ShadowCash: Zero-knowledge Anonymous Distributed E-Cash via Traceable Ring Signatures

Rynomster and Tecnovert
http://www.shadow.cash
sdcoin@sdcoin.co

Abstract-We introduce Shadowcash, an anonymous cryptographic transaction protocol: Anonymous transactions are implemented using traceable ring signatures[5]**, which utilise a non-interactive zero knowledge proof**[6]**.**

1. INTRODUCTION

We believe privacy is a human right - as enshrined in article 12 of the Universal Declaration of Human Rights of the United Nations. Transactions of value are an essential part of our daily lives. As such we strive to provide you with tools to transact in confidence[1].

E-cash systems, or virtual currencies[2], have become a very common way to transact due to the many benefits they hold over traditional methods of exchange. One of the largest problems for virtual currencies is preventing double-spending, where a currency holder sends the same coins to multiple recipients. Bitcoin solves this problem using the blockchain, a public record of all transactions in the system. By viewing the blockchain, all participants in the currency can see the current state of the system at the same time, thus the double-spending problem is solved. However, adding the blockchain causes a severe reduction of the anonymity and privacy of the participants in the currency[3]. As the blockchain is public, anyone can see the transactions and total holdings of participants, unless suitable precautions are taken.

2. Overview

In this paper we will present our anonymous cryptographic transaction protocol which utilises: dual-key stealth addresses, traceable ring signatures and non-interactive zero knowledge proofs.

We prove how our scheme introduces a much higher level of privacy and anonymity to the network while still preserving the core principles of trustless decentralization, unforgeability and double-spend prevention. We will also present performance data of our scheme which includes proof sizes, signature generation times and verification times.

Finally, we will present planned future improvements to the current scheme. We present this paper as a first draft towards receiving peer review.

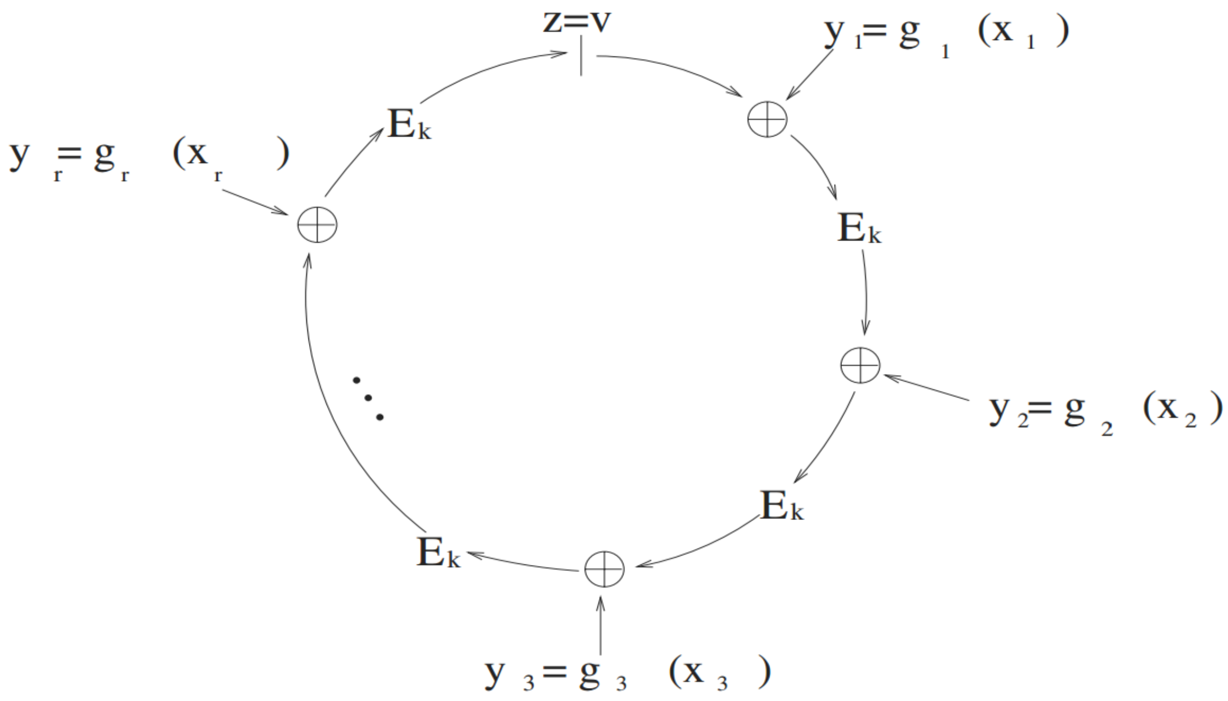

Fig. 1. Ring Signatures

3. Decentralized E-Cash

3.1. Trustless

3.1.1. Our system functions on the same core principles from which Bitcoin was founded. There is no central authority or bank mechanism that controls the flow of transactions. Furthermore, our scheme does not require the initial trusted parameter setup which is present in the Zerocoin and Zerocash scheme.

3.2. Unforgeability

3.2.1. In order to transact anonymously, we have introduced an anonymous token[8], which we will refer to as Shadow. Shadow can be minted, which will destroy SDC (ShadowCash), and will output a group of Shadow tokens totaling the same value (minus the transaction fee) of the destroyed SDC.

3.2.2. Shadow tokens take the form of outputs on the ShadowCash chain. Shadow tokens are spendable only by providing a traceable ring signature to prove ownership of the token..

3.3. Anonymity

3.3.1. The ring signature consists of the public key of the token being spent, plus the public keys from 3 to 200 other tokens of the same value as the token being spent. The nature of ring signatures makes it impossible to discover which of the member coins in the ring signature is being spent, and transactions are no longer traceable.

3.3.2. It is not possible to determine which tokens have been spent, so all tokens remain in the blockchain as spendable outputs available as members of ring signatures for other token spends.

3.3.3. To increase the pool of outputs available for ring signatures, the SDC value is broken up into separate Shadow tokens for each decimal place of the total value. The tokens are further broken up to values of 1, 3, 4 and 5. For example 1.7 sdc would become 3 tokens of values 1.0, 0.3 and 0.4.

3.4. Double-Spend Prevention

3.4.1. The ring signature tags (**keyImage**) of the spent Shadow tokens are embedded in the blockchain to prevent double spends. Each tag is unique to the Shadow token, regardless of the other members of the ring signature.

4. Spending Shadow

There are two ways in which Shadow tokens can be spent: they can be sent as Shadow tokens or redeemed as SDC.

1. When sent as Shadow tokens, new tokens are minted for the recipient to the value of the input Shadow minus the transaction fee.

2. When redeemed as SDC, new SDC is created to the value of the input Shadow minus the transaction fee.

In both cases the input tokens become unspendable.

The transaction fee for spending Shadow is 100x greater than the fee for standard transactions. This is to cover the cost of the extra activity required by the network to transmit, verify and store shadow transactions, which are larger and require more processing than standard transactions.

In order to spend Shadow, we use ring signatures to sign the transaction[5][6]. Our scheme consists of three functions, **generateRingSignature**, **generateKeyImage**, **verifyRingSignature**.

For efficiency's sake, when spending Shadow, we get a list of all anonymous outputs in the system, then we remove coins that don't have enough same value outputs in the system, then we choose the smallest coin or least number of smallest coins that can cover the amount + transaction fee.

Each Shadow coin has its own private key, so when spending Shadow, each coin or anonymous input, will need to have its own ring signature generated, and will then have to be verified.

- **generateRingSignature**
  - Executed by the sender / signer, when spending Shadow.
  - After executing generateRingSignature, the ring signature will have to be verified.
  - Takes an input-output / reference to the keyImage, the transaction hash, the ring size, the secret key offset, the secret key for signing the transaction, a list of public keys for all the coins or outputs in the ring signature and the ring signature, and a hash of the transaction in which the signature is included (preimage).

- **generateKeyImage**
  - Executed by the receiver, when receiving Shadow

- The key image is revealed to the network to prevent a token from being spent more than once.

- **verifyRingSignature**
  - Executed by each node, when connecting the inputs of anonymous / Shadow transactions.
  - A preimage is calculated for the transaction and verified against the ring signature.
  - The public key of each input token in the transaction is extracted and is looked up in the blockchain to ensure it refers to an existing, valid token.
  - The blockchain is searched for the provided keyImage, if one is found the transaction is considered a double-spend attempt and denied.
  - Takes an input-output / reference to the keyImage, the transaction hash, the ring size, the a list of public keys for all the coins or outputs in the ring signature and the ring signature.

4.1. Prover

4.1.1. **Signing protocol** : To sign message m ∈ {0, 1}

To sign message m ∈ {0, 1} ∗ with respect to tag L = (issue, pk N), using the secret-key sk$_i$, proceed as follows:

1. Compute $h = H(L)$ and σ i = h^{Xi}, using $x_i \in Zq$.
2. Set $A_0 = H^{'}(L, m)$ and $A_1 = (\frac{\sigma_i}{A_0})^{1/i}$
3. For all $j \neq i$, compute $\sigma_j = A_0 A^{j}{}_1 \in G$. Notice that every $(j,\ log_h\ (\sigma_j))$ is on the line defined by $(0,\ log_h(A_0))$ and (i, x_i), where $x_i = log_h(\sigma_i)$.
4. Generate signature $(c_N,\ z_N)$ on $(L,\ m)$, based on a (non-interactive) zero-knowledge proof of knowledge for the relation derived from language $\iota \triangleq \{(L, h, \sigma_N))|\ \exists i' \in N$ such that $log_g(y_i') = log_h(\sigma_i').\}$, where $\sigma_N = (\sigma^1, ..., \sigma_n)$, as follows:
   a. Pick up random $w_i \leftarrow Zq$ and set $a_i = g^{w_i},\ b_i = h^{w_i} \in G.$
   b. Pick up at random $z_j, c_j \leftarrow Zq,$ and set $a_j = g^{z_j} y_i^{c_j}, b_j = h^{z_j} \sigma_j^{c_j} \in$ for every $j \neq i.$
   c. Set $c = H''(L, A_0, A_1, a_N, b_N)$ where $a_N = (a_1, ..., a_n)$ and $b_N = (b_1, ..., b_n).$
   d. Set $c_i = c - \Sigma_{j \neq i} c_j (mod\ q)$ and $z_i = w_i - c_i x_i (mod\ q).$ Return $(c_N, z_N),$ where $c_N = (c_1, ..., c_n)$ and $z_N(z_1, ..., z_n)$, as a proof of ι.
5. Output = (A_1, c_N, z_N) as the signature on (L, m).

4.2. Verifier

4.2.1. **Verification protocol**: To verify signature $\sigma = (A_1,\ c_N,\ z_N)$ on message m with respect to tag L, check the following:

1. Parse L as $(issue,\ pk_N)$. Check $g,\ A_1 \in G$, $c_i, z_i \in \mathbb{Z}_q$ and $y_i \in G$ for all $i \in N$. Set $h = H(L)$ and $A_0 = H'(L, m)$, and compute $\sigma_i = A_0 A_i^{i} \in G$ for all $i \in N$.
2. Compute $a_i = g^{zi} y_i^{ci}$ and $b_i = h^{zi} \sigma_i^{zi}$ for all $i \in N$.
3. Check that $H''(L, m, A_0, A_1, a_N, b_N) \equiv \Sigma_{i \in N^{ci}} (mod\ q)$, where $a_N = (a_1, ...\ a_N)$ and $b_N = (b_1, ..., b_N)$.
4. If all the above checks are successfully completed, accept, otherwise reject.

5. Performance

5.1. Proof Sizes

The affine coordinates are 64 bytes per ring member per coin value.
We store the public key or keyImage, which is 33 bytes
That leaves us with ~97 bytes / ring member / input

5.2. Benchmarks

The following benchmarks were done on an Intel(R) Core(TM) i7-3537U CPU @ 2.00GHz with 8GB of RAM, using the average times out of 300000 iterations:

Algorithm	**Ring members**	**Average Time**	**Average time / ring member**
Signing	200	449ms	2.25ms
Verification	200	440ms	2.2ms

6. Future work and Improvements

6.1. By extending the “PRF made public” paradigm by Bellare and Gold-wasser (BG), we could have a simple, general, and unified construction for a unique ring signature. The signature scheme simply uses a combination of pseudorandom function (PRF) and non-interactive zero-knowledge (NIZK) proof system (where the PRF key is committed). Using the unique ring signature framework would not only help explain prior constructions for linkable ring signatures and traceable ring signatures, but give refined constructions with simpler and more intuitive design and improved efficiency[9].

6.1.1. Unique Ring Signature Model

We begin by recalling the definition of a ring signature scheme **RS = (RK, RS, RV)** that consists of three algorithms:

- **RK** (1^{λ}). The randomized user key generation algorithm takes as input the security parameter λ and outputs a public key pk and a secret key sk.
- **RS** (sk, R, m). The probabilistic ring signing algorithm takes as input a user secret key sk, a ring R that is a set of public keys (such that $pk \in R$), and a message m to return a signature σ on m with respect to the ring R.
- **RV** (R, m, σ). The deterministic ring verification algorithm takes as input a ring R, a message m, and a signature σ for m to return a single bit b.

The following correctness condition is required: for any security parameter λ, any integer n, any $\{(pk_i, sk_i)\}_1{}^n \leftarrow RK(1^{\lambda})$ (where now $R = \{pk_i\}_1{}^n$, any $i \in [n]$, and any m, it holds that $RV(R,\ m,\ RS(R,\ sk_i m))\ =\ 1$.

Unique ring signature from the DDH assumption in the random oracle model:

- **Setup** (1^{λ}). The setup algorithm takes as input the security parameter λ and outputs a multiplicative group G of prime order q and a randomly chosen generator g of G. It also provides two hash functions $H':\ \{0,1\}*\ \rightarrow \mathbb{Z}q$. It outputs the public parameters as $pp\ =\ (\lambda,\ q,\ G,\ H,\ H')$.

- **RG** $(1^{\lambda}), pp)$. The key generation algorithm for user i takes as input the parameter pp and selects an element $x_i \leftarrow \mathbb{Z}q$ and computes $y_i \leftarrow g^{x_i}$. It outputs the public key as $pk_i = (pp,\ y_i)$ and the secret key as $sk_i = (pp,\ x_i)$.
- **RS** $(sk,\ R,\ m)$. To sign the message m in the ring $R\ =\ (pk_1, ..., pk_n)$, the signer i with the secret key $sk_i = x_i$ generates the signature in the following way:
    1. For $j \in [n]$ and $j \neq i$, select $c_j, t_j \leftarrow \mathbb{Z}q$ and compute $a_j \leftarrow g^{t_j} y_j^{c_j}$ and $b_j \leftarrow H(m\|R)^{t_j}(H(m\|R)^{x_i})^{c_j}$.
    2. For $j\ =\ i$, select $r_i \leftarrow \mathbb{Z}q$ and compute $a_i \leftarrow g^{r_i}$ and $b_i \leftarrow H(m\|R)^{r_i}$.
    3. Let $c_i \leftarrow H'(m,\ R,\ \{a_j, b_j\}_1{}^n) - \Sigma_{j \neq i} c_j mod\ q$ and $t_i \leftarrow r_i - c_i x_i\ mod\ q$.
    4. Return $(R,\ m,\ H(m\|R)^{x_i}, c_1, t_1, ..., c_n, t_n)$.
- **RV** $(R,\ m,\ \sigma)$. On receiving the signature (R, m, σ), the verification algorithm first parses $\sigma\ as\ (r,\ c_1, t_1, ..., c_n, t_n)$ and checks if $\Sigma_1{}^n c_j = H'(m, R, \{G^{t_j} y_j^{c_j}, H(m\|R)^{t_j} r^{c_j}\}_1{}^n)$.

7. Conclusion

In this paper we have presented our approach to securing financial privacy through our combination of unique ring signatures, stealth addresses and non-interactive zero knowledge proofs. We've shown how our approach achieves the highest level of financial anonymity available at the time of publication without compromising the integrity of the Satoshi network's core principles: Unforgeability, Double-Spend prevention and trustless decentralization.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System" http://bitcoin.org/bitcoin.pdf, 2008

[2] "The Universal Declaration of Human Rights" http://www.un.org/en/documents/udhr/index.shtml, 1948

[3] Reuters – Emily Flitter, Stella Dawson, and Mark Hosenball, "U.S. to let spy agencies scour Americans' finances" http://www.reuters.com/article/2013/03/13/usa-banks-spying-idINDEE92C0EH20130313 , 2013

[4] The Wall Street Journal – Sionhan Gorman, Devlin Barrett and Jennifer Valentino-Devries, "CIA's Financial Spying Bags Data on Americans" http://www.wsj.com/articles/SB10001424052702303559504579198370113163530, 2014

[5] Ronald L. Rivest, Adi Shamir, and Yael Tauman "How to Leak a Secret" http://people.csail.mit.edu/rivest/pubs/RST01.pdf, 2001

[6] Eiichiro Fujisaki and Koutarou Suzuki, "Traceable Ring Signature *" https://eprint.iacr.org/2006/389.pdf, 2006

[7] Nicolas van Saberhagen, "CryptoNote v 2.0" http://cryptonote.org/whitepaper.pdf, 2013

[8] Ian Miers, Christina Garman, Matthew Green and Aviel D. Rubin, "Zerocoin: Anonymous Distributed E-Cash from Bitcoin" http://zerocoin.org/media/pdf/ZerocoinOakland.pdf, 2013

[9] Matthew Franklin and Haibin Zhang, "A Framework for Unique Ring Signatures", https://eprint.iacr.org/2012/577, 2012