REVIEW OF CRYPTONOTE WHITE PAPER

SURAE NOETHER

This paper is dedicated to Emmy Noether, Satoshi Nakamoto, and the Bourbaki Group.

Abstract. This document is not intended as financial advice; it is one mathematician's take on the white paper before digging into the code. I've learned a lot writing this document. I am writing this review, in part, to augment the academic rigor of the CryptoNote (CN) white paper, which is already miles ahead the available alternatives, but also to critique that document for it's inconsistencies and incompleteness. For full disclosure, I was hired by the Monero (XMR) developers to investigate the CryptoNote protocol and the ByteCoin code base from which Monero has been forked. The folks involved in hiring me have had no involvement in my review process other than answering my technical questions and sending me money occasionally when I ask them politely and show them the annotations I've made. They are paying me in Bitcoin, not in any CryptoNote based currency. Yes, I do own and hold CryptoNote currencies.

Further, I'd like to preemptively apologize to Nicolas van Saberhagen and the CryptoNote developers if I step on any of your toes. You've probably been knee-deep in this for years. If I have said something blatantly incorrect, idiotic, uninformed, or wrong, I am happy to correct the record and be informed. I am here to learn, not to lecture. Just shoot me an e-mail, we'll chat. But, I was hired to write a critique, I'm a mathematician, and some of your indexing was wrong. Don't take it personally.

Introduction

Decentralization and anonymity are important in the financial world because of the inevitable conflicts of interest between any centralization authority and users. The Man needs to be paid, and we need our privacy. It's not a bad thing, it's just how it is, and it can be solved with technology. Bitcoin was the first decentralized, peer-to-peer, pseudonymous attempt at a solution. To this author's knowledge, few other truly unique solutions have been proposed. Overall, the CryptoNote (CN) protocol represents the first new step in the cryptocurrency space since Bitcoin and it's one that deserves as much shoulder space as Bitcoin. It's a heavy hitter, no doubt about that, with quite a few basic improvements over the Bitcoin protocol and a few big improvements. This paper is intended to review the CN white paper, point out at least some of the advantages and disadvantages of the proposed protocol, and illustrate points of possible improvements of the protocol.

So how does CN work? Well, just like anything in the cryptography world that works well, it works weirdly. We can imagine the CN protocol as a post-office-box system. Each user has a set of public keys and private keys, just like in the BTC protocol. Rather than sending CrytpoNote directly to each others public keys, users execute a Diffie-Hellman exchange and create ring signatures to make a new

Date: July 14, 2014.

one-time post-office-box at which the CNs are stored. And when we send our CN, we include our key image, which is just the hash of the private destination key that gave us the right to send those CN in the first place. If that key image has not yet been used, then that one-time ring signature has not yet been spent.

In a sense, were saying "Okay, see that box over there? It has 1.0 CN in it and I have a private key K to it. With a Diffie-Hellman-exchange, you make a new private key K^* and I will make a new public address and I will send the 1.0 CN to the new address and I will announce the key image $\mathrm{Im}(K)$ used to do it. Anyone who sees this transaction will check if $\mathrm{Im}(K)$ has been used before to make sure that my box still has my 1.0 CN inside of it, and now you have a new private key K^* to your very own post-office-box stuffed full of cash!"

Or, the way I like to think about it, someone says "Everyone take your hands off the money while it is being transferred around! It is simply enough to know that your keys can open that box... oh, wait, no! It's enough to know that a hash of your keys are equivalent to the location of that box! And that we know how much money is in the box! Never put your fingerprints on the post-office-box or actually use it, just trade the hashes of the keys that are equivalent to the locations of the boxes filled with cash! That way we don't know who sent what! But these trades are still friction-less, fungible, divisible, and still possess all the other nice qualities of money we desire like in the Bitcoin protocol."

Now, as weird as that sounds, that's how CryptoNote works. I considered putting a flow-chart of the whole thing up on a poster at work, just to see what people say. After reading through the entire white paper, my biggest concern is whether the elliptic curve parameters were chosen fairly, or how an honest coin developer could go about changing them if they wanted.

I'm going to organize this document in the following order:

(I) literature review,
(II) improvements CN represents over BTC,
(III) possible problems with the CN protocol,
(IV) deficiencies in the white paper,

1. Literature Review

Okay, I'll be honest; like most academics, I just skimmed van Saberhagen's lit review. But gosh, van Saberhagen picked some damn good papers for me to skim. Van Saberhagen's history of ring signatures, starting from group signatures, is quite interesting. Group signatures started out as the first way of allowing any public member of a group to anonymously sign a message on behalf of that group, in which there was a group manager who could, at her own discretion, revoke user's anonymity. The name "group signature" annoyed algebraists, who pretty much invented cryptography, thank you very much.

Ring signatures were developed to remove the centralization point of a group manager. The name also annoyed algebraists. Essentially, if everyone has a public key and a private key pair, all we do is sign our message with our private key like usual, and then publish a set of our friends' public keys together with our own public key, and establish some protocol so that we use the whole public key set to verify the message. Simple! This establishes untraceability.

Then *linkable ring signatures* came along, which is confusing because, using our terminology in this paper, it's really *not untraceable*, or at least scalably traceable. These are ring signatures in which the sender can essentially revoke their own anonymity as desired. Next up came *traceable ring signatures*, which is also confusing, because it doesn't have to do so much with traceability in our sense. The way traceable ring signatures work is to establish a one-time method of using a ring signature, perhaps in voting schemes. Definitely useful in any token-based system. Finally, we get to *ad hoc group signatures* in which we just pick our group members on the fly. Using the historical constructions

- Group signatures
- Untraceable Ring Signatures
- Scalably Traceable Ring Signatures
- One-time-use Scalably Traceable Ring Signatures
- Ad hoc group signatures

Van Saberhagen glued all of that together to get CryptoNote. Pretty clever, really, if you ask me. However, we aren't going to use any of that terminology at all for the rest of the paper. That's just where all of this fits into the "old crypto" context. Which, it turns out, is super important: if you try new crypto, you usually get burned.

2. Improvements over Bitcoin

Little improvements over the Bitcoin protocol abound abound in CryptoNote. Your password is equivalent to your private keys, and each user needs only one address. There is no transaction collisions. You can give away half of your private keys without fear of lost security to, say, a payment processor. Any user can generate an income-auditable address by deterministically generating half of their keys. Transaction scripts are super simple. Global variables like block size and block reward dynamically adjust so we need not worry about network consensus for infrastructure-like changes to code. Really, most of the nicest benefits are just a bunch of little things.

2.1. **Untraceability and Unlinkability.** Some bigger stuff makes you just *feel good*, even if it isn't really a feature: a proof that transactions are unconditionally unlinkable (under the random oracle assumption). According to van Saberhagen, two transactions are *unlinkable* if we can't prove they went to the same person, and a system is unlinkable if, for any two transactions, they are unlinkable. The random oracle assumes the existence of some perfect hash function, which is somewhat unrealistic but not well approximated by current hash functions. The only better proof would be under the standard model, and almost no cryptographic models are proved under the standard model. No other developers have proven anonymity in their coins. I cannot stress this enough. *No other coin has the weight of mathematical proof behind their product.* Even Bitcoin only recently has had a rigorous security analysis applied to its methods, and it is known that Bitcoin fails unlinkability and untraceability! This absolutely blows any competing coin out of the water. The closest contender is Zerocoin/Zerocash. The CryptoNote white paper is wrong in some of their criticism of Zerocash, by the way, since the Zerocash protocol has made a recent breakthrough in their size constraints; however, I quote CN developer Maurice Planck on this one:

> [Maurice P]Zerocoin, Zerocash. This is the most advanced technology, I must admit. Yes, the quote above [from the white paper] is from the analysis of the previous version of the protocol. To my knowledge, its not 288, but 384 bytes, but anyway this is good news [the latest trimming of sizes]. They used a brand new technic [sic] called SNARK, which has certain downsides: for example, large initial database of public parameters required to create a signature (more than 1 GB) and significant time required to create a transaction (more than a minute). Finally, they're using a young crypto, which I've mentioned to be an arguable idea: `https://forum.cryptonote.org/viewtopic.php?f=2&t=19#p55`

Now, notice that since we are still pushing information through a function (our random oracle function), and not using a zero-knowledge system, our system is still not fully zero-knowledge anonymous. Andrew Poelstra described a wonderful state-of-the-art on anonymous coins on the CryptoStackExchange website, including an in-depth discussion about CryptoNote and some proposed tweaks.

Transactions are also scalably untraceable. According to van Saberhagen, a transaction is untraceable if all possible senders are equiprobable, meaning a sender chooses a ring signature set of public keys, and from any attacker's point of view, all members of that set are equiprobable as possible senders. Further, this feature is scalable in the sense that you can choose the size of your obfuscating set (ambiguity degree is the number of public keys in your obfuscating set), and even choose your ambiguity degree to be $n = 0$ if you so desire.

Van Saberhagen, lists two desirable properties of a cryptocurrency: untraceability and unlinkability, and we've mentioned both. Let's say I'm listening in on some cryptocurrency network and I compare two transactions. We can imagine a transaction as being an arrow from one user to another with an amount as it's length. So we have two arrows, of different length. Problem is, we don't know who is sending or receiving, right? At least, ideally, we don't know. So we'll just throw random names on here:

$$A = \text{Alice} \xrightarrow{\qquad\qquad 1.101 \qquad\qquad} \text{Bob} = B$$

$$C = \text{Charlie} \xrightarrow{\quad 0.0637 \quad} \text{Danielle} = D$$

Van Saberhagen, defines a system to be *untraceable* if, for each incoming transaction, all possible senders are equiprobable, and *unlinkable* if, for any two outgoing transactions, it's impossible to prove they were sent to the same individual. I interpret "impossible" here to mean "of probability that can be made arbitrarily small, if not zero." Let's pause and think about these definitions, because they are great! Notice that the best possible "anonymous" coin would provide no information whatsoever about any given transaction. That is to say, any given sender is as likely as another for any given transaction. So this definition of untraceable is nice and natural.

3. Problems with the Protocol

This is saying a lot: my single biggest question after reading the entire paper is the "how did they choose their elliptic curve constants?" The protocol appears

sound; who chose the constants? Will there be a plan for choosing new constants in the future if needed? How can I choose other constants if I decide to fork it? Did the NSA come up with CryptoNote and choose these constants so any CryptoNote network has 10% the entropy of any other coin? Who knows. It's probably not a big deal, and every coin has this as a critical point. Indeed, it's a centralization point. If we all go happily forking the CryptoNote code left and right, we are trusting those developers to have made good decisions on the constants.

Next up is the not-so-obvious, already mentioned: this is not a zero-knowledge system, so some information is still preserved after each step. Andrew Poelstra made a wonderful post on CryptoStackExchange about it:

> [Andrew Poelstra] This [one-time ring signature scheme] provides good anonymity, but even with the improvements listed presently, this is not a zero-knowledge scheme. This means that linkability is confounded but an adversary with good analysis tools will certainly be able to glean a non-zero (literally, innity times as much as zero) amount of information.

Andrew is being a bit hyperbolic here. Non-zero simply means "not zero" whereas he's thinking of infinitesimal. Doesn't matter, point is made! The idea is this: a zero-knowledge proof does not use ANY information to construct the proof. Whereas, for example, a random oracle H H(private stuff), a function of the keys. It's "random," it's uniquely identified with the keys in a way that no outsider can duplicate, and so on, but information is passed through the function. Therefore, if I were the God Almighty on high, if I were the Greek God of Entropy and Statistics, I could peer through this mortal function H and recover your private information.

Anonymity can be violated in a few ways; any time you spend an output and set the ambiguity degree $n = 0$, you reveal yourself as the spender of that CryptoNote and anyone can go back through the blockchain. Any obfuscating output set with your now-spent output as a member becomes less ambiguous by a degree of 1. Indeed, ambiguity degree becomes monotonically decreasing over time. However, users never need to set a low ambiguity number since they can prove they made a payment in other ways.

More drawbacks include: keys are twice as large as in the Bitcoin protocol, the CN protocol experiences long-term uncontrolled growth of unspent transaction outputs (UTXO) and a large blockchain. Unfortunately, this seems to simply be ignored by the author, but, honestly, I probably would have ignored it too. You invent something, and it's really heavy. You bring it to the County Fair. Are you going to be like... "hey guys, look at my really heavy thing?" Or are you going to be like... "hey guys, this thing will cut your hair and take your dog for a walk!"

Some jerk may come out of the crowd and may be like "but, dude, that thing weighs like 1000 pounds and it gets heavier every time it sucks up your hair or picks up your dog's doo, which are critical tasks with respect to this creation of yours." And you just shrug, because you made something, right?

Anyway, some other guy may come along and figure out that it needs a hatch so you can lighten it up occasionally. Apparently Andrew Poelstra and G. Maxwell are both working on that now, using Merkle trees and required prefixes for one half of the Cryptonote private keys (the half you would give away to a payment processor). I am hoping to come up with something.

3.1. **New technology.** The CN protocol implements a piece of cryptography unseen in cryptocurrencies before, in particular, the idea of using *key images* to protect against double spending. This is boldly treading on dangerous ground; no matter how deeply I, or any mathematician scrutinizes an algorithm in any white paper, it's possible some 16 year old in South Africa will figure out a way to crack the encryption. On the other hand, if you throw together some RSA or ECDSA libraries, you know that works. You know that works. This is due to a famous effect in mathematics called "Just because I can't see it doesn't mean it isn't there." I have looked through the CryptoNote white paper and it looks good. This just means I'm not as clever as whoever will eventually break CN and BTC wide open. On the other hand, the number of eyes and brains trying to crack open "old cryptography" is different by orders of magnitude, with history on its side. However, the only thing for it at this point is to let it stand the test of time.

3.2. **New Algorithm.** Van Saberhagen makes excellent points that cost of investment should grow faster than linearly with power, and ge describes a perfect algorithm to accomplish the task. But without providing that algorithm, it's just a bunch of snake-oil. I guess the proof is in the code. However, implementing an entirely new Proof-of-Work algorithm could be just as vulnerable to exploitation as implementing any new piece of software. To be frank, without any sort of explicit, clear explanation of how it's been done, it can't necessarily be trusted. With Bitcoin, the task was clear: find the nonce so that the SHA hash is small. With this algorithm? I have no idea.

Over the past few years, and probably for the next few years, it's been the case that a CPU is better at dealing with stuff that requires lots of random access to memory, whereas GPUs and ASICs have been better at dealing with sequential, iterated data that can be constructed in a lazy way. So, it *appears* that van Saberhagen has simply taken the Scrypt construction and either iterated it so that each hash depends on all previous states, rather than just the last previous state, or concatenated the input, or something along those lines. This way, all previous states need to be kept in memory and randomly accessed, the process can't be sequentialized easily, and it will be years before ASICs can handle it. However, this is just my best guess. I have no good reason to think this based solely on my reading of the white paper.

What we need is an explicit description of the algorithm, and we need some analysis done on the algorithm.

3.3. **Dynamic Variables.** Variables adjust dynamically in time. This is, if you recall, a positive from above. It's also a negative. If care is not taken, this can lead to either blow-ups, or wild fishtailing. The CryptoNote authors propose rejecting blocks if they are too large (larger than twice the median). This can, if a long-term attack is executed, lead to an exponential blowup of the blockchain. It's unlikely, and costly, but possible. Furthermore, given the already unwieldy size of the CryptoNote blockchains, doubling the size of the average block even once or twice could lead to significant problems with the network. A true "blow-up" to infinity is not necessary to cause disruption, and a smaller attack can be avoided mathematically.

To discourage such behavior, a block reward penalty for abnormally sized blocks is introduced; but this may lead to increased fees in times of high economic traffic

like Christmas shopping, which would be intolerable to customer acceptance of the coin. This encourages us to find a mathematical solution to block-acceptance rather than to find an economic incentive to users.

4. Deficiencies in the white paper and Further Questions

The white paper is well-organized in terms of sections, for the most part, but extremely poorly written and uses inconsistent terminology. But guys, I'm going to give van Saberhagen a break: there is a LOT of information in a white paper. It can't really be a manual, but in my opinion, there should at least be enough information to re-develop the technology from scratch. Vital information is left out, important equations are not indexed correctly, and notation is left unexplained. The so-called "standard transaction sequence" in section 4.3 doesn't include any information about where signatures take place. Unfortunately, a lot of this section is... simply hard to read and not well explained. Bad notation is typical; is that the destination key? Or the transaction public key? Which is what? Where is the key image? Where is it even used? Check the diagram!

For example, the following two statements should be strung together and followed with an explanatory diagram and a bit of pseudocode.

> [Nicolas van Saberhagen] The identity of the signer is indistinguishable from the other users whose public keys are in the set until the owner produces a second signature using the same key pair.
>
> [Nicolas van Saberhagen] In case Alice wants to prove she sent a transaction to Bobs address she can either disclose r or use any kind of zero-knowledge protocol to prove she knows r (for example by signing the transaction with r).

This aspect of CryptoNote (choosing to violate one's own untraceability to prove payment) is critical for usage as a currency, and hands are waved.

It's absolutely unconscionable to to come up with a new "Proof of Work Algorithm" and then refrain from including any sort of pseudocode to describe that algorithm. Upon which. Your entire. Coin. Is. Based. Ugh.

5. Conclusion

The CryptoNote protocol is absolutely spectacular. It cannot really be compared to Bitcoin because a layer of anonymization takes place between a user's public addresses and their transactions. Further, myriad improved features are scattered throughout. It's a genuinely *different* way of transferring wealth cryptographically via Blockchain-by-Proof-of-Work, comapred to Bitcoin. It cannot be directly said to be a Bitcoin 2.0, but more like a completely different protocol that establishes and obtains different objectives.

There are some critical problems with CryptoNote. The size of the the entire project is just enormous. Key sizes are double the usual size. Unspent transaction output sets and key image sets both grow in an uncontrolled way. Most troubling is the centralization point of allowing an anonymous person on the internet choosing all of our elliptic curve constants without explaining himself.

However, having said all that, CryptoNote is absolutely spectacular. If you have a problem with the constants, and if you can figure out how to generate new ones, I say go for it. The protocol looks secure and tight.

BTC Address: 192VR9ZH83URGRRASCZF9DX85DCCSJ6J6E
E-mail address: surae.noether@gmail.com