CRYPTOCASH, CRYPTOCURRENCIES, AND CRYPTOCONTRACTS

NEAL KOBLITZ AND ALFRED J. MENEZES

Abstract. One of the central challenges for mathematical cryptography is to create a payment system that provides the advantages of cash in a digital world. In this expository article we describe two very different solutions to this problem. The first is an elliptic-curve-based version of a construction of S. Brands, and the second is Bitcoin. We also discuss a generalization of Bitcoin that supports peer-to-peer contracts.

AMS subject classifications. 94A60, 68P25, 14G50, 94-02.

1. Introduction

Throughout most of history cash was king. In the U.S. as recently as the 1970s most people used cash for the vast majority of their purchases. Expensive items were usually paid for with a personal check; since out-of-town checks were rarely accepted, in order to avoid having large amounts of cash on one's person tourists were advised to purchase travelers' cheques. Although credit cards were starting to be widely used, supermarkets and most shops — except for upscale ones — did not accept them.

In most other countries the dependence on cash was even greater. For example, in the Soviet Union neither credit cards nor checking accounts were available to consumers. You would receive your salary as a fistful of rubles at the end of each month. If you were fortunate enough to be able to buy a car, you would pay with a large stack of bills.

This dependency on cash might seem primitive to us in the 21st century, but it had certain advantages from the standpoint of efficiency and privacy. With cash there is no transaction cost and no involvement of a bank in each purchase. Neither the merchant nor the bank has to learn the identity of the buyer. No one keeps a detailed record of each consumer's buying habits.

In the 1980s and 1990s, as the predominant form of payment in many countries shifted from cash to credit cards, David Chaum, Stefan Brands, and other cryptographers worked to develop ways to recapture the advantages of cash in an economy based increasingly on e-commerce. This was no easy task.

In early electronic cash systems the players are the customer (user) $\mathcal{U}$, the merchant $\mathcal{M}$, and the bank $\mathcal{B}$. The three stages (protocols) in the system

Date: 1 October 2015.

Key words and phrases. Cryptography, public key, cryptocurrency, Bitcoin, Ethereum.

are withdrawal from the customer's account, payment to the merchant, and the merchant's deposit of the funds in the bank. In later systems the bank is replaced by a peer-to-peer network, and the withdrawal and deposit stages are transformed respectively into earlier and later transactions.

Trying to achieve all the desirable goals for electronic cash is a difficult challenge. The most important objectives are the following:

(i) security — no one should be able to counterfeit an e-coin or spend the same e-coin twice;
(ii) privacy — the payment method should not reveal the customer's identity, and it should not be possible to trace the previous owners of an e-coin or the previous transactions that were made with it;
(iii) efficiency — the payment infrastructure should be fast and relatively simple and should not require expensive hardware.

At first it might seem that it is not only difficult, but impossible to achieve all of these goals at the same time. Since an e-coin, by definition, is just a bitstring with no physical substance, it can be trivially duplicated. Don't anonymity and untraceability imply that there's no way to find someone who fraudulently spends the same e-coin in two transactions? Remarkably, this apparent contradiction is not insurmountable, and mathematical solutions can be found, as we shall see.

2. Cryptocash with a Central Authority

The first constructions of currency systems based on cryptographic one-way functions were due to David Chaum [4] over 30 years ago. They were based on the RSA function. We shall give a more efficient construction, due to Stefan Brands [2], that can be implemented using Elliptic Curve Cryptography (ECC).

The main purpose of these systems is to duplicate the anonymity and untraceability properties of cash while allowing remote electronic purchases. Another important feature of cash that one wants to carry over to e-commerce is that a "coin" that a merchant receives from a buyer does not have to be deposited immediately; he can send the merchandise right away and deposit the payment coins later with the assurance that they will be accepted by the bank. We call this an "offline" system.

In the systems discussed in this section one assumes that a standard currency (such as the dollar) will be used, and a central banking authority will manage electronic deposits and withdrawals. In §3 we will discuss Bitcoin, which rejects these assumptions.

Mathematical ingredients. Let $E(\mathbb{F}_q)$ be the group of $\mathbb{F}_q$-points of an elliptic curve E defined over the finite field $\mathbb{F}_q$. As usual in ECC, we assume that the group order $\#E(\mathbb{F}_q)$ has a prime factor p of bitlength almost as great as that of q, and we let $\mathcal{G}$ denote the subgroup of $E(\mathbb{F}_q)$ of order p. Although we write the group law additively and speak of point "multiples" rather than "powers," we think of $\mathbb{Z}/p\mathbb{Z}$ as the "exponent space" of $\mathcal{G}$ by analogy with the multiplicative group of a finite field. Indeed, in the early 1990s the most common setting for discrete-log-based cryptography was the multiplicative subgroup of order p in $\mathbb{F}_q^\times$, where p was a divisor of $q-1$.

The Elliptic Curve Discrete Log Problem (ECDLP) is the problem, given two elements $P, Q \in \mathcal{G}$, of finding $s \in \mathbb{Z}/p\mathbb{Z}$ such that $Q = sP$. We assume that the ECDLP is intractable for appropriate choices of $E(\mathbb{F}_q)$ (see [10]).

We shall use capital letters to denote elements of $\mathcal{G}$ and lower-case letters to denote elements of the exponent space $\mathbb{Z}/p\mathbb{Z}$. In particular, the letter h will denote values of a hash function $\mathcal{H}$ that maps from arbitrary bitstrings to the exponent space. Although $\mathcal{H}$ is defined deterministically, it should be thought of as producing values in $\mathbb{Z}/p\mathbb{Z}$ that for all practical purposes appear random. The purpose of this section is to give an intuitive picture of the mathematical construction of a cryptocurrency, so we shall not be precise about the desired properties of $\mathcal{H}$.

When a non-identity element $P \in \mathcal{G}$ is fixed, we say that an element $A = sP$ determines s "implicitly." A basic type of interaction between two parties Alice and Bob will take the following form. Alice knows the slope s and y-intercept t of a line in the exponent space, but Bob knows only the values $A = sP$ and $B = tP$ that determine them implicitly. Alice needs to convince Bob that she knows the discrete logs s and t without revealing their values to him.

Here is the protocol that accomplishes that. Bob chooses a random $x \in \mathbb{Z}/p\mathbb{Z}$, which is called his "challenge." Alice must give him the corresponding $y = sx + t$ such that (x, y) is a point on her line. Bob can verify this equation without knowing s or t; he just checks that $yP = xA + B$. An obvious — but crucially important — observation is that if this protocol is carried out twice with the same s and t but different x, then Bob learns s and t, because two points determine a line. This observation is key to Brands' method of avoiding double-spending — more precisely, catching the culprit in the event double-spending occurs. It should be noted that the hardest task in designing an electronic currency scheme is to handle the double-spending problem. Unlike a physical form of cash, which is no simple matter to counterfeit, the same electronic coin can be sent repeatedly to different merchants. Later we shall see how Brands deals with this problem.

When $P_1, P_2 \in \mathcal{G}$ are fixed, by a "representation" of an element $B \in \mathcal{G}$ we mean writing it in the form $B = t_1P_1 + t_2P_2$. An element $B \in \mathcal{G}$ implicitly determines a pair (t_1, t_2), but not uniquely. Rather, the set of possible pairs (t_1, t_2) form a line in the exponent space $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z}$. It is a straightforward exercise to show that the problem of finding a representation of an element is intractable if and only if the ECDLP is intractable.

A simple cryptographic goal, such as sharing a key, can be achieved using only a single generator $P \in \mathcal{G}$; in the Diffie-Hellman protocol [5] the desired key is just $(zu)P$, where z and u are the two users' secret exponents. However, a coin is much more complicated. It must contain elements that implicitly determine enough secret exponents s and exponent pairs (t_1, t_2) so that several authentications can be performed (upon withdrawal, payment, and deposit) and at the same time double-spending is effectively prevented — and all this must be done while preserving anonymity. In Brands' construction this means that we need to use three generators P, P_1, P_2.

The set-up. In what follows we let $\mathcal{B}$ denote the bank, $\mathcal{U}$ denote the user/customer, and $\mathcal{M}$ denote the merchant. We shall describe how $\mathcal{U}$ withdraws a "coin" of a fixed denomination from her bank account, how she pays it to $\mathcal{M}$ as part of a purchase, and how $\mathcal{M}$ deposits the coin in his bank account.

At the very beginning the bank $\mathcal{B}$ randomly chooses three non-identity elements $P, P_1, P_2 \in \mathcal{G}$. The bank also randomly chooses a secret exponent z and publishes $Q = zP$, $Q_1 = zP_1$, $Q_2 = zP_2$. These are its permanent private and public keys.

Meanwhile, the customer $\mathcal{U}$ randomly chooses a secret exponent u_1 and sets her account number equal to $I = I_{\mathcal{U}} = u_1 P_1$. Let $\widetilde{I}$ denote the modified account number $\widetilde{I} = I + P_2$. When opening an account, $\mathcal{U}$ establishes her identity with the bank in the traditional manner (passport or driver's license) and gives the bank I. The bank and the customer also share knowledge of the element $C = z\widetilde{I} = u_1 Q_1 + Q_2$. The first formula (requiring knowledge of z) is how $\mathcal{B}$ computes it, while the second formula (requiring knowledge of u_1) is how $\mathcal{U}$ computes it. This C is similar to a shared Diffie-Hellman key.

Coin signature. For $A, B \in \mathcal{G}$ a signature on the pair (A, B) consists of a triple (C', R, S) of elements of $\mathcal{G}$ along with an exponent y that satisfies both of the following two verification equations:

$$(1) \qquad yP = hQ + R \text{ and } yA = hC' + S, \text{ where } h = \mathcal{H}(A, B, C', R, S).$$

By definition, a coin is a pair (A, B) together with a signature on that pair.

Withdrawal protocol. First the bank $\mathcal{B}$ receives a digitally signed message from $\mathcal{U}$ saying she wishes to withdraw a "coin" in a given denomination. Then $\mathcal{B}$ randomly chooses a secret exponent w and sends $R = wP$ and $S = w\widetilde{I}$ to $\mathcal{U}$.

At this point $\mathcal{U}$ must make several random choices. The customer's steps during withdrawal are the most complicated part of the system. Once we get through that, the rest is simpler and more natural. First $\mathcal{U}$ chooses exponents s, t_1, t_2, and sets $A = s\widetilde{I}$ and $B = t_1 P_1 + t_2 P_2$. Before going farther, let's see what happens if we were to take $s = 1$. After $\mathcal{U}$ sends the bank $h = \mathcal{H}(A, B, C, R, S)$, the bank sends $y = hz + w$ to $\mathcal{U}$. Then $\mathcal{U}$ has a valid coin (A, B, C, R, S, y), since the two equations (1) then become $yP = (hz + w)P$ and $y\widetilde{I} = (hz + w)\widetilde{I}$.

However, we do not want the coin to carry with it the identifying information $\widetilde{I}$ of the person who withdrew it from the bank; for this reason we cannot take $A = \widetilde{I}$ and need to randomize A by setting $A = s\widetilde{I}$. There's a second, subtler problem with the signature in the last paragraph. If the signature is constructed at the time of withdrawal, then the bank can link it to $\mathcal{U}$, and when the coin is eventually deposited the bank will know whose account the particular coin was withdrawn from. The whole point of untraceable electronic cash is to make this impossible.

Thus, something a little different from a signature — but which $\mathcal{U}$ can use to make a signature — is shared with $\mathcal{B}$ during the withdrawal. In addition to s, t_1, t_2 the customer also randomly chooses two exponents u, v. She sets $C' = sC$, $R' = uR + vP$, $S' = suS + vA$, and $r = u^{-1}\mathcal{H}(A, B, C', R', S')$. The purpose of u and v is to "blind" the signature, that is, to create a "quasi-signature" that will enable $\mathcal{U}$ but not $\mathcal{B}$ to form a valid signature from the values that $\mathcal{B}$ and $\mathcal{U}$ exchange.

The rest of the withdrawal proceeds quickly. The customer sends r (but not A or B) to the bank, which computes $y = rz + w$ and sends y to $\mathcal{U}$, who verifies two quasi-signature equations

$$yP = rQ + R \text{ and } y\widetilde{I} = rC + S. \tag{2}$$

If the bank went through the steps properly, these two equations will verify. Conversely, if the equations verify, then the customer need only set $y' = uy + v$ in order to have a coin consisting of the pair (A, B) and the signature (C', R', S') and y'. Let's check the latter claim:

$$y'P = uyP + vP \text{ and } hQ + R' = urQ + uR + vP = uyP + vP$$

by the definition of R' and the first equation in (2); and

$$y'A = (uy + v)s\widetilde{I} = us(rC + S) + vA$$

by the second equation in (2); and

$$hC' + S' = ursC + suS + vA = us(rC + S) + vA$$

by the definition of C' and S'. This shows that $\mathcal{U}$ has a signature, as claimed. Note that the bank does not learn anything about the coin (A, B, C', R', S', y').

Payment protocol. When $\mathcal{U}$ wants to send money to the merchant $\mathcal{M}$, she first sends him the pair (A, B) and its signature (C', R', S') and y'. The merchant verifies the signature.

Let $I' = I_{\mathcal{M}}$ be the merchant's account number, and let d be a bitstring representing the date and time of the transaction. Let $h_0 = \mathcal{H}(A, B, I', d)$, which both $\mathcal{U}$ and $\mathcal{M}$ compute. The hash value h_0 functions as a challenge that $\mathcal{U}$ uses to demonstrate knowledge of her secret exponents u_1, s, t_1, t_2. She does this by computing

$$y_1 = (u_1 s)h_0 + t_1 \tag{3}$$

and

$$y_2 = sh_0 + t_2 \tag{4}$$

and sending (y_1, y_2) to $\mathcal{M}$. Because the point (h_0, y_1) is on the line with slope $u_1 s$ and y-intercept t_1 and the point (h_0, y_2) is on the line with slope s and y-intercept t_2, the merchant can verify the relation

$$y_1 P_1 + y_2 P_2 = h_0 A + B; \tag{5}$$

that is, (3) says that the coefficients of P_1 on both sides of (5) are equal and (4) says the same for the coefficients of P_2. If this equation holds and if the signature on (A, B) verified, then $\mathcal{M}$ accepts the payment.

Notice that the merchant has no need to know the identity of $\mathcal{U}$. For example, if an anonymous remailer such as Tor is used for their communications, $\mathcal{U}$ can buy an e-book or video from $\mathcal{M}$ without $\mathcal{M}$ being able to determine her identity or location.

Deposit protocol. After some time has elapsed, $\mathcal{M}$ wants to deposit the coin in his account. He sends the bank $\mathcal{B}$ the pair (A, B) with its signature (C', R', S') and y', along with y_1, y_2, and the date/time d of the transaction. The bank verifies the coin's signature and, after computing $h_0 = \mathcal{H}(A, B, I_{\mathcal{M}}, d)$, also verifies (5). If (and only if) the relations in (1) and (5) hold, the bank tentatively accepts the deposit, subject only to checking that it has not been spent before. We next describe that last step.

Double-spending. After tentatively accepting the coin, $\mathcal{B}$ searches its database of prior deposits to find out whether (A, B) has been stored as part of an earlier deposit. If not, the bank finalizes the deposit. If the same (A, B) appeared before, then a fraud occurred, and there are two possibilities. First suppose that the challenge h_0 was the same. Then (except with negligible probability) the arguments $A, B, I_{\mathcal{M}}, d$ in $h_0 = \mathcal{H}(A, B, I_{\mathcal{M}}, d)$ are the same in both deposits, and this means that the merchant is trying to deposit the same coin twice.

Now suppose that h_0 was different — that is, the earlier deposit followed a different transaction that took place at a different time — and let h_0', d', y_1', y_2' be the corresponding values from the earlier deposit. In that case the customer $\mathcal{U}$ rather than the merchant $\mathcal{M}$ is trying to spend the same coin twice. The consequence is that the customer loses all anonymity; the bank obtains her identity and conclusive evidence of her fraud. Namely, since both deposits satisfy (3), $\mathcal{B}$ now knows two points (h_0, y_1) and (h_0', y_1') on the line with slope $u_1 s$ and y-intercept t_1; and because of (4) the bank also knows two points (h_0, y_2) and (h_0', y_2') on the line with slope s and y-intercept t_2. From this $\mathcal{B}$ easily computes s and u_1, from which the identity of $\mathcal{U}$ can be computed either as $I = u_1 P_1$ or as $I = s^{-1} A - P_2$.

Security results. Assuming that $E(\mathbb{F}_q)$ has an intractable discrete log problem and $\mathcal{H}$ has good hash function properties, Brands [2, 3] showed that the above construction provides certain security guarantees. We shall not go into detail or be very precise about these results, but rather just give a brief informal list:

- It is infeasible to forge a coin.
- $\mathcal{U}$ can spend a coin if and only if she knows representations of A and B.
- If $\mathcal{U}$ follows the protocols and does not double-spend, then $\mathcal{B}$ cannot compute a proof of double-spending.
- An eavesdropper who listens in on the withdrawal and payment protocols cannot obtain a coin that he can deposit in his account.

Criticisms. Although e-cash systems such as the one described above are an elegant solution to the problem of secure and anonymous payments, several objections — both technical and social — have been raised. In the first place, the untraceability makes the job of law enforcement more difficult. Police and national security agencies can no longer follow the money trail. Money-laundering, tax evasion, terrorist activities, and purchase of child pornography all become harder to stop. In 1996 an article [12] by three NSA researchers explained the mathematics of e-cash systems and concluded with some comments about the problem they pose for law enforcement and a recommendation to implement such a system only with a key-escrow feature:

> The untraceability property of electronic cash creates problems in detecting money-laundering and tax evasion because there is no way to link the payer and payee. To counter this problem, it is possible to design a system that has an option to restore traceability using an escrow mechanism. If certain conditions are met (such as a court order), a deposit or withdrawal record can be turned over to a commonly trusted entity holding a key that can decrypt information connecting the deposit to a withdrawal or vice versa. This will identify the payer or payee in a particular transaction.

A second criticism is that coins are not transferable, at least not in the system described above. This means that the merchant who receives a coin from a buyer cannot use it to pay his supplier or his employees; he can only deposit the coin in his bank account and then, if he wants, withdraw a coin in a separate action. In other words, the bank has to step in between each payer/payee in a sequence of money transfers. If one wants to avoid this, an e-cash system would have to allow a large accumulation of data on a coin in order to handle the double-spending problem. Double-spending would not be detected until the coin is finally deposited, and then the bank must have a way to identity the user in the chain of transfers who was guilty of spending it twice. This is complicated, and no efficient way is known to do this.

A third criticism is that the bank that controls the administration of the system can easily impose substantial fees for each transaction. The most common objection to our current system of credit card payments and wire transfer of funds is not the lack of privacy (although certainly many people are bothered by that), but rather the steep fees that cut into merchants' profits and cause them to raise their prices. Those fees also prevent the poorest sections of the world's population from being able to use the banking system, and they cause a great burden on immigrant workers in the wealthy countries who wish to wire remittances to their family back home. These issues are not addressed by an e-cash system that remains under the control of the banks, at least not if the fees are similar to those charged for credit card transactions.

It should be noted that the systems of Chaum, Brands, and others never took off commercially. There were many reasons for that, but perhaps the most important reason was that the banks did not believe that it was in their interest to promote e-cash. Even though they could charge high fees, they must have anticipated that there would be push-back from customers who might object to paying substantial fees for something that was called "cash." The banks would feel much more pressure to keep fees low than in the case of credit cards. Thus, they would have no interest in supporting a system that would compete with credit cards and would result in less profit.

Another likely explanation for the banks' unenthusiastic response to e-cash was that the anonymity feature meant that banks and credit card companies could not gather valuable data about customer spending that they could then sell for targeted advertising.

A different type of objection to the e-cash systems of the 1990s was that they relied on a central authority — a government-regulated bank — that was assumed to be trustworthy. Many civil libertarians, cypherpunks, and others did not believe that one should put such confidence in a central bank or government. What if a powerful government that has a dismal record on human rights were to pressure the bank into barring e-coin deposits from an organization that was exposing the misdeeds of that government?

In 2010 the U.S. government unwittingly provided a strong argument in favor of the cypherpunk viewpoint when it blatantly violated the political neutrality of the banking system. As described in a column in *Forbes* online [13],

> Following a massive release of secret U.S. diplomatic cables in November 2010, donations to WikiLeaks were blocked by Bank of America, VISA, MasterCard, PayPal and Western Union on December 7th, 2010.... It was coordinated pressure exerted in a politicized climate by the U.S. government...

The U.S. government perceived WikiLeaks — which had earlier released secret files and videos documenting U.S. atrocities in Iraq and Afghanistan, including the so-called "collateral murder" video — as a threat to its national interests. In contrast, many human rights advocates saw WikiLeaks as a valuable democratic institution. Supporters of WikiLeaks responded to the U.S. government's politicization of the banking system by turning to a new type of currency, called Bitcoin, that was completely out of the control of any central authority.

Remark 1. The WikiLeaks episode is not the only example of U.S. government interference with the use of dollars for legitimate purposes. For example, during certain time periods before U.S.–Cuban relations started to normalize, the Cuban Interests Section in Washington, D.C. was unable to open bank accounts because of banks' reluctance to risk inadvertent violation of U.S. sanctions against countries that were on the U.S. State Department's list of "State Sponsors of Terrorism;" see [1]. This meant, for example, that Cubans living in the U.S. whose Cuban passports had expired were unable to get new passports so that they could visit family in Cuba.

Among U.S. states, about half have legalized marijuana for medical purposes, and four (Colorado, Oregon, Washington, and Alaska) have legalized it for other uses as well. However, stores that sell marijuana have had great difficulty opening bank accounts (and as a result some have been repeatedly burglarized by thieves who know that they do a large cash business) because the central government in Washington D.C. still lists marijuana as an illegal drug, and banks fear that opening an account for a marijuana business could run afoul of money-laundering laws (see [11]).

In both cases Bitcoin could have been a solution, although it would have been only a partial solution because use and acceptance of Bitcoin are still on a relatively small scale.

3. Peer-to-Peer Cryptocurrencies

The most important difference between a peer-to-peer cryptocurrency and the types of e-cash considered in the last section is that double-spending is prevented through a consensus mechanism of the network itself without any need to resort to a trusted authority such as a bank. Another consequence of shunning all banking and government authorities is that a new type of money is created that is independent of the dollar, euro, and all other fiat currencies. Several hundred peer-to-peer cryptocurrency systems have been created, including many that serve a particular niche or have a specialized purpose. We shall limit our discussion to Bitcoin, which is by far the most important and successful one, and we shall slightly simplify some of the details so as to focus on the essential elements. Bitcoin was created in 2008 by the pseudonymous Satoshi Nakamoto, and it has a market capitalization of several billion U.S. dollars.

Mathematical ingredients. The two main mathematical ingredients in Bitcoin are hash functions and the Elliptic Curve Digital Signature Algorithm (ECDSA) [7]. We shall again use $\mathcal{H}$ to denote our hash function, which takes an arbitrarily long string of bits and produces a "fingerprint" of a fixed bitlength — in Bitcoin it is 256. We assume that $\mathcal{H}$ has certain properties that make hash values appear random. Namely, its values are uniformly distributed over all possible strings of 256 bits; it's infeasible to find two bitstrings with the same hash value; and knowing the hash values of related bitstrings won't give you any advance information about the hash value of the bitstring you're interested in. The hash function used in Bitcoin is SHA256 [6], which is believed to satisfy all the desired properties.

At the heart of how Bitcoin functions is the following computational task: Given a bitstring u and a bound B, search for an integer ℓ (called a "nonce") such that $\mathcal{H}(u\|\ell) < B$. Here $u\|\ell$ denotes the concatenation of u with the binary representation of ℓ, and the values of $\mathcal{H}$ are regarded as integers between 0 and $2^{256}-1$. For example, if $B = 2^{256-j}$, then one needs to find ℓ such that the hash value of $u\|\ell$ starts with j zero-bits.

The second basic ingredient in Bitcoin is the ECDSA, which is used to authorize payments. Suppose that a user $\mathcal{U}$ wishes to sign a message M that authorizes a certain payment to a merchant $\mathcal{M}$. We describe how the signature is constructed and how it is verified. As before let $\mathcal{G}$ be a subgroup of prime order p of the group of points on an elliptic curve defined over a finite field, where we suppose that it is computationally infeasible to find discrete logarithms in $\mathcal{G}$. Let $P \in \mathcal{G}$ be a fixed basepoint. The private key of the user $\mathcal{U}$ is a random integer z mod p, and her public key is the point $Q = zP$. Let $\mathcal{H}'$ be a hash function whose values are integers mod p; for example, we could define $\mathcal{H}'(x) = \mathcal{H}(x) \bmod p$, where $\mathcal{H}$ is SHA256. To sign a message M, $\mathcal{U}$ first randomly selects an ephemeral secret integer k mod p that must be different for each message she signs. She computes kP and lets r denote the x-coordinate of kP, regarded mod p. She computes $s = k^{-1}(\mathcal{H}'(M) + zr) \bmod p$; her signature is (r, s). Anyone can verify this signature by checking that the x-coordinate of $s^{-1}\mathcal{H}'(M)P + s^{-1}rQ$ is congruent to r mod p.

A user's identity on the Bitcoin network is her ECDSA public key,[1] since that is all that other users need to know about her. If she wants to minimize the likelihood that her real-world identity will be linked to her purchases, she should use a different ECDSA public/private key pair for each transaction.

Transactions. Money gets passed along from one user to another through a sequence of transactions. Suppose, for example, that $\mathcal{U}$ wishes to buy something from a merchant $\mathcal{M}$ that costs half a bitcoin, abbreviated 0.5 BTC. She forms an "input" to the transaction consisting of one or more earlier transactions $T_1, T_2, \ldots, T_m$ in which she was a payee such that the amounts $b_1, b_2, \ldots, b_m$ she was paid cover the cost: $b_{\text{total}} = \sum b_i \geq 0.5$ BTC. For instance, suppose that $b_{\text{total}} = 0.75$ BTC. For simplicity let's suppose that she used the same ECDSA key pair for all of those transactions. Here T_i is the identifying information for the earlier transaction, essentially the hash value of the transaction data.

Next, $\mathcal{U}$ forms the "output" of the transaction, which specifies that 0.5 BTC goes to $\mathcal{M}$ and 0.2495 BTC gets returned to $\mathcal{U}$ (her "change"), where both $\mathcal{U}$ and $\mathcal{M}$ are identified simply by their ECDSA public keys. This

[1] A Bitcoin address is derived from the public key but is not identical to it, since it is convenient to shorten the address by hashing. However, we shall disregard such features of Bitcoin in the interest of simplicity.

accounts for all but 0.0005 BTC, which is the transaction fee. (The Bitcoin transaction fee is not required, but is a good idea in order to incentivize miners to include the transaction in their block. In any case, the fee is miniscule compared to fees for credit cards, wire transfers, and other bank services. The appropriate fee in Bitcoin typically depends not on the amount of money in the transaction, but rather on the number of bytes in the transaction text.) Finally, $\mathcal{U}$ signs the message giving the input followed by the output:

$$c_1, T_1, \ldots, c_m, T_m, 0.5, \mathcal{M}, 0.2495, \mathcal{U},$$

where c_i is the index of the output in T_i for which $\mathcal{U}$ was the payee. That message along with her signature is the new transaction, which she then sends out to the Bitcoin network. Each node in the network checks the validity of the transaction — that it is properly formed and draws upon unspent earlier transactions, and that $\mathcal{U}$'s signature verifies using her ECDSA public key — before sending it on through the network.

Forming the blockchain. The most important contribution of Nakamoto's seminal paper [15] was to propose a method of establishing a chronological sequence of transactions that all users can agree on — without, of course, relying on a trusted authority. The idea was to organize transactions into blocks, each of which, through extensive computation, gets confirmed and then placed immediately after the most recently confirmed block. The same transaction cannot be spent twice in a block, and it cannot be included as an input in a transaction in a later block if it already was spent in a block that was confirmed earlier. A block, typically containing several hundred transactions, gets confirmed and added to the Bitcoin blockchain roughly every 10 minutes.

The very first bitcoins were created by Satoshi Nakamoto on 3 January 2009. This was done by means of a 50 BTC transaction with payee represented by Nakamoto's public key (and no payer). This transaction comprises the first block, called the "genesis" block, which was also created by Nakamoto. The first transaction and the genesis block are embedded in the Bitcoin software.

Subsequently, the organization of blocks and the computations to confirm them are carried out by "miners" (a term meant as an homage to the gold miners of times past). When a miner forms a block, the very first transaction he puts in it (called a "coinbase transaction") is a special one that gives him a reward (25 BTC as of 1 September 2015[2] plus the fees for all non-coinbase transactions in that block) in the event his block is confirmed. This is the only way that new bitcoins get added to the system.

The miner then gathers together a large number of the transactions that have been sent out over the Bitcoin network but have not yet been included

[2] The reward will be halved every 210,000 blocks until the year 2140, when the total number of bitcoins will reach 21 million; after that, the only incentive to miners will be the transaction fees.

in a confirmed block. Since his incentive also includes the transaction fees, he might pass over a very lengthy transaction that includes only the minimal fee. But it's in his interest to include almost all the transactions he can find. Once he's put together his block, he computes the hash values of the transactions and the Merkle hash tree (see below). He then forms a block header containing two essential pieces of information — its Merkle root and the hash of the most recent block that this block is supposed to follow. At the end of the header is a nonce ℓ which he is free to increment. That is, the header has the form $H\|\ell$.

The miner then computes $\mathcal{H}(H\|\ell)$, hoping that its value is less than B. The odds against this are long, namely, $2^{256}/B$ to one. So he keeps incrementing ℓ and hashing $H\|\ell$, hoping that the next ℓ-value is a lucky one. (It should be noted that the Bitcoin software automatically adjusts the difficulty level, which depends on the bound B, roughly once every two weeks; the change in B depends on the average length of time it took to confirm a block, taken over the previous 2016 blocks.)

Meanwhile, other miners are carrying out the same procedure. Their blocks are formed somewhat differently, with not necessarily the same transactions (although most likely a large overlap). Even if another miner has exactly the same set of transactions in her block, the Merkle root will be different because she won't have exactly the same order of transactions. The first miner to achieve the inequality $\mathcal{H}(H\|\ell) < B$ sends his block to the network to be added to the blockchain. The value of ℓ such that $\mathcal{H}(H\|\ell) < B$ is called a "proof of work." The "full nodes" (as opposed to "light nodes," see below) — and this includes the other miners — carefully check the validity of all of the transactions in the proposed next block, and of course they also check that the inequality $\mathcal{H}(H\|\ell) < B$ holds. Once this is done, most likely the miner who broadcast the block will soon get his 25 BTC plus transaction fees. However, he can't celebrate yet — in the words of Arizona State University football fans,[3] he should "fear the fork."

Blockchain forks. A fork in the blockchain occurs when two miners obtain a proof of work — that is, find values $H_1\|\ell_1$ and $H_2\|\ell_2$ such that $\mathcal{H}(H_i\|\ell_i) < B$, $i = 1, 2$ — at almost the same time. When other miners get a block with the required proof of work, they immediately stop working on their current block and start forming a new block to follow the one they just received. If they receive two such confirmed blocks in rapid succession, they choose the one they received first and proceed to construct a new block to follow it. (Recall that their new block header will contain the hash value of the block that it's supposed to directly follow.) They'll keep the second block in reserve, just in case it turns out to be the first one to have a confirmed block follow it. Of the two blocks that arrive almost simultaneously, the "winner" — that is, the one that is first followed by an "offspring" — largely depends

[3] http://www.collegefootballstore.com/CFS_Arizona_State_Sun_Devils_T-Shirts

on which has the most miners accept it as the first to arrive, except that "the most miners" is understood in the sense of computing power. In other words, the principle is: one CPU, one vote.

Before long, one branch of the fork will be shorter than the other. Since the longer branch is accepted as the main branch by almost all the miners, after it becomes more than two or three blocks longer — to be extra-sure, it's recommended to wait until 6 blocks have been added, that is, for one hour — it is virtually impossible for the shorter one to catch up. As miners reach a consensus on this, the transactions in the secondary chain get put back into the pool so as to be included in blocks intended for the main chain (and note that only the miners with confirmed blocks on the main chain get rewards). It should take no more than a few hours for all this to be sorted out and for any given transaction to be included in a confirmed block.

In the seven years that Bitcoin has existed, there has been only one time when a major fork occurred that lasted several blocks (see [18]). This was in March 2013, and it was caused by a software update that introduced an incompatibility between two versions. Users with version 0.7 were accepting a different block than those using version 0.8. This was fixed in roughly a day, and no lasting damage was done. As far as anyone knows, the temporary breakdown in the one-main-branch system did not permit any double-spending or other abuses.

The choice of 10 minutes for a block to be confirmed was not arbitrary. Rather, the idea was that if blocks were confirmed more rapidly — that is, if the difficulty level for confirmation were reduced — forks would occur too frequently. On the other hand, if the time to confirm a block were increased substantially, then it would take an inconveniently long time to get confirmation that a transaction is valid.

For small purchases, the merchant $\mathcal{M}$ would probably accept a payment in bitcoin right away, without waiting for the confirmation, because the risk of loss is small. However, before a large payment is accepted as valid (with no risk of double-spending) it's strongly recommended to wait until about 6 blocks of the Bitcoin blockchain have been confirmed.

Merkle trees and light nodes. We now describe a technique due to Ralph Merkle [14] for greatly increasing the efficiency of managing data in a network such as Bitcoin. Merkle hash trees make it possible for a user who does not want to download the entire Bitcoin blockchain (over 40 gigabytes as of 1 September 2015) to be certain that a given transaction really is included in the block it's claimed to be in. Such a user with limited resources is called a "light node."

For simplicity suppose that a given block has 512 transactions (or, more generally, the number of transactions in the block is a power of 2). Let the transactions be indexed by the set of all 9-bit strings, and let T_i for $i \in \{0,1\}^9$ be the hash value of the corresponding transaction. We visualize them as 512 "leaves" of an (upside-down) tree that are lined up in order on

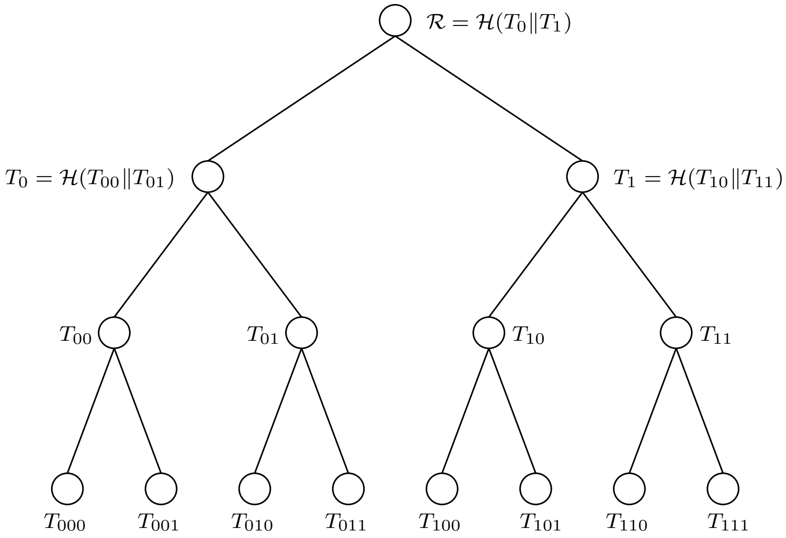
$\mathcal{R} = \mathcal{H}(T_0 \| T_1)$
$T_0 = \mathcal{H}(T_{00} \| T_{01})$
$T_1 = \mathcal{H}(T_{10} \| T_{11})$
T_{00}
T_{01}
T_{10}
T_{11}
T_{000}
T_{001}
T_{010}
T_{011}
T_{100}
T_{101}
T_{110}
T_{111}

Figure 1. Merkle tree with 8 transactions.

the bottom row. Moving one row higher, for any $i \in \{0,1\}^8$ we define T_i to be the hash value of $T_{i'} \| T_{i''}$, where $i' = i\|0$ and $i'' = i\|1$, and we continue inductively. Finally, we get to the hash values T_0 and T_1 corresponding to a single bit; and we define the "Merkle root" of the block to be the hash value of $T_0 \| T_1$. Figure 1 shows the structure of the Merkle tree when there are 8 transactions. The lines going down from T_i to $T_{i\|0}$ and $T_{i\|1}$ represent the relation $T_i = \mathcal{H}(T_{i\|0} \| T_{i\|1})$.

If you're a light node, all you have is the hash value of the transaction in question and the header of the block that supposedly contains it; that header contains the Merkle root of the block but little else. In order to verify that the transaction really is there, you enlist the help of one of the full nodes, who supplies you with 9 hash values. The full node searches through the block and finds the transaction; suppose it has index 011010001. Then the full node sends you the following 9 hash values: $T_{011010000}$, $T_{01101001}$, $T_{0110101}$, T_{011011}, T_{01100}, T_{0111}, T_{010}, T_{00}, and T_1. (In going from one to the next, we drop the last bit and switch the second-to-last bit.) You then hash $T_{011010000}$ concatenated with the hash value of your transaction (that is, with $T_{011010001}$); you next hash this value, which supposedly is $T_{01101000}$, with $T_{01101001}$; then hash the result (which supposedly is $T_{0110100}$) concatenated with $T_{0110101}$, and so on. You should end up with the Merkle root. At this point you are convinced that the only way you could have gotten the Merkle root is if the hash value of the transaction in question really belonged in the 011010001 position in the block.

A convenient feature of the Merkle tree is that one can drop from the blockchain record most of the transactions that were spent long ago. As soon as both leaves whose indices share all but the last bit are spent, they can be deleted from the record, since they'll no longer be needed to verify any hashes. Eventually one can delete the sets of transactions corresponding to whole branches of a Merkle tree; not even full nodes will need to keep their records.

Pros and cons. There are three main appeals of Bitcoin:

- To civil libertarians, it frees people from abuse of authority by governments and banks (as occurred, for example, when the U.S. government blocked the use of most credit cards for donations to WikiLeaks).

- To rank-and-file users — and especially the poor — it liberates them from onerous banking fees. For example, immigrant workers who want to send remittances home to their family can do so in Bitcoin at negligible cost. Bitcoin makes possible small payments by cellphone, and makes some of the conveniences enjoyed by affluent people available to the "unbanked" (the large proportion of the world's population that has no relation with any bank).

- To merchants there are two major attractions of Bitcoin. In the first place, the transaction fees are very small, and in any case are paid by the buyer, not by the merchant. With credit cards, the fees borne by the merchants are burdensome, especially to small shopowners, and cause them to raise prices. In the second place, transactions are irreversible — just as with cash, there is no way to cancel payment later. One of the great banes of merchants is that a customer who paid by credit card can easily dispute and reverse the charge (this is called a "chargeback") with little need for convincing justification.

On the other hand, there are also many objections to Bitcoin:

- The exchange value of a bitcoin has fluctuated wildly. In U.S. dollars its lowest value was $0.0025 on 22 May 2010, and its highest value was $1126.82 on 30 November 2013. In the course of 2014 its value, which started out at $747.56, fell by a factor of three. On 1 September 2015 a bitcoin was worth $230.76. Reasons for the volatility include the relatively small volume of users and transactions (compared to the dollar), the influence of speculators, and the effects of rumors and news items. If an important political authority announces plans to restrict or regulate Bitcoin transactions, confidence in the cryptocurrency might take a hit. If a major retailer announces plans to accept Bitcoin, its value could suddenly rise. Strangely, BitPay's sponsorship of the Bitcoin St. Petersburg Bowl (a U.S. college football event) on 26 December 2014 did not seem to appreciably help the exchange value of a bitcoin.

- Even though the Bitcoin network itself seems to have been well designed, in its relations with the rest of the financial world it has to rely on enterprises and organizations that are outside the network and may turn out to be unreliable. For example, in 2013 the largest exchange company was Mt. Gox.

Based in Tokyo and headed by a Frenchman named Mark Karpelès, who ran it poorly [16], the company lost 744,408 customer bitcoins (worth about $400 million at the time) shortly before declaring bankruptcy in February 2014.

• Anonymity and untraceability of currency transactions — basic features of cash that are achieved by the e-cash described in the last section — are not the main objectives of Bitcoin, and they are not guaranteed. Although nothing in the blockchain record directly indicates a user's real-world identity, the blockchain is a public ledger giving the history of all transactions, and from all that information it might be possible to deduce a user's actual identity. Anyone who wants to be completely anonymous must hope that all of her business associates (that is, users whom she pays or who pay her) similarly want to keep their real-world identity private. Otherwise, an investigator can examine her transactions to determine whom she has done business with. Knowing someone's friends or business associates is often enough to make an educated guess about the person's identity.

• The proof of work — finding quadrillions of hash values — wastes vast amounts of electricity and computer resources that might otherwise be used for projects that have social or scientific value.

• The original idea was that Bitcoin mining would be democratic, with any user free to put her PC to work looking for a nonce ℓ such that $\mathcal{H}(H\|\ell) < B$. However, once it became clear that there was real money to be earned by extremely rapid computation of hash values, professionals moved in and largely took over, especially after the development and mass production of ASICs (application-specific integrated circuits) for computing SHA256 values. As of 1 September 2015 the many Bitcoin mining ASICs that are at work at any time altogether can run approximately 2^{59} (that is, about half a billion billion) hash computations per second. This means that the difficulty level is set at about 2^{68} (in other words, $B \approx 2^{188}$). Meanwhile, a PC can compute about 2^{23} hash values per second. So in any 10-minute period a PC user has only about a 2^{-36} (one in sixty billion) chance of being the first to find a lucky nonce. Because of this, many users have formed mining pools, where they group their computing resources and share any rewards. There are a small number of very large pools, and if several of them decided to combine forces for fraudulent purposes, they could possibly undermine the integrity of the block chain.

• Police and national security agencies have the same objections as to earlier versions of e-cash (despite the imperfect anonymity mentioned above). One of the most active users of Bitcoin was a website called Silk Road, which specialized in Internet sale of illegal drugs. In October 2013 it was shut down by U.S. authorities, and its alleged mastermind, Ross Ulbricht, was arrested. In addition, an executive of the Bitcoin Foundation named Charlie Shrem was arrested and convicted of money-laundering; in December 2014 he was sentenced to two years in prison.

4. Cryptocontracts

The idea of a peer-to-peer system of contracts that are enforced cryptographically rather than through any government authority goes back at least 20 years (see [17]). But it was largely the success of Bitcoin that provided the impetus for putting the idea into practice. We shall discuss the most important project to do this, called Ethereum [8, 9]. The main designer of Ethereum is Vitalik Buterin, a co-founder of *Bitcoin Magazine*.

Ethereum has its own currency, called "ether" and denoted ETH, that is modeled closely on Bitcoin. However, the currency plays a secondary role and is not an end in itself. Rather, Ethereum provides a platform and a programming language that has enough expressive flexibility to describe any cryptographically enforceable contract (the technical term is "Turing complete"). The currency serves merely as the grease that lubricates the enforcement mechanism. A transaction that includes a piece of code must include a fee that is proportional to the number of computer instructions that are executed when the code is run. (Among other things, this prevents code that has infinite loops.) In addition, if two users have Ethereum wallets, then the contract can be enforced (i.e., money can be moved) without needing to access accounts in other currency systems (such as USD or BTC).

We first give a simple example and then a somewhat more complicated example of how Ethereum contracts might work.

Example 1. Suppose that two users $\mathcal{U}$ and $\mathcal{V}$ living in different parts of the world wish to bet 1000 ethers on the outcome of the World Cup. They each digitally sign an Ethereum smart contract and make 1000 ETH available to be frozen under the terms of the contract. While waiting for the championship game, they do not have access to those funds. As soon as the game is over, the smart contract consults several websites to determine which team won and then automatically transfers the frozen money to the winner of the bet.

Example 2. Suppose that $\mathcal{U}$ agrees to buy a house from $\mathcal{V}$, provided that he makes all the repairs and replacements that are recommended by an engineering company $\mathcal{E}$. The buyer $\mathcal{U}$ agrees to pay 10000 ETH earnest money and later an additional 2000 BTC (so that the purchase price is 10000 ETH + 2000 BTC). Before a certain date d_1 she will pay 5 BTC to $\mathcal{E}$, who will perform the engineering inspection and send $\mathcal{U}$ and $\mathcal{V}$ a report by date d_2, after which the seller $\mathcal{V}$ has until the closing date d_3 to make the recommended repairs to the satisfaction of $\mathcal{E}$. The three parties to the agreement, $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{E}$, digitally sign the Ethereum smart contract, at which point 10000 ETH from the buyer's account are automatically frozen in escrow. On d_1 the smart contract goes into the Bitcoin network and verifies that a confirmed 5 BTC payment from $\mathcal{U}$ to $\mathcal{E}$ has been made. On d_2 it verifies a signed message from $\mathcal{E}$ saying that a report has been sent to $\mathcal{V}$, and on d_3 it verifies a signed message from $\mathcal{E}$ saying that $\mathcal{V}$ has made all necessary repairs. If all of the verifications go through — or if the first verification fails (meaning that $\mathcal{U}$ did not pay for the engineering inspection) — then the smart contract pays the frozen 10000 ETH to $\mathcal{V}$, who receives the earnest money whether or not $\mathcal{U}$ ultimately sends the 2000 BTC payment to purchase the house. If, however, $\mathcal{V}$ did not cooperate with $\mathcal{E}$ (either by not making it possible for the engineering inspection to be carried out before d_2 or by not making all of the recommended repairs before d_3), then the smart contract unfreezes the 10000 ETH, that is, returns it to $\mathcal{U}$.

Some differences with Bitcoin. Unlike Bitcoin, the amount of Ethereum currency is not capped at a fixed upper bound, and so it is not deflationary. New ether will be issued at the constant rate of approximately 1.8×10^7 ETH/year.

In 2014 there was a pre-sale of ether that collected a total of 31,531 BTC (worth US$ 18,439,086 at the time of sale) in exchange for 60,102,216 ETH. The ether bought in the pre-sale is not usable or transferable until the launch of the genesis block, which is expected to happen shortly.

Controversies. The Ethereum designers are trying to avoid some of the drawbacks that have been identified in Bitcoin. For example, two solutions have been proposed to the problem of the waste of electricity and computational resources caused by the vast number of hash computations whose only purpose is to satisfy an arbitrary mathematical inequality:

- The hash computations could be replaced by a socially or scientifically useful type of computation. Preferably, the computations would still have the property that the results of extensive work could be verified very quickly, providing an efficient "proof of work." For example, the Sloan Digital Sky Survey[4] involves a systematic search for certain types of astronomically interesting objects. Some of them are like "needles in a haystack" — taking much computation to find, but easy to verify once found.

- The concept of proof of work could be replaced by a broader notion called "proof of stake." For example, users of the network could be required to maintain a certain level of ETH deposits. Instead of "one CPU, one vote," the principle could be "one ETH invested, one vote." Someone who seriously violates the rules (for example, trying to get two blocks confirmed so as to double-spend a transaction) would be required to forfeit his deposit.

Both of these modifications would make the confirmation protocol more complicated than in the current Bitcoin system, and the details require a lot of careful preparation in order to avoid possible attacks.

To some extent cryptocontracts will face the same opposition as Bitcoin has. Law enforcement and national security agencies, for example, would point out that Ethereum can potentially facilitate illegal gambling, and,

[4] http://www.sdss.org

more generally, the possibilities for cross-border criminal consortia and terrorist networks will be greater because illegitimate enterprises would have the same access to the technology as respectable businesses.

There is one group of people who might feel threatened by cryptocontracts, though not by cryptocurrencies. Those are the lawyers, many of whom would be put out of business if Ethereum contracts came into widespread use. The legal profession might oppose Ethereum as a type of unlicensed lawyering — much as the medical profession (at least in the U.S.) has strenuously opposed healers who operate outside the established allopathic institutions.

References

[1] Adams D.: Analysis: U.S. sanctions make Cuba's bank account too toxic for banks (29 November 2013). http://www.reuters.com/article/2013/11/29/us-cuba-usa-banking-analysis-idUSBRE9AS0QE20131129.

[2] Brands S.: Untraceable off-line cash in wallets with observers. In: Advances in Cryptology—Crypto '93. LNCS, vol. 773, pp. 302-318, Springer, Berlin (1994).

[3] Brands S.: An efficient off-line electronic cash system based on the representation problem. Technical Report CS-R9323, CWI (1993).

[4] Chaum D.: Blind signatures for untraceable payments. In: Advances in Cryptolog —Crypto '82. pp. 199-203, Plenum Press (1983).

[5] Diffie W., Hellman M.: New directions in cryptography. IEEE Trans. Inf. Theory. **IT-22**, 644-654 (1976).

[6] FIPS 180-3: Secure Hash Standard (SHS), Federal Information Processing Standards Publication 180-3, National Institute of Standards and Technology (2008).

[7] FIPS 186-4: Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4, National Institute of Standards and Technology (2013).

[8] http://ethereum.org

[9] A next-generation smart contract and decentralized application platform (2015). http://github.com/ethereum/wiki/wiki/White-Paper

[10] Galbraith S., Gaudry P.: Recent progress on the elliptic curve discrete logarithm problem. Des. Codes Cryptogr. (to appear).

[11] Kiley B.: What marijuana businesses are doing with their stacks of cash (22 October 2014). http://www.thestranger.com/seattle/what-marijuana-businesses-are-doing-with-their-stacks-of-cash/Content?oid=20884534.

[12] Law L., Sabett S., Solinas J.: How to make a mint: The cryptography of anonymous electronic cash. The American University Law Review. **46**, 1131-1162 (1996).

[13] Matonis J.: WikiLeaks bypasses financial blockade with Bitcoin (8 August 2012). http://onforb.es/NATDQt.

[14] Merkle R.C.: Protocols for public key cryptosystems. In: Proc. Symp. Security and Privacy, pp. 122-133. IEEE (1980).

[15] Nakamoto S.: Bitcoin: A peer-to-peer electronic cash system (2008). https://bitcoin.org/bitcoin.pdf

[16] Pagliery J.: Bitcoin and the Future of Money. Triumph Books (2014).

[17] Szabo N.: Formalizing and securing relationships on public networks. First Monday. **2** (9) (1997).

[18] Taylor D.: Now that it's over: The blockchain fork explained for regular users (12 March 2013). http://www.reddit.com/comments/1a51xx/.

Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 U.S.A.
E-mail address: koblitz@uw.edu

Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada
E-mail address: ajmeneze@uwaterloo.ca