IDENTITY CHAINS

ANDREW EGBERT: DIVBIT@MAIL.COM
BRADLEY CHUN: BRAD@BUTTERKISS.COM
THOMAS OTTE: THOMAS@BUTTERKISS.COM

Abstract. In this short technical summary, the authors describe how the mathematical primitives of Ring Confidential Transactions [Noe] may be used to provide anonymous identity authentication services in a similar manner to "Anonrep" [ZWC+16] but in a trustless (or permissioned), distributed manner, and with the additional security and resilience provided by a blockchain. The use of the mathematics in the RingCT paper additionally, and importantly, allows for combining different types of authentication in a seamless manner, in essence if the Bitcoin or Monero blockchain is a single "thread," then the protocol here allows one to "weave" such threads together. The resulting protocol is dubbed an "Identity Chain" and provides anonymous authentication working in a lightweight and interoperable manner between any number of different service providers running different identity chains.

Contents

1. Introduction

Recall the ring confidential transactions protocol in [Noe] provides a way to send money so that the sender, receiver, and amount in the transaction are hidden. This payment protocol also provides a trustless, uncompromisable setup,

which is a potential advantage over other attempts at an anonymous currency, such as ZeroCash [BSCG+14]. The new cryptographic technique in that paper which makes this possible is an Multilayered Linkable Spontaneous Anonymous Group Signature which we will use in this paper to create an interlacing system of chains providing anonymous, trustless (or permissioned) reputation and identity services, which can optionally be combined with payment tokens, such as one using the Ring CT protocol (e.g. Monero), or with traditional banking services. Anonymous authentication is desirable in a number of scenarios such as age verification (i.e. verify you are old enough without giving away your actual age), voting (i.e. vote for your candidate of choice without fear of repercussion), and governmental authentication (i.e. verify that you have a social security number without actually revealing it). In this paper we propose perhaps the first way to combine multiple forms of anonymous authentication provably linked to the same user and combinable with payment schemes.

2. Ring Confidential Transactions Recap

We briefly recall Ring Confidential Transactions verbatim as described in [Noe] which relies on public-key cryptography, for example, using an cryptographically safe elliptic curve, such as Curve25519 [saf, Ber06]. In a given transaction, the sender has a keypair

$$(x, P = xG)$$

consisting of secret key x a scalar in the underlying field) and public key $P = xG$ where G is the basepoint of the cryptographically secure elliptic curve group, and they also have a Pedersen Commitment to an amount:

$$C = aG + bH$$

with a another secret key, and b equal to the amount. The curve point H corresponds to a point whose discrete logarithm with respect to G is assumed to be unknown. (For instance $HashToPoint(G)$).

The sender of the transaction chooses a number of other pairs (P, C), not belonging to them, from the publically available pool of such pairs visible on the blockchain.

2.0.1. *Ring CT: Transaction Signing.* To sign a transaction, an MLSAG signature is constructed (as described in [Noe]) in the following manner:

- Let $\{(P_\pi^1, C_\pi^1), ..., (P_\pi^m, C_\pi^m)\}$ be a collection of addresses / commitments with corresponding secret keys $x_j,\ j = 1, ..., m$. Each pair (P_i, C_i), with $i \in \{1, ..., m\}$, corresponds to an input public key P_i and commitment C_i in a given group which satisfies the discrete logarithm assumption.
- Find $q + 1$ collections $\{(P_i^1, C_i^1), ..., (P_i^m, C_i^m)\}, i = 1, ..., q + 1$ which are not already tag linked in the sense of [FS07, page 6]. These will serve as additional inputs to the MLSAG to mask the actual input column.

- Decide on a set of output addresses $(Q_i, C_{i,out})$ such that $\sum_{j=1}^{m} C_{\pi}^{j} - \sum_i C_{i,out}$ is a commitment to zero.
- Let

$$\mathfrak{R} := \left\{ \left\{ \left(P_1^1, C_1^1\right), ..., \left(P_1^m, C_1^m\right), \left(\sum_{j=1}^{m} C_1^j - \sum_i C_{i,out}\right) \right\}, \right.$$
$$\ldots,$$
$$\left. \left\{ \left(P_{q+1}^1, C_{q+1}^1\right), ..., \left(P_{q+1}^m, C_{q+1}^m\right), \left(\sum_{j=1}^{m} C_{q+1}^j - \sum_i C_{i,out}\right) \right\} \right\}.$$

  be the key matrix which we wish to sign. Note that the last column is a Ring-CT ring in the sense of [Noe15, Section 4].
- Compute the MLSAG signature Σ on $\mathfrak{R}$ with respect to message $\mathfrak{m}$ which is a cryptographically secure hash of the set of $(Q_i, C_{i,out})$
- Compute range proofs, as described in [Max15, Noe] for each $C_{i,out}$.

In [Noe] proofs are given of the cryptographic security and anonymity of the sender.

2.0.2. *Ring CT: Transaction Verification.* Given a Ring-CT transaction generated as above, the transaction can be verified as generated correctly by simply verifying each of the signatures

- Verify the MLSAG part of the transaction with respect to the given message consisting of outputs and commitments $(Q_i, C_{i,out})$ including rejecting the transaction as duplicate if the key-image part of the signature appears as part of a previously received transaction.
- Verify each of the range proofs for the $C_{i,out}$.

3. Identity Chains

The Ring CT construction is essentially one possible currency facet of the multi-chain protocol which can consist of many identity chains and coexisting Identity Chains. This protocol removes the need for complicated scheduling protocols, and reduces anonymous authentication and reputation provision to identity combined with blockchain technology described in [Noe] which is uniquely suitable for such a reputation system. The basic idea is as follows: In the above Ring CT protocol, the point H is the result of recovering a point after taking a hash of some fixed value. The value H corresponds to one atomic unit of the underlying currency (Ring CT is in fact a part of the "Monero" Currency). Thus in the Pedersen Commitment, $C = aG + bH$, the point bH corresponds to b atomic units of the currency. Consider generalizing "currency" to more general "identity vouchers" provided by various services. For example, when a user authenticates by logging in with their account at somewhere such as Facebook or Twitter, they are essentially being "vouched for" by that provider. This voucher can be constructed by sending a "transaction" to $Hash(\text{"}Facebook\text{"})$ or $Hash(\text{"}Twitter\text{"})$ rather than to $H(G)$,

or more generally, if an expiring voucher is desired, a date can be included, such as $Hash(\text{"}Facebook\text{"} : \text{"}May\ 2020\text{"})$ (In fact, one can supply arbitrary JSON data as parameters). Using the mathematics provided by the above Ring CT process will provide provable anonymity guarantees, and the MLSAG signatures allow one to combine multiple validations into a single transaction where two or more chains cross. In contrast with the Ring CT protocol, in the Identity Chain protocol, we do not necessarily require linkability (although this can be used in certain cases), but note that instead, we can make certain types of "vouchers" non-transferrable (and thus re-usable).

3.1. **Issuing Vouchers.** In an "Identity Chain", most transactions, such as the one necessary for authenticating on a website, are not recorded on a central identity blockchain. Instead, blockchains are used by authentication providers such as, for example, the US Passport Agency and queried by services to verify that claimed vouchers actually exist. Vouchers are stored by users in a wallet, and authentication can be made offline via NFC, QR-code, or another such manner, as long as the verifier has access to either a stored copy of the relevant part of the blockchain, or is able to query the online blockchain.

Each user of an identity chain has a private keypair (a, b) and a public keypair $(A = aG, B = bG)$ with G the basepoint of a given cryptographically secure cyclic group. This paper doesn't fix a set method for these, however common choices could be ed25519 public and private keys as in [Noe]. To issue a voucher for service "service" which expires at time "expire time", an identity provider adds a transaction to their blockchain with destination

$$V = HashToScalar(rA)G + B + HashToPoint(\text{"}service : expire\ time\text{"})$$

where r is a random scalar generated using a cryptographically secure random number generator (this manner of creating P is the same as the CryptoNote [vS13] one-time addresses). This type of addressing prevents linking a users key to their vouchers, since the term $HashToScalar(rA)G$ is not included as part of the transaction. The identity provider includes $R = rG$ encoded as part of the transaction, and the recipient can recover the secret key to P in a similar manner as in [vS13], namely, they can check each transaction for their voucher by computing

$$HashToScalar(Ra)G + B + HashToPoint(\text{"}service : expire\ time\text{"})$$

Such transactions will occur as part of a request for a voucher from a user, or in response to a real-world payment, and the voucher recipient can therefore be notified out of band, by the respective identity provider, of which block in the blockchain the transaction occurs in. (Vouchers can alternatively be found by passively scanning each transaction if greater anonymity is desired).

3.1.1. *Age verification.* An alternative way to issue vouchers which have numerical content which should be kept confidential (such as someones age) is to issue a

voucher with a nonzero multiplier attached to the hashed JSON. For example, in the case of age, a vendor such as a hospital, could issue a voucher to

$$V = HashToScalar(Ra)G + B + (1986 - 1900)HashToPoint("Born")$$

and the user can verify their age by computing a range proof (in addition the the MLSAG) in the method of [Max15, Noe] to prove that the voucher is a commitment to a number in a certain range. (If the current year is 2016, then one could prove that the voucher is a commitment to a value in the range of $[0, 100]$, thus proving that the user is at least 16 years old and no more than 116 years old).

3.2. **Identity Chain transactions: Expenditure.** Suppose that a user wishes to authenticate for m reusable properties

- $H_1 = Hash("reusable - service_1 : expire - time_1")$
- $\cdots$
- $H_m = Hash("reusable - service_m : expire - time_m")$

using reusable vouchers $V_i^\pi, i = 1, ..., m$, and at the same time authenticate for n non-reusable properties:

- $P_1 = Hash("nonreusable - service_1 : expire - time_1")$
- ...
- $P_n = Hash("nonreusable - service_n : expire - time_n")$

using vouchers $W_i^\pi, i = 1, ..., n$.

For example, a user could wish to authenticate using the reusable properties of

$$Age\ over\ 18 :\ expires\ never$$

$$Resident\ of\ California :\ expires\ 2021$$

and one non-reusable property

$$Vote\ in\ Presidential\ Election :\ expires\ Nov.\ 2020$$

To use a set of vouchers $V_\pi^i, i = 1, ..., m$, a user chooses their desired anonymity level k, fixes π randomly as an element of $\{1, ..., k\}$, such that there should be only a 1 out of k chance of guessing the user for each voucher used and chooses, for each such voucher, k other vouchers $V_j^i, j = 1, ..., k$ from the given blockchain (for offline usage, these can be pre-selected and stored on a users mobile device) and creates an MLSAG, as described in [Noe] on the following key matrix:

$$\begin{pmatrix} V_1^1 - H_1 & V_1^2 - H_1 & \cdots & V_1^\pi - H_1 & \cdots & V_1^k - H_2 \\ V_2^1 - H_2 & V_2^2 - H_2 & \cdots & V_2^\pi - H_2 & \cdots & V_2^k - H_2 \\ & & & \vdots & & \\ V_n^1 - H_n & V_n^2 - H_n & \cdots & V_n^\pi - H_n & \cdots & V_n^k - H_n \\ \sum_i W_i^1 - \sum_i P_i & \sum_i W_i^2 - \sum_i P_i & \cdots & \sum_i W_i^\pi - \sum_i P_i & \cdots & \sum_i W_i^n - \sum_i P_i \end{pmatrix}$$

Resulting in a signature

$$\sigma = (I_1, ..., I_n, c_1, s_1^1, ..., s_m^1, s_1^2, ..., s_m^2, ..., ..., s_1^n, ..., s_m^n).$$

FIGURE 3.1. Crossing Identity Chains

For each non-reusable property, $H_1, ..., H_n$, there is either a distinct blockchain (for example, continuing the above example, there would be a blockchain dedicated to the presidential election in 2020 for voters who are residents of California), or a distinct blockchain for a group of related, but distinct, properties, determined by JSON values. This prevents bloat of any given identity chain. Given the above signature, as in [Noe], if the signature verifies (which is done using the usual MLSAG procedure described in [Noe]), the key-images I_j are added to the ledger corresponding to property H_j, and the second such key-image is rejected if it appears again (thus preventing double-voting). The verifier of the signature finally checks that each key V_i and W_i appears in the relevant identity chain itself.

4. Voucher Revocation

Voucher revocation happens when the issuer of a voucher on a chain desires to invalidate the voucher before its expiration date. This could be necessary for a number of reasons. For example, on an identity chain where the vouchers are public keys of SSL certificates accepted by a certain web browser (Note that blockchains specifically providing web-browser certificates exist, the original perhaps being dnschain [dns13] which was initiated in December 2013, although they were later explored in [FVY14]), the private key to the voucher could be stolen, which would represent a security risk to users of the web-browser, since, before the expiration date of the voucher, they could be tricked into visiting a website containing malware.

4.1. **Permissioned Identity Chain Voucher Revocation.** One way to do revocation in the case of a permissioned ledger is to use the Merkle tree approach

which Bitcoin [Nak08, Section 7] uses to compactify the blockchain. A given transaction block issuing vouchers will contain the vouchers at the lowest level of the tree, and their hashes are concatenated and hashed again to make the next level. Certificate revocation is accomplished by simply stubbing off branches of the tree, as if they were Bitcoin transactions being compactified. Using this method, the higher level hashes are kept, and the chain retains auditability, due to the one-way nature of cryptographic hash functions.

Vouchers which should be non-revocable shall be put on a permissionless chain.

4.2. **Permissionless Identity Chain Voucher Revocation.** On a permissionless identity chain, an anonymous revocable voucher can be issued by sending a token to a 1 out of 2 ring multisignature address such that either issuer or receiver can post the token to the identity chain. Creation of these ring multisignature addresses is discussed in detail in [SN16], and can be adapted to the above vouchers in the identity chain in the obvious manner.

The idea is essentially for each party in the multisignature to commit to a key (by proving knowledge of its corresponding private key), and then create shared keys as the sum of their commitments. In the case of vouchers, the voucher identifying point ($H(\text{"}property\text{"})$) is added after the shared key is created. Thus, given two users public keys, $A = aG$ and $B = bG$, a 1 out of 2 voucher will look like:

$$Hash(abG)G + Hash(\text{"}property\text{"}).$$

This can be done with one-time keys by setting A, and B to be the output of the one-time key creation scheme described in [vS13]. This can be simply extended to t out of n threshold vouchers using the techniques of [SN16].

Note that this method could also be used in the permissioned case, however it does take slightly more blockchain storage, and requires more work on the part of the end-user, since they must look for both the issuance and the revocation when deciding if a voucher is valid.

5. Combining Currency with Authentication

In some cases it may be desirable to combine a proof of identity with a payment. For example, a user in the United States may wish to combine a payment for an alcoholic beverage with a proof that their age is over 21. As another example, a member of a shopping loyalty club could combine their proof of membership in the club with a discounted payment.

To do this we can combine the Ring CT protocol, described in section 2 with the vouchers in the above section. We can slightly generalize the Ring CT protocol by taking H to be $HashToPoint(\text{"}CurrencyName\text{"})$ to allow for multiple currencies. So that a given commitment to amount "b" looks like

$$C = aG + bH(\text{"}CurrencyName\text{"})$$

Note that the vouchers of section 3 have no associated commitment to an amount and thus they do not change the final commitment to zero in the last row of the

Ring-CT MLSAG and hence can be added in an arbitrary row of an MLSAG without even verifying that they exist previously on the blockchain. (We assume that in the permissionless case, a per-kilobyte transaction fee is applied to prevent spamming transactions, such as in Monero). Denoting this commitment to zero by

$$C_i = \sum_j C^j_{in,i} - \sum_j C^j_{out,i},$$

the result is an MLSAG on a key matrix such as the following (using the notation of sections 2 and 3)

$$\begin{pmatrix} Pub^1_1 & Pub^2_1 & \cdots & Pub^\pi_1 & \cdots & Pub^k_1 \\ Pub^1_2 & Pub^2_2 & \cdots & Pub^\pi_2 & \cdots & Pub^k_2 \\ & & & \vdots & & \\ Pub^1_l & Pub^2_l & \cdots & Pub^\pi_l & \cdots & Pub^k_l \\ C_1 & C_2 & \cdots & C_\pi & \cdots & C_k \\ V^1_1 - H_1 & V^2_1 - H_1 & \cdots & V^\pi_1 - H_1 & \cdots & V^k_1 - H_2 \\ V^1_2 - H_2 & V^2_2 - H_2 & \cdots & V^\pi_2 - H_2 & \cdots & V^k_2 - H_2 \\ & & & \vdots & & \\ V^1_n - H_n & V^2_n - H_n & \cdots & V^\pi_n - H_n & \cdots & V^k_n - H_n \\ \sum_i W^1_i - \sum_i P_i & \sum_i W^2_i - \sum_i P_i & \cdots & \sum_i W^\pi_i - \sum_i P_i & \cdots & \sum_i W^n_i - \sum_i P_i \end{pmatrix}$$

6. Two-Way Identity Pegs

One additional, potentially useful, property of the above protocol is the capability to do anonymous two-way pegs between currency chains. This is useful in the permissioned scenario, but could potentially work in the case of trustless blockchains if mining is done simultaneously on both chains. Suppose we decide that currency A is worth 10 times the amount of currency B. Then we simply have to represent amounts in currency A as Pedersen Commitments $C = aG + bH$ and amounts in currency B by $C' = a'G + b'10H$. Given a set of blockchains, the miners (in the Federated case, miners just correspond to the identity or currency providers), one may create an MLSAG with vouchers on non-reusable rows, and the commitment sum on the reusable row, such as in the following key-matrix:

$$\begin{pmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,\pi} & \cdots & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,\pi} & \cdots & V_{2,n} \\ & & & \vdots & & \\ V_{m,1} & V_{m,2} & \cdots & V_{m,\pi} & \cdots & V_{m,n} \\ \sum_i C^1_i - \sum_i C'_i & \sum_i C^2_i - \sum_i C'_i & \cdots & \sum_i C^\pi_i - \sum_i C'_i & \cdots & \sum_i C^n_i - \sum_i C'_i \end{pmatrix}$$

where each C'_i represents a commitment to a value in chain B, and the destination addresses of this transaction would be keys in chain B. A given transaction, such as the above, exists on both chains, so long as the underlying cryptographic

group is the same (given the current crypto-graphic monoculture centered around Curve25519, this isn't a large hurdle).

Very important distinctions between the traditional interpretation of a two-way peg [vL16] and this two-way identity peg:

- This method does not anchor to a main chain.
- Instead of locking coins, the conversion coins are actually burned on chain A (spent to a generated address for which the private key is not known but is cryptographically linked to an address you own in chain B, via the vouchers mentioned above) and the consensus code agreed pegged equivalent are created on chain B and vice versa.
- The peg does not need to be fixed rate. I.e. it can be a calculated peg that floats with the coin supply and inflation of each chain.

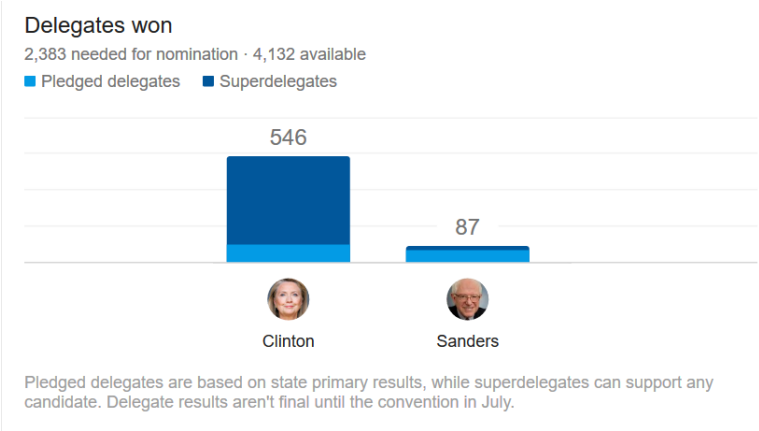

FIGURE 6.1. Superdelegates affecting U.S. election perception (Image from Google)

6.1. **Example use for two-way identity pegs.** In this example, we illustrate how to digitize an existing unequal voting system through the use of two way pegged identity chains. For instance one might have SuperDelegates whose votes are worth 10 times that of Delegates, and Delegates whose votes are worth 20 times that of an ordinary disenfranchised voter. These parties would respectively be issued commitments to

$$aG + 100H$$
$$a'G + 10H$$
$$a''G + .5H$$

and would then vote by sending their votes to a counter who could sum the total commitments received and produce a total cryptographically verifiable vote count without revealing who, out of super-delegates, delegates, and the disenfranchised voters, voted for a given candidate.

7. Conclusion

Identity chains are a way to authenticate for a variety of properties in a cryptographically secure and trustless manner, backed up by the security provided by Satoshi Nakomoto's blockchain technology [Nak08] and utilizing the ring confidential transactions protocol of [Noe]. Identity chains have a variety of use cases such as voting, authenticated payments, age verification, anonymously verifying logins, and do so in a way which is cryptographically simpler and more secure than previous attempts. Both permissioned, highly scalable identity chains are possible, and trustless bitcoin-like identity chains are possible (in fact Monero, with the recent inclusion of Ring Confidential transactions could be perhaps categorized as the simplest example), and this paper provides a way for many such chains to coexist, or "weave" together, hopefully bridging the gap from individual segregated blockchains, to a more general paradigm.

References

[Ber06] Daniel J Bernstein. Curve25519: new diffie-hellman speed records. In *Public Key Cryptography-PKC 2006*, pages 207–228. Springer, 2006.

[BSCG+14] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 459–474. IEEE, 2014.

[dns13] Dnschain. https://github.com/okTurtles/dnschain, 2013.

[FS07] Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In *Public Key Cryptography–PKC 2007*, pages 181–200. Springer, 2007.

[FVY14] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. Certcoin: A namecoin based decentralized authentication system 6.857 class project. 2014.

[Max15] Greg Maxwell. Confidential Transactions. https://people.xiph.org/~greg/confidential_values.txt, 2015. [Online; accessed 1-June-2015].

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1(2012):28, 2008.

[Noe] Shen Noether. Ring signature confidential transactions for monero. http://eprint.iacr.org/2015/1098.

[Noe15] Shen Noether. Mininero. https://github.com/ShenNoether/MiniNero, 2015.

[saf] Safecurves: Choosing safe curves for elliptic-curve cryptography. https://safecurves.cr.yp.to/.

[SN16] Adam Mckenzie Shen Noether. Ring multisignature. https://shnoe.wordpress.com/2016/03/22/ring-multisignature/, 2016.

[vL16] Sergio D van Lerner. Drivechains, sidechains and hybrid 2-way peg designs. https://uploads.strikinglycdn.com/files/27311e59-0832-49b5-ab0e-2b0a73899561/Drivechains_Sidechains_and_Hybrid_2-way_peg_Designs_R9.pdf?id=27476, 2016.

[vS13] Nicolas van Saberhagen. Cryptonote. https://cryptonote.org/whitepaper.pdf, 2013.

[ZWC+16] Ennan Zhai, David Isaac Wolinsky, Ruichuan Chen, Ewa Syta, Chao Teng, and Bryan Ford. Anonrep: towards tracking-resistant anonymous reputation. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 583–596, 2016.