graphics Processing Units (GPUs) are designed to be relatively free from the management of resources and thus (when appropriately programmed using low-level software) are able to perform repeated calculations much faster than CPUs. Additionally, GPUs are designed to work in parallel, so while a multi-core, multithreaded CPU may be able to perform a certain amount of work in parallel, a modern GPU can perform thousands of computations in parallel. Some GPUs are ideally suited to perform SHA-256 calculations because

they have been designed to perform XOR logic in a single step, rather than the two steps (or cycles) needed for other devices (the SHA-256 algorithm relies extensively on XOR transformations). For real-world comparison, on a slightly aging (2008) Mac Pro computer I was able to perform roughly 30 Kh/s (kilo hashes per second, or thousands of hashes per second) using the CPU (a server-grade 3GHz quad core Intel Xeon processor). When I installed a modern mid-level gaming video card (AMD Radeon HD 5850) with a dedicated GPU the same machine was able to perform roughly 350 Mh/s (millions of hashes per second), using only the GPU for calculating the hashes.

While 350 Mh/s may seem like considerable computational power—and it is, especially for the corollary purpose of password cracking—newer technologies have all but obsoleted GPU Bitcoin mining. For the last several years more dedicated Bitcoin mining individuals have purchased Field Programmable Gate Array (FPGA) devices that are tailored to perform these sorts of computational tasks, doing so much more quickly and with less power consumption. By the end of 2012 the newest type of Bitcoin mining device entered the commercial market, eclipsing even FPGA devices in terms of speed and power efficiency. These Application-Specific Integrated Circuit (ASIC) devices are custom-designed for Bitcoin mining and thus do so with remarkable speed and power efficiency. As of 2013 there are commercially available Bitcoin miners available for $150 USD that perform 5 Gh/s (billions of hashes per second) and use only 30 watts of power (compared to an average video card's consumption of 100-150 watts), with more expensive versions performing hundreds or even thousands of Gh/s.

Efficient Bitcoin mining is only possible using specially built software, tailored to take advantage of built-in hardware capabilities. On GPUs the programming language used to write the portion of software that performs the hash calculations is typically Compute Unified Device Architecture (CUDA) or Open Compute Language (OpenCL), whereas the high-level software that controls the input/output, networking, and display of graphical user interfaces can be written in any suitable programming language. Contemporary cryptography algorithms are highly repetitive, requiring round after round of simple logic transformations, just like digital signal processing, big data and science computing, and gaming (computing millions of polygons). For this reason, cryptography shares many of the technological advances with these computing fields.

Politics of Bitcoin

As demonstrated above, a more expansive view of cryptography suggests that cryptography can be used for more than just secrecy; it can be fruitfully understood as a notational system. When this notational system is operationalized it orders its symbolic input according to particular logics. When these ordered symbols represent the world, as in the case of economic transactions, a politics of ordering is present.

That the economy is a "slave to the algorithm" (Slater, 2013) is not due to Bitcoin; dominant capital is now almost exclusively run through digital trading software (acknowledged as "high-frequency trading"), and much of the developed world's "cash" passes digitally direct from bank to merchant (to bank) through debit or credit machines at point-of-sale terminals. Cash money already seems quaint: the stuff of slightly unscrupulous transactions (a manual trade exchanged for tax-free cash payment), or downright illegal transactions (purchasing drugs).

The politics of this "new" economy can be read in light of Gilles Deleuze's short "Postscript on the societies of control" (1992). Deleuze summarizes Foucault's (1979) periodization of history by reflecting on the "transience of the model," noting that the sovereign society was replaced by a disciplinary one—a transition occurring roughly at the dawn of the eighteenth century. According to Deleuze, the next shift occurs at the outset of the twentieth century, towards a control society (Deleuze, 1992). Deleuze charges his readers not to "fear or hope [for]" these mechanisms but instead to "look for new weapons" (Deleuze, 1992, p. 4). I argue that cryptography is one such new weapon in the control society: controlling economics through the ordering application of Bitcoin.

Deleuze argues that the control society is characterized by modulation, rather than the "molds, distinct castings" of the prior disciplinary society (Deleuze, 1992). In the English translation "modulation" evokes the Latin modus, meaning measurement, accomplished by numbers ("numerical"). Similarly, each society is characterized by a kind of language: the disciplinary society is *analogical* and the control society is *numerical*. Yet, Foucault spends considerable effort in *Order of Things* (2002) showing how measurement and number are still foundational to the logic of the classical era, summed up in the term "order."

While it is commonly believed that cryptography is mathematical (mathematicians, after all, are the gatekeepers of contemporary cryptography), above I showed how mathematics is but one of many ways to enact a deeper ordering. The notational system—discrete marks capable of being rearranged—is, I argue, the heart of cryptography. Numeracy is an after-effect of the deeper logic that Leibniz, Turing, and others recognized—and proven daily by capable software developers with almost no mathematical skill. So long as you can represent objects within a notational system, the world can be ordered with no mathematics at all.

Yet, Deleuze thinks that the mechanism of the control society is modulation or measurement, and Kittler too thinks that by counting the Greek alphabet we arrive at all the mathematical truths of the world (Winthrop-Young, 2011). It is exactly the infinite variability of measurement that cannot be ordered, since there is always a more precise measurement possible (Goodman's formalization explicitly rules out continuous measurement). Kittler's gramophone records the smooth contours of a voice perfectly, but the result is not susceptible to decomposition, and therefore cannot be ordered. It is only by ignoring the measurements (continuous waves) that ordering technologies are able to function. When we discretize the voice we perform violence, and we lose the knowledge of what king Akhneten called his N-f-r-t-t. For Bitcoin, the smooth transfer of cash (previously "filthy lucre") from hand-to-hand is replaced with an algorithmic logic.

In our society, the largely invisible but very powerful effects of control often result from ordering technologies. Algorithmic technologies are able to sort, move, and re-arrange entire populations in ways that mimetic technologies are unable to accomplish. The Hollerith tabulator, an early proto-computer, was effective in aiding the Nazi extermination of Jews (Luebeke and Milton, 1994) because it could collect, process, and order entire populations in ways that mimetic technologies such as radio or television were incapable of.

Cryptography now functions infrastructurally, and sits invisibly behind most of the world's communications (machinic "data" transmissions such as encrypted Netflix TV streams or Trusted Computing modules, but also human correspondences such as email communications). As Bitcoin and other electronic crypto-cash systems become prevalent, cryptographic ordering will become more entrenched in the economic realm (it already functions invisibly at electronic point-of-sale machines, automated bank tellers, financial trading, and so on).

For Bitcoin, the specific ordering is the logic of SHA-256 described above: discrete symbols arranged through a collection of logical transformations, built block by block of irreversible containers with strong identity parameters (necessarily discrete and disjoint). The hash digests are then organized into a tree structure. Finally, the hash tree is sent through peer-to-peer networks, succumbing to a logic of collusion and virality.

Cryptography is "code," and code is cryptography. Code is powerful because it represents, that is, it both "re-presents" or makes something present again, and "stands for" or "substitutes" (Prendergast, 2000). Code worries us for the reasons that Rousseau rallied against political representation, fearing a dictatorial relationship where we permit others to "stand in" for us (2003). Similarly, Heidegger calls representation the

master category of modern thought because it forces the division of subject and object (Heidegger, 2002; Prendergast, 2000).

More concretely, our cryptographic technologies are at once Privacy Enhancing Technologies and also weapons in the commonplace cyberwars amongst developed nations. Even prosaic questions become ambiguous. Does Google increase my privacy by encrypting my Gmail communications, now open to only the machinic display of advertising? Am I better off by having a cryptographically secure boot sequence for my computer (and thus preventing the installation of "competitor" operating systems, like Linux)? Is it a "feature" that Bitcoin transactions are cryptographically irreversible? In this new control society there is no second-guessing your economic decisions, and no need to involve messy legal and political authorities, since code has become law in frighteningly efficient ways (Lessig, 2006).

This paper showed how the traditional (secrecy) conceptualization of cryptography is wanting ; in its place, I argued for a new conceptualization as a notational system. As Leibniz and Turing understood, a notational system can order the world in powerful ways. Reading the politics of Bitcoin in light of a reimagination of Deleuze's control society (replacing modulation with order) I suggested that cryptography is a powerful "new weapon," functioning as an ordering machine. I argued that Bitcoin is at the forefront of advancing this ordering logic, updating the economic system for our new control society.

Quinn DuPont
quinn.dupont@utoronto.ca
Faculty of Information, University of Toronto

Endnotes

i Mt. Gox was the most popular bitcoin exchange, and dealt primarily in USD, although many other exchanges are available (some with regional or jurisdictional foci). Mt. Gox started as a trading card exchange for the online game Magic: The Gathering Online, but eventually rebranded to focus exclusively on bitcoin.

ii This was the first spike in value for 2013; the second (much larger) spike occurred a few months later when bitcoins were trading above 1000 USD/bitcoin, often swinging more than 50% in value in a matter of hours.

iii The recent revelations of US military contractor Edward Snowden perpetuate this ideology, even if they eroded the universal trust in cryptography. For many, the response was simply a need for better cryptography, seemingly safe from the revealed fact that US Intelligence services can readily crack or circumvent available cryptography.

iv The Electronic Frontier Foundation advocates the widespread use of cryptography (Opsahl et al., 2013); see below for evidence of bitcoin's origins on the Cypherpunks mailing list.

v See for example Rosenheim (1997) and Shoptaw (2000) on literature and poetics, Ellison (2011, 2008) on textual studies, Gleick (2011) on information, Pesic (2000a, 2000b, 1997) on science, Blanchette (2012) on documentation, Eco (1986) on semiotics, and Zielinski (2008) on media.

vi See the journal of record, Cryptologia, for the history of cryptography. While many of the articles published in Cryptologia are the only source of history and should be applauded for their trailblazing work, most are published by hobbyists or experts in engineering who have an interest in historical matters. Even Kahn himself, who over the years has become a highly capable self-trained historian, started as a journalist. These histories, correspondingly, usually suffer from the evils of bad historiographical methodology: "impact" accounts of technical change, over-reporting of military technologies, and whiggish, teleologically-driven narratives.

vii Although written prior to his famous Essay, Mercury was published posthumously.

viii For an assessment of universal and philosophical language schemes, with occasional reference to the role of cryptography, see (Cohen, 1954; Formigari, 2004, 1993, 1988; Maat, 2004; Rossi, 2000; Salmon, 1971, 1966; Slaughter, 1982; Wilding, 2001).

ix Note that Glidden (1987) argues that the French version (here pictured) of Trithemius' Polygraphiae contains a slightly artificial "reinforced" link to Lullian and cabalistic influences.

x Cryptography is often twinned with war. At times these associations can be productive (Deleuze and Guattari, 1987; Kittler, 1999), but far too often they constrain our thinking, leading to discourse (and associated models) of "adversaries," "man-in-the-middle," "attack" and so on. The bellicose nature of cryptography cannot be dismissed—it is a war machine—but I think we should derive, not assume, such connections.

xi Although only occasionally recognized, the histories of information and cryptography are intimately tied (see e.g., Gleick, 2011).

xii "Discrete" and "discreet" share the Latin origin discretus.

xiii For a more extensive description of the dialectic of mimetic and algorithmic technologies see Takhteyev and DuPont (n.d.).

xiv The skytale has been disqualified as cryptography simply because it doesn't offer plausibly good secrecy. As I have been trying to argue, we need to think more expansively about our conceptualization of cryptography, and may want to return to the original understanding of the skytale as an encryption device, but one that offers something other than secrecy.

xv It is worth noting that Alberti's cryptographic work was also influenced by Lull (Kahn, 1980).

xvi Asymmetric-key cryptography was initially invented in 1973 at the Government Communication Headquarters in the UK by James Ellis, Clifford Cocks, and Malcom Williamson but kept secret; it was then publically re-invented in 1976 by Whitfield Diffie and Martin Hellman (Levy, 2002).

xvii More complicated three-pass schemes are possible too, in which I encrypt a message and pass it to you, then you encrypt the already-encrypted message and pass it back, I then decrypt my encryption and pass it back to you, at which point you can finally decrypt your encryption and read the message—successfully transferred in public while remaining encrypted.

xviii At this point the anarchist/libertarian undercurrents are completely at the fore, Dai (1998) starts his proposal, "I am fascinated by Tim May's crypto-anarchy."

xix Of course, this is the ideal scenario. Any competing network of greater computational power could best the legitimate one, and therefore human vagaries of collusion and consolidation come to play. This has actually happened, when due to a technical bug the blockchain became "forked," and was only reset when a cartel of powerful mining pools colluded to switch to the "corrected" blockchain.

References

Alberti, G. B. (2010). De componendis cifris. In K. Williams, L. March, & S. R. Wassell (Eds.), The Mathematical Works of Leon Battista Alberti (pp. 169–187). Basel: Springer Basel. Retrieved from http://link.springer.com.myaccess.library.utoronto.ca/content/pdf/10.1007%2F978-3-0346-0474-1_4

Aspray, W. F. (1985). The Scientific Conceptualization of Information: A Survey. Annals of the History of Computing, 7(2), 117–140.

Back, A. (1997, March 28). hash cash postage implementation. Cypherpunks. Retrieved from http://www.hashcash.org/papers/announce.txt

Bacon, F. (1762). Novum organum scientiarum. Venetiis: Typis G. Girardi.

Bauer, F. L. (2005). Cryptology. In Encyclopedia of cryptography and security. New York: Springer. Retrieved from http://link.library.utoronto.ca/eir/EIRdetail.cfm?Resources__ID=604588&T=F

Bellman, B. L. (1979). The Paradox of Secrecy. Human Studies, 4(1), 1–24.

Blanchette, J.-F. (2012). Burdens of Proof: Cryptographic Culture and Evidence Law in the Age of Electronic Documents. MIT Press.

Carpo, M. (2011). The Alphabet and the Algorithm. Cambridge, Mass.: MIT Press.

Chaum, D. (1982). Blind Signatures for Untraceable Payments. In R. L. Rivest, D. Chaum, & A. T. Sherman (Eds.), (pp. 199–203). Presented at the Advances in Cryptology Proceedings of Crypto 82, Plenum. Retrieved from http://blog.koehntopp.de/uploads/Chaum.BlindSigForPayment.1982.PDF

Cherry, E. (1953). A history of the theory of information. Information Theory, IEEE Transactions on, 1(1), 22–43.

Cohen, J. (1954). On the Project of a Universal Character. Mind, 63(249), 49–63.

Cooke, M. (1983). Ibn Khaldun and Language: From Linguistic Habit to Philological Craft. Journal of Asian and African Studies, 18(3-4), 179–188.

Dai, W. (1998, November 26). PipeNet 1.1 and b-money. Cypherpunks. Retrieved from http://marc.info/?l=cypherpunks&m=95279516022393&w=2

Deleuze, G. (1992). Postscript on the Societies of Control. October, 59(Winter), 3–7.

Deleuze, G., & Guattari, F. (1987). A Thousand Plateaus: Capitalism and Schizophrenia. (B. Massumi, Trans.). University of Minnesota Press.

Derrida, J. (1998). Of Grammatology. (G. C. Spivak, Trans.) (Corrected.). Baltimore and London: The Johns Hopkins University Press.

Eco, U. (1986). Semiotics and the Philosophy of Language. Bloomington, Indiana: Indiana University Press.

Ellison, K. (2008). Cryptogrammatophoria: The Romance and Novelty of Losing Readers in Code. Eighteenth Century Fiction, 20(3), 281–305.

Ellison, K. (2011). Millions of Millions of Distinct Orders: Multimodality in Seventeenth-Century Cryptography Manuals. Book History, 14(1), 1–24.

Finney, H. (2004, August 15). RPOW – Reusable Proofs of Work. Cypherpunks. Retrieved from http://marc.info/?l=cypherpunks&m=109259877510186&w=2

Formigari, L. (1988). Language and Experience in 17th-Century British Philosophy. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Formigari, L. (1993). Signs, Science, and Politics: Philosophies of Language in Europe, 1700-1830. (W. Dodd, Trans.). Amsterdam/Philadelphia: John Benjamins Publishing Company.

Formigari, L. (2004). A History of Language Philosophies. (G. Poole, Trans.). Philadelphia: John Benjamins Pub.

Foucault, M. (1979). Discipline and Punish: The Birth of the Prison. (A. Sheridan, Trans.). New York: Vintage Books.

Foucault, M. (2002). The Order of Things: An Archaeology of the Human Sciences. New York: Routledge.

Gardner, M. (1958). Logic Machines and Diagrams. New York: McGraw-Hill.

Geoghegan, B. D. (2008). Historiographic Conceptualization of Information: A Critical Survey. IEEE Annals of the History of Computing, 30(1), 66–81.

Gleick, J. (2011). The Information: A History, a Theory, a Flood (1st ed.). New York: Pantheon Books.

Glidden, H. H. (1987). Polygraphia and the Renaissance Sign: The Case of Trithemius. Neophilologus, 71(2), 183–195. doi:10.1007/BF00209168

Goodman, N. (1976). Languages of Art: An Approach to a Theory of Symbols. Indianapolis: Hackett Publishing.

Hayles, N. K. (1999). How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics. University of Chicago Press.

Heidegger, M. (2002). The Age of the World Picture. In J. Young & K. Haynes (Trans.), Off the Beaten Track (pp. 57–85). Cambridge UK: Cambridge University Press.

J.M.P. (2013, December 25). Bitcoin paradise. The Economist. Retrieved from http://www.economist.com/blogs/schumpeter/2013/12/libertarian-enclaves

Kahn, D. (1967). The Codebreakers: The Story of Secret Writing. New York: Macmillan.

Kahn, D. (1980). On the Origin of Polyalphabetic Substitution. Isis, 71(1), 122–127.

Kelly, T. (1998). The Myth of the Skytale. Cryptologia, 22(3), 244–260. doi:10.1080/0161-119891886902

Kittler, F. (1990). Discourse Networks 1800/1900. Stanford, Calif.: Stanford University Press.

Kittler, F. (1999). Gramophone, Film, Typewriter (1st ed.). Stanford University Press.

Kockmeyer, B. (2007). A schematic that shows the SHA-2 algorithm. Retrieved from http://en.wikipedia.org/wiki/File:SHA-2.svg

Kohno, T., Ferguson, N., & Schneier, B. (2010). Cryptography Engineering: Design Principles and Practical Applications. Indianapolis, IN: Wiley Publishing, Inc.

Lateiner, D. (2005). Signifying names and other ominous accidental utterances in classical historiography. Greek, Roman, and Byzantine Studies, 45(1), 35–57.

Leibniz, G. W. (1666). Dissertatio de Arte Combinatoria. Leipzig: Joh. Simon Fickium et Joh. Polycarp. Retrieved from http://www.rarebookroom.org/Control/leiart/index.html?page=7

Leibniz, G. W. (1989). Dissertation on the Art of Combinations. In L. E. Loemker (Ed.), Philosophical Papers and Letters (pp. 73–84). Dordrecht / Boston / London: Kluwer Academic Publishers.

Lessig, L. (2006). Code v2. Basic Books.

Levy, S. (2002). Crypto: Secrecy and Privacy in the New Code War. London: Penguin.

Luebeke, D. M., & Milton, S. (1994). Locating the Victim: An Overview of Census-Taking, Tabulation Technology and Persecution in Nazi Germany. IEEE Annals of the History of Computing, 16(3), 25–39.

Maat, J. (2004). Philosophical Languages in the Seventeenth Century: Dalgarno, Wilkins, Leibniz: Dalgarno, Wilkins, Leibniz. Springer.

Manske, M. (2005). Four rotor German naval Enigma on display at Bletchley Park. Retrieved from http://en.wikipedia.org/wiki/File:Bletchley_Park_Naval_Enigma_IMG_3604.JPG

Mackenzie, A. (1996). Undecidability: The History and Time of the Universal Turing Machine. Configurations, 4(3), 359–379. doi:10.1353/con.1996.0020

Merkle, R. C. (1982, January 5). Method of providing digital signatures. Retrieved from http://www.google.com/patents?id=Dd4zAAAAEBAJ

Montiglio, S. (2000). Silence in the Land of Logos. Princeton, N.J.: Princeton University Press.

Nakamoto, S. (2008, October 31). Bitcoin P2P e-cash paper. Cypherpunks. Retrieved from http://marc.info/?l=cryptography&m=122558139832010&w=2

Opsahl, K., Cardozo, N., & Higgins, P. (2013, December 16). UPDATE: Encrypt the Web Report: Who's Doing What. Electronic Frontier Foundation. Retrieved December 17, 2013, from https://www.eff.org/deeplinks/2013/11/encrypt-web-report-whos-doing-what

Pesic, P. (1997). Secrets, Symbols, and Systems: Parallels between Cryptanalysis and Algebra, 1580-1700. Isis, 88(4), 674–692.

Pesic, P. (2000a). Labyrinth: A Search for the Hidden Meaning of Science. Cambridge, Mass.: MIT Press.

Pesic, P. (2000b). The Clue to the Labyrinth: Francis Bacon and the Decryption of Nature. Cryptologia, 24(3), 193–211. doi:10.1080/01611190008984242

Poe, E. A. (1991). The Gold-Bug and Other Tales (Dover Thrift Editions.). Mineola, N.Y.: Dover Publications.

Poe, E. A. (2013). The Gold Bug (Unabridged.). Audio Books by Mike Vendetti.

Prendergast, C. (2000). The Triangle of Representation. New York: Columbia University Press.

Rosenheim, S. (1997). The Cryptographic Imagination: Secret Writings from Edgar Allen Poe to the Internet. Baltimore: The Johns Hopkins University Press.

Rossi, P. (2000). Logic and the Art of Memory: The Quest for a Universal Language. London: Athlone Press.

Rousseau, J.-J. (2003). On the Social Contract. (G. D. H. Cole, Trans.). New York: Dover Publications.

Salmon, V. (1966). Language-Planning in Seventeenth-Century England; Its Context and Aims. In C. E. Bazell, J. C. Catford, M. A. K. Halliday, & R. H. Robins (Eds.), In Memory of J.R. Firth (pp. 370–395). London: Longmans, Green and Co Ltd.

Salmon, V. (1971). The Evolution of Dalgarno's "Ars Signorum." In Studies in Language and Literature in Honour of Marget Schlauch (pp. 353–371). New York: Russell & Russell.

Schmeh, K. (2012). The Pathology of Cryptology—A Current Survey. Cryptologia, 36(1), 14–45. doi:10.1080/01611194.2011.632803

Shannon, C. (1945). A Mathematical Theory of Cryptography (No. 20878). New Jersey: Bell Labs. Retrieved from http://www.cs.bell-labs.com/who/dmr/pdfs/shannoncryptshrt.pdf

Shannon, C., & Weaver, W. (1948). A Mathematical Theory of Communication. Bell Syst. Tech. J, 27, 379–423.

Shoptaw, J. (2000). Lyric Cryptography. Poetics Today, 21(1), 221–262.

Slater, J. B. (Ed.). (2013). Slave to the Algorithm (Vol. 3). Mute.

Slaughter, M. M. (1982). Universal Languages and Scientific Taxonomy in the Seventeenth Century. Cambridge University Press.

Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. First Monday, 2(9). doi:10.5210/fm.v2i9.548

Szabo, N. (2008, December 27). Bit gold. Unenumerated. Retrieved January 5, 2014, from http://unenumerated.blogspot.ca/2005/12/bit-gold.html

Takhteyev, Y., & DuPont, Q. (n.d.). Ordering Space: Alternative Views of ICTs and Geography. Manuscript in Preparation.

Thomsen, S. W. (2009). Some evidence concerning the genesis of Shannon's information theory. Studies in History and Philosophy of Science, 40(1), 81–91.

Tritheme, M. I. (1561). Polygraphie: Universelle escriture Cabalistique de M.I. Tritheme Abbé. Paris: Jaques Keruer. Retrieved from http://fantastic.library.cornell.edu/bookrecord.php?record=F034

Turing, A. (c. 1939-42). Turing's Notes on the Enigma Machine (Catalogue reference: HW 25/3). Retrieved from http://www.nationalarchives.gov.uk/spies/ciphers/enigma/en1.htm

West, S. (1988). Archilochus' Message-Stick. The Classical Quarterly, 38(1), 42–48.

Wilding, N. (2001). "If You Have A Secret, Either Keep It, Or Reveal It": Cryptography and Universal Language. In D. Stolzenberg (Ed.), The Great Art of Knowing: The Baroque Encyclopedia of Athanasius Kircher (pp. 93–103). Firenze, Italia: CADMO.

Wilkins, J. (1668). Essay Towards a Real Character and a Philosophical Language. London: Royal Society.

Wilkins, J. (1694). Mercury: or, The secret and swift messenger. Shewing, how a man may with privacy and speed communicate his thoughts to a friend at any distance. London: R. Baldwin.

Winthrop-Young, G. (2011). Kittler and the Media. Cambridge, UK; Malden, MA: Polity Press.

Zielinski, S. (2008). Deep Time of the Media: Toward an Archaeology of Hearing and Seeing by Technical Means. (G. Custance, Trans.). The MIT Press.