A Fistful of Bitcoins

Characterizing Payments Among Men with No Names

SARAH MEIKLEJOHN, MARJORI POMAROLE, GRANT JORDAN, KIRILL LEVCHENKO, DAMON MCCOY, GEOFFREY M. VOELKER AND STEFAN SAVAGE

Sarah Meiklejohn is a PhD candidate in computer science and engineering at the University of California, San Diego. She previously received an ScM in computer science and an ScB in mathematics from Brown University. At UCSD, she is co-advised by Mihir Bellare and Stefan Savage, and has broad research interests in cryptography and security. smeiklej@cs.ucsd.edu

Marjori Pomarole is an undergraduate at the University of California, San Diego, studying computer science. She has interned at Google and Facebook, working with infrastructure monitoring. marjoripomarole@gmail.com

Grant Jordan is a graduate student at the University of California, San Diego, focusing on computer security. His previous work has included research in online spam distribution and botnets, as well as UAV development for the Air Force Research Lab. gejordan@cs.ucsd.edu

Kirill Levchenko is a research scientist at the University of California, San Diego. His research is focused on computer networking and security. klevchen@cs.ucsd.edu

Damon McCoy is an assistant professor in the CS department at George Mason University. He obtained his PhD from the University of Colorado, Boulder, and his research includes work on anonymous communication systems, cyber-physical security, e-crime, and wireless privacy. damon.mccoy@gmail.com

Geoffrey M. Voelker is a Professor of Computer Science at the University of California, San Diego. He works in computer systems, networking, and security. voelker@cs.ucsd.edu

Stefan Savage is a professor of computer science and engineering at the University of California, San Diego. He received his PhD in computer science and engineering from the University of Washington and a BS in applied history from Carnegie Mellon University. Savage is a Sloan Fellow and an ACM Fellow, but is a fairly down-to-earth guy and only writes about himself in the third person when asked. savage@cs.ucsd.edu

Bitcoin is a decentralized virtual currency whose usage has skyrocketed since its introduction in January 2009. Like cash, the ownership of bitcoins is anonymous, as participants transact bitcoins using pseudonyms rather than persistent real-world identities. In this article, we examine the limitations of Bitcoin anonymity and discover that the ability to cluster pseudonyms according to heuristics about shared ownership allows us to identify (i.e., associate with a real-world entity or user) a significant and active slice of the Bitcoin economy. Along the way, we explain a lot about how Bitcoin works.

Bitcoin is a form of electronic cash that was introduced by Satoshi Nakamoto (a pseudonym) in 2008. As its name suggests, Bitcoin is similar to cash in that transactions are irreversible and participants in transactions are not explicitly identified: both the sender(s) and receiver(s) are identified solely by pseudonym, and participants in the system can use many different pseudonyms without incurring any meaningful cost. Bitcoin has two other properties, however, that make it unlike cash: (1) it is completely decentralized, meaning a global peer-to-peer network, rather than a single central entity, acts to regulate and generate bitcoins, and (2) it provides a public transaction ledger, so that although transactions operate between pseudonyms rather than explicit real-world individuals, every such transaction is globally visible.

Since its introduction, Bitcoin has attracted increasing amounts of attention, from both the media and from governments seeking ways to regulate Bitcoin. In large part, much of this attention has been due to either the nature of Bitcoin, which has caused government organizations to express concern that it might enable money laundering or other criminal activity, or to its volatility and ultimate growth as a currency; in late 2012 the exchange rate began an exponential climb, ultimately peaking at $235 US per bitcoin in April 2013, before settling to approximately $100 US per bitcoin (as of September 2013).

In spite of the concerns about Bitcoin, its use of pseudonyms has made gaining any real understanding of how and for what purposes Bitcoin is used a fairly difficult task, as the abstract Bitcoin protocol—if exploited to its fullest extent—provides a fairly robust notion of anonymity. Nevertheless, in modern Bitcoin usage, many users rely on third-party services

to store their bitcoins, such as exchanges and wallet services (i.e., banks), rather than individual desktop clients that they operate themselves. In this context, our goal is to exploit this behavior to erode the anonymity of the users that interact with these and other services. In doing so, we do not seek to de-anonymize individual users, but rather to de-anonymize *flows* of bitcoins throughout the network.

Our approach consists of two techniques. First, we engage in a variety of Bitcoin transactions to gain ground-truth data; for example, by depositing bitcoins into an account at the biggest Bitcoin exchange, Mt. Gox, we are able to tag one address as definitively belonging to that service, and by later withdrawing those bitcoins we are able to identify another. To expand on this minimal ground-truth data, we next *cluster* Bitcoin addresses according to two heuristics: one exploits an inherent property of the Bitcoin protocol, and another exploits a current idiom of use in the Bitcoin network. By layering this clustering analysis on top of our ground-truth data collection, we transitively taint entire clusters of addresses as belonging to certain users and services; for example, if our analysis indicated that the address we had previously tagged as belonging to Mt. Gox was contained in a certain cluster, we could confidently tag all of the addresses in that cluster as belonging to Mt. Gox as well.

How Bitcoin Works

Before describing our analysis, gaining an understanding of the Bitcoin protocol is necessary. Cryptographically, Bitcoin is composed of two primitives: a digital signature scheme (in practice, ECDSA) and a one-way hash function (in practice, SHA-256). Users' pseudonyms are public keys for the signature scheme, and users can create arbitrarily many pseudonyms by generating signing keypairs. In here and what follows, we use Bitcoin to mean the peer-to-peer network and abstract protocol, and bitcoin, or BTC, to mean the unit of currency; we also use the terms public key, address, and pseudonym interchangeably.

To see how bitcoins get spent, suppose a user has some number of bitcoins stored with one of his pseudonyms. For simplicity, we describe transactions with one input and one output, but transactions can more generally have any number of input and output addresses. To send these bitcoins, the user first creates a message containing (among other things) the intended receiver of the bitcoins, identified by public key, and the transaction in which his pseudonym received the bitcoins. The sender can then sign this message using the private key corresponding to his pseudonym to create a signature. He then broadcasts the signature and message—which together make up the transaction—to his peers, who in turn broadcast it to their peers (see Figure 1).

Before broadcasting the transaction, each peer confirms that the transaction is valid by checking for two things: first, that the signature verifies and thus (by the unforgeability of the signature scheme) was formed correctly by the honest owner of the bitcoins; and second, that no other transaction already used the same previous transaction. This second property is crucial in ensuring that the bitcoins are not double-spent, which is why every peer needs to have access to the entire transaction history (or at least to the transactions in which the received bitcoins have not already been spent). A bitcoin is then not a single object, but rather a chain of these transactions.

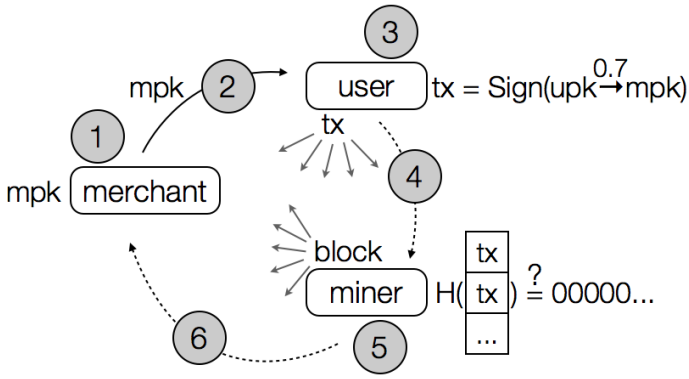

Figure 1: How a Bitcoin transaction works: In this example, a user wants to send 0.7 bitcoins as payment to a merchant. In (1), the merchant generates or picks an existing public key mpk, and (2) sends this public key to the user. By creating a digital signature (3), the user forms the transaction tx to transfer the 0.7 bitcoins from his public key upk to the merchant's address mpk. In (4), the user broadcasts this transaction to his peers, which (if the transaction is valid) allows it to flood the network. In this way, a miner learns about his transaction. In (5), the miner works to incorporate this and other transactions into a block by checking whether their hash is within some target range. In (6), the miner broadcasts this block to her peers, which (if the block is valid) allows it to flood the network. In this way, the merchant learns that the transaction has been accepted into the global block chain, and has thus received the user's payment.

After transactions such as these flood the network, they are collected into *blocks*, which serve to timestamp the transactions and further vouch for their validity. The process of creating a block is called *mining*, as it is also the process by which bitcoins are created. Miners (i.e., users seeking to create blocks) first collect all the transactions they hear about into a pool of transactions that have not already been incorporated into blocks; priority often is given to transactions that include a small fee, although at present most transactions do not need to include a fee (the exceptions being transactions that have many inputs and/or outputs, or transactions that carry a large amount of bitcoins). The miner then adds a special *coin generation* transaction to the pool and hashes this collection of transactions.

The miner aims to have a collection of transactions (and other metadata, including a reference to the most recently generated block) that hashes to a value starting with a certain number of zeroes. This and what follows are a somewhat simplified sketch of the mining process; in reality, the miner is trying to generate a

hash that is smaller than some target hash. The required number of leading zeroes is proportional to the *difficulty* of the network, which is determined by its current hash rate. The goal is to have the network produce a new block every ten minutes, so the difficulty is adjusted accordingly (e.g., if the hash rate increases, then the difficulty increases as well).

In order to produce this target hash while maintaining the same pool of transactions, the miner also folds in a *nonce* value. One can then think of the mining process as starting with the collection of transactions and the nonce set to 1; if this produces a hash within the target range, then the miner has produced a valid block, and if it doesn't, then she can increment the nonce and try again.

Once the miner does have a valid block, she broadcasts it throughout the network in a manner analogous to the broadcast of transactions, with peers checking the validity of her block by checking whether its hash is within the target range. Her block is accepted into the global transaction ledger after it has been referenced by another block. Because each block references a previous block, blocks form a chain just as transactions do, so this transaction ledger is referred to as the *block chain*.

As a reward for generating this block, which, because of the one-wayness of the hash function, is a computationally intensive task, the miner receives a certain number of bitcoins in the public key specified in her coin generation transaction. This number of bitcoins is determined by the *height* of the block chain: initially, the reward was 50 bitcoins, but at height 210,000 (i.e., after 210,000 blocks were generated, which happened on November 28, 2012), the reward halved, and will continue halving until 21 million bitcoins are generated, at which point the reward will be 0 and miners will be incentivized solely by transaction fees, which will presumably increase as a result.

To summarize, the ledger that every peer downloads when joining the Bitcoin network is the block chain, which consists of a series of blocks, each referencing the one that preceded it. Blocks are accepted into the block chain by consensus: if enough peers agree that a block is valid (for example, it is within the required target range and creates an appropriate number of bitcoins), then they will choose to reference it when generating their own blocks, so that the mining of blocks (and consequent generation of bitcoins) follows a consensus-defined set of rules rather than system requirements. These blocks contain collections of transactions that, like blocks, are validated through their acceptance by peers in the network, which specify the transfer of bitcoins from one set of pseudonyms to another.

Where Bitcoins Are Spent

As of April 13, 2013, the block chain contained more than 16 million transactions between 12 million distinct public keys. More than 11 million bitcoins had been generated (recall that this is more than half of all the bitcoins that will ever be generated), and those bitcoins had been spent many times over, to the point that more than 1 trillion bitcoins had been transacted.

Given this rate of movement, one might naturally wonder where bitcoins are being spent. Since 2010, a variety of Bitcoin services have been introduced at an ever-increasing rate. One of the most widely used categories, *exchanges*, allows users to exchange bitcoins for other currencies, including both fiat currencies such as dollars, and other virtual currencies such as Second Life Lindens. Most of these exchanges also function as banks, meaning they will store your bitcoins for you, although there are also *wallet services* dedicated to doing just that. With all of these services, one runs the risk of theft, which in fact happens fairly often.

Bitcoin mining ASICs were introduced in February 2013 and are capable of computing 64 billion SHA-256 computations per second, meaning the odds of generating a block using just a CPU or even GPU are negligibly small . Due to the computational intensity of generating bitcoins, *mining pools* have become another popular service in the Bitcoin economy, allowing miners to perform some amount of work (e.g., the examination of some slice of the nonce space) and earn fractional bitcoin amounts for every share they contribute.

Users seeking to spend rather than only store or generate bitcoins can do so with a number of merchants, including ones such as WordPress that use the payment gateway BitPay, which accepts payment in bitcoins but pays the merchant in the currency of their choice (thus eliminating all Bitcoin-based risk for the merchant). Users can also gamble with their bitcoins, using poker sites such as BitZino or wildly popular dice games such as Satoshi Dice.

Finally, users seeking to use Bitcoin for criminal purposes can purchase drugs and other contraband on sites such as Silk Road, which are often accessible only via the Tor network. They can also mix (i.e., launder) bitcoins with services such as Bitfog, which promise to take bitcoins and send (to the address of one's choice) new bitcoins that have no association with the ones they received.

The first phase of our analysis involved interacting with these and many other services. In total, we kept accounts with 26 exchanges and ten wallet services, and made purchases with 25 different vendors, nine of which used the payment gateway BitPay; a full list of the services with which we interacted can be found in Table 1, and images of our tangible purchases can be found in Figure 2. We engaged in 344 transactions

Figure 2: The physical items we purchased with bitcoins, ranging from beef jerky from BitPantry to a used Boston CD from Bitmit. The items in green were purchased from CoinDL (the "iTunes of Bitcoin"), in blue from Bitmit (the "eBay of Bitcoin"), and in red using the payment gateway BitPay.

with these services, which allowed us definitively to tag 832 addresses (recall that transactions can have arbitrarily many input addresses, which allows us to tag multiple addresses per transaction). We additionally scraped various publicly claimed addresses that we found, such as users' signatures in Bitcoin forums, although we were careful to use only tags for which we could perform some manual due diligence.

Clustering Bitcoin Addresses

In theory, the use of pseudonyms within Bitcoin provides a property called *unlinkability*, which says that users' transactions using one set of pseudonyms should not be linked to their transactions using a different set of pseudonyms. In practice, however, certain properties of Bitcoin usage erodes this anonymity.

Recall that, in order to create a valid Bitcoin transaction, the sender must know the private signing key corresponding to the public key in which the bitcoins are held. Now suppose that a user wishes to send 10 BTC to a merchant, but has 4 BTC in one address and 6 BTC in another. One potential way to pay the merchant would be to create a new address, send the 4 BTC and 6 BTC to this new address, and then send the 10 BTC now contained in this new address to the merchant. (In fact, this is the method that preserves the most anonymity.) Instead, the Bitcoin protocol allows for a simpler and more efficient solution: transactions can have arbitrarily many inputs, so the 4 BTC and 6 BTC addresses can be used as input to the same transaction, in which the receiver is the merchant.

Mining			
	50 BTC	BTC Guild	Itzod
	ABC Pool	Deepbit	Ozcoin
	Bitclockers	EclipseMC	Slush
	Bitminter	Eligius	
Wallets			
	Bitcoin Faucet	Easywallet	Strongcoin
	My Wallet	Flexcoin	WalletBit
	Coinbase	Instawallet	
	Easycoin	Paytunia	
Exchanges			
	Bitcoin 24	BTC-e	Aurum Xchange
	Bitcoin Central	CampBX	BitInstant
	Bitcoin.de	CA VirtEx	Bitcoin Nordic
	Bitcurex	ICBit	BTC Quick
	Bitfloor	Mercado Bitcoin	FastCash4Bitcoins
	Bitmarket	Mt Gox	Lilion Transfer
	Bitme	The Rock	Nanaimo Gold
	Bitstamp	Vircurex	OKPay
	BTC China	Virwox	
Vendors			
	ABU Games	BTC Buy	HealthRX
	Bitbrew	BTC Gadgets	JJ Games
	Bitdomain	Casascius	NZBs R Us
	Bitmit	Coinabul	Silk Road
	Bitpay	CoinDL	WalletBit
	Bit Usenet	Etsy	Yoku
	Bit Elfin	BitZino	Gold Game Land
	Bitcoin 24/7	BTC Griffin	Satoshi Dice
	Bitcoin Darts	BTC Lucky	Seals with Clubs
	Bitcoin Kamikaze	BTC on Tilt	
	Bitcoin Minefield	Clone Dice	
Miscellaneous			
	Bit Visitor	Bitfog	CoinAd
	Bitcoin Advertisers	Bitlaundry	Coinapult
	Bitcoin Laundry	BitMix	Wikileaks

Table 1: We interacted with many services, and provide approximate groupings as shown here.

This observation gives rise to our first clustering heuristic: if two addresses have been used as input to the same transaction, they are controlled by the same user. This heuristic is quite safe, as the sender must know the private keys corresponding to all input addresses in order to form a valid transaction, and as such it has already been used in the Bitcoin literature to the point where freely available tools exist online for performing this analysis.

Our second clustering heuristic expands on this first heuristic and exploits the way in which change is made. In the Bitcoin protocol, when an address receives some number of bitcoins, it has no choice but to spend those bitcoins all at once (recall that this is because each transaction must reference a previous transaction, and transactions cannot be referenced multiple times). If this number of bitcoins is in excess of what the sender wants to spend (e.g., if he has 4 BTC stored in an address and wants to

send 3 BTC to a merchant), then he creates a transaction with two outputs: one for the actual recipient (e.g., the merchant receiving 3 BTC) and one change address that he controls and can use to receive the change (e.g., the 1 BTC left over).

This behavior gives rise to our second clustering heuristic: the change address in a transaction is controlled by the sender. As change addresses do not a priori look any different from other addresses, significant care must be taken in identifying them. As a first step, we observe that in the standard Bitcoin client, a change address is created internally and is not even known to the user (although he can always learn it by examining the block chain manually). Furthermore, these change addresses are used only twice: once to receive the change in a transaction, and once to spend their contents fully as the input in another transaction (in which the client will create a fresh address to receive any change).

By examining transactions and identifying the outputs that meet this pattern of one-time usage, we identify the change addresses. If more than one output meets this pattern, then we err on the side of safety and do not tag anything as a change address. Using this pattern—with a number of additional precautions, such as waiting a week to identify change addresses—we identified 3.5 million change addresses, with an estimated false positive rate of 0.17%, noting that the false positive rate can only be estimated, as in the absence of ground-truth data we cannot know what truly is and isn't a change address. By then clustering addresses according to this heuristic, we collapsed the 12 million public keys into 3.3 million clusters.

Putting It All Together

By layering our clustering analysis on top of our ground-truth data (and thus transitively tagging entire clusters that contain previously tagged addresses), we were able to identify 1.9 million public keys with some real-world service or identity, although in many cases the identity was not a real name, but rather (for example) a username on a forum. Although this is a somewhat small fraction (about 16%) of all public keys, it nevertheless allows us to de-anonymize significant flows of bitcoins throughout the network.

Toward this goal, we first examined interactions with known Bitcoin services. By identifying a large number of addresses for various services (e.g., we identified 500,000 addresses as controlled by Mt. Gox, and more than 250,000 addresses as controlled by Silk Road), we were able to observe interactions with these services, such as deposits into and withdrawals from exchanges. Although this does not de-anonymize the individual participating in the transaction (i.e., we could see that a user was interacting with a service, but did not necessarily know which user), it does serve to de-anonymize the flow of bitcoins into and out of the service.

To demonstrate the usefulness of this type of analysis, we turned our attention to criminal activity. In the Bitcoin economy, criminal activity can appear in a number of forms, such as dealing drugs on Silk Road or simply stealing someone else's bitcoins. We followed the flow of bitcoins out of Silk Road (in particular, from one notorious address) and from a number of highly publicized thefts to see whether we could track the bitcoins to known services. Although some of the thieves attempted to use sophisticated mixing techniques (or possibly mix services) to obscure the flow of bitcoins, for the most part tracking the bitcoins was quite straightforward, and we ultimately saw large quantities of bitcoins flow to a variety of exchanges directly from the point of theft (or the withdrawal from Silk Road).

As acknowledged above, following stolen bitcoins to the point at which they are deposited into an exchange does not in itself identify the thief; however, it does enable further de-anonymization in the case in which certain agencies can determine (through, for example, subpoena power) the real-world owner of the account into which the stolen bitcoins were deposited. Because such exchanges seem to serve as chokepoints into and out of the Bitcoin economy (i.e., there are few alternative ways to cash out), we conclude that using Bitcoin for money laundering or other illicit purposes does not (at least at present) seem to be particularly attractive.