Partial Linkable SAG (PLSAG) signatures

authorYamamoto Haruto,[1,*] Chen-Mou Cheng,[1] and Masahiro Mambo[1]

[1] *Kanazawa University, School of Information and Communication Engineering, Information Security Lab*

[*] *haruto.y0327@gmail.com*

Abstract: aaaaaaaaaaaaaa

1. Introduction

2. Related Work

CLSAG signatures use a multilayer key set, and improve version of MLSAG in Monero[8]. The signer has m private keys in the key set, but the only first key image of the first private key has linkability, and the other key images of the other private keys called "Auxiliary key images" do not have linkability. actually, the Key images from the only first private are necessary for Monero on MLSAG, so linkability is removed from the other key images.It makes the size of the CLSAG signature reduced compared with MLSAG signatures, since CLSAG signatures only includes $c_1$and n random numbers, but MLSAG Signatures includes c_1 and $n \cdot m$ random numbers.Currently, CLSAG Signatures are used as the ring signatures on Monero.
Triptych is also a Linkable Ring Signature based on zero-knowledge proofs without a trusted setup, but the strong advantage is that the signature size increases with logarithmic size [3]. The triptych will officially be implemented on Monero in near future. Triptych applies Pedersen Commitment and sigma protocol and uses a new way to aggregate public keys on Sigma protocol to construct logarithmic sized Linkable ring signature.

3. Proposed Method and Algorithm

3.1. Partial Linkable SAG (PLSAG) signatures

In this section, PLSAG(Partial Linkable SAG) signatures are introduced.
In Monero, a ring signature is used to prove that transactions are legitimate while the amount of money is kept secret. In this study, we propose an algorithm for PLSAG (Partial Linkable Spontaneous Anonymous Group) signatures, which is an improvement of the CLSAG signature, a ring signature currently implemented on Monero. Ring signatures have the ability that the verifier can not identify who among the ring members created the signature. PLSAG signatures have the characteristics of ring signatures, such as Unforgetability and Signer Ambiguity, and the ability to add linkability to the key images of the odd-numbered private keys in the $(n \cdot m)$key set. Linkability means that if the key image generated by the private key does not match any other key images generated on Monero in the past, it can be confirmed that the private key has not been used and a double-spending attack can be prevented.

• The advantages compared to CLSAG signatures
In CLSAG signatures, Linkability can be added to the only first key images of the first private key in the $(n \cdot m)$key set, but in PLSAG signatures, we consider adding Linkability to odd-numbered private keys in the $(n \cdot m)$key set. In our algorithm PLSAG, the number of random numbers is reduced compared to CLSAG by aggregating the public keys and key images.
CLSAG signatures require as many signatures as there are inputs, so the size of CLSAG signatures increases linearly with the number of inputs. In Monero, 2-CLSAG is used where the first line is the private keys of the transaction and the second line is used to prove the validity of the amount remittance. However, we designed PLSAG like one PLSAG signature is sufficient for one transaction regardless of the number of inputs. First, PLSAG signatures use 2m-CLSAG, with the private keys of the transactions on the odd-numbered lines and the validity of the amount of remittance on the even-numbered lines. Furthermore, it can prevent double-spending attacks by adding linkability to the odd-numbered lines. By extending the $(n \cdot m \cdot 2)$public key set to multiple m inputs and aggregating the public key set and key images, then PLSAG signatures can reduce the number of random numbers.

3.2. Algorithm of PLSAG

This section explains the Algorithm of the PLSAG signatures scheme. Figure 1 illustrates how to compute c_i following the ring. In this algorithm, we consider that there are three transactions for the input.

- **Initial Condition on Monero**

n: the number of anonymity TX. 3: the number of inputs. R: Ring $K_{i,j}[K_{1,1}, K_{1,2}, \ldots, K_{n,6}]$
$K_{i,j}$: Public key. $k_{i,j}$: Private key. $(K_{i,j} = k_{i,j}G)$
$\mathscr{H}_n$: hash function to map to integer in the range 0 to l-1. $\mathscr{H}_p$: hash function to map to curve point.

- **Signing Algorithm**

1. Calculate Key Images$\tilde{K}_{s,j} = k_{\pi,j}\mathscr{H}_p(K_{\pi,s})$ for s=1,3,5, j=1,...,6.
2. Generate random $\alpha, r_1, \ldots, r_n$ except r_π.
3. Calculate aggregate public keys, key images, public keys.

$$W_i = \sum_{j=1}^{6} \mathscr{H}_n(T_j, R, \tilde{K}_{1,1}, \ldots, \tilde{K}_{5,6}) * K_{i,j}$$

$$\tilde{W}_1 = \sum_{j=1}^{6} \mathscr{H}_n(T_j, R, \tilde{K}_{1,1}, \ldots, \tilde{K}_{5,6}) * \tilde{K}_{1,j}$$

$$\tilde{W}_2 = \sum_{j=1}^{6} \mathscr{H}_n(T_j, R, \tilde{K}_{1,1}, \ldots, \tilde{K}_{5,6}) * \tilde{K}_{3,j}$$

$$\tilde{W}_3 = \sum_{j=1}^{6} \mathscr{H}_n(T_j, R, \tilde{K}_{1,1}, \ldots, \tilde{K}_{5,6}) * \tilde{K}_{5,j}$$

4. Compute $c_{\pi+1} = \mathscr{H}_n(T_c||R||m||\alpha G||\alpha\mathscr{H}_p(K_{\pi,1})||\alpha\mathscr{H}_p(K_{\pi,3})||\alpha\mathscr{H}_p(K_{\pi,5}))$
5. Compute for $i = \pi+1, \pi+2, \ldots, n, 1, 2, \ldots, \pi-1$, replacing $n+1 \to 1$
   $c_{i+1} = \mathscr{H}_n(T_c||R||m||r_iG + c_iW_i||r_i\mathscr{H}_p(K_{i,1}) + c_i\tilde{W}_1||r_i\mathscr{H}_p(K_{i,3}) + c_i\tilde{W}_2||r_i\mathscr{H}_p(K_{i,5}) + c_i\tilde{W}_3)$.
6. Define $r_\pi = \alpha - c_\pi w_\pi$ where $w_\pi = \sum_{j=1}^{6} \mathscr{H}_n(T_j, R, \tilde{K}_{1,1}, \ldots, \tilde{K}_{5,6}) * k_{\pi,j}$
   The signature is $\sigma(c_1, r_1, \ldots, r_n)$ with key images $\tilde{K}_{1,1}, \ldots, \tilde{K}_{5,6}$.

- **Verifying Algorithm**

1. Check all Key Images $l\tilde{K}_{i,s} = 0$.
2. Calculate aggregate public keys, key images.
3. Compute for $i = 1, \ldots, n$, replacing $n+1 \to 1$
   $c'_{i+1} = \mathscr{H}_n(T_c||R||m||r_iG + c_iW_i||r_i\mathscr{H}_p(K_{i,1}) + c_i\tilde{W}_1||r_i\mathscr{H}_p(K_{i,3}) + c_i\tilde{W}_2||r_i\mathscr{H}_p(K_{i,5}) + c_i\tilde{W}_3)$.
4. If $c'_1 = c_1$ then the signature is valid.

3.3. Correctness

This chapter checks if PLSAG Signature works well or not and proves why it works. To speak simple, comparing c_i between Signing and Verifying algorithm makes this signature works well from a cryptographic aspect.
• If $i \neq \pi$ then, clearly the values $c'_{i+1} = c_{i+1}$, because the verifying algorithm uses same c_1 at the beginning of start value and same calculation algorithm. • If $i = \pi$ then, since $r_\pi = \alpha - c_\pi w_\pi$

$$r_\pi G + c_\pi W_\pi = (\alpha - c_\pi w_\pi)G + c_\pi W_\pi = \alpha G - c_\pi w_\pi G + c_\pi W_\pi = \alpha G$$

And,

$$r_\pi\mathscr{H}_p(K_{i,1}) + c_\pi\tilde{W}_1 = (\alpha - c_\pi w_\pi)\mathscr{H}_p(K_{i,1}) + c_\pi\tilde{W}_1 = \alpha\mathscr{H}_p(K_{i,1}) - c_\pi w_\pi\mathscr{H}_p(K_{i,1}) + c_\pi\tilde{W}_1 = \alpha\mathscr{H}_p(K_{i,1})$$

Because of $w_\pi G = W_\pi$,$w_\pi\mathscr{H}_p(K_{i,1}) = \tilde{W}_1$, $w_\pi\mathscr{H}_p(K_{i,3}) = \tilde{W}_2$,$w_\pi\mathscr{H}_p(K_{i,5}) = \tilde{W}_3$.
Therefore, it is also clear to find $c_{\pi+1} = c'_{\pi+1}$.

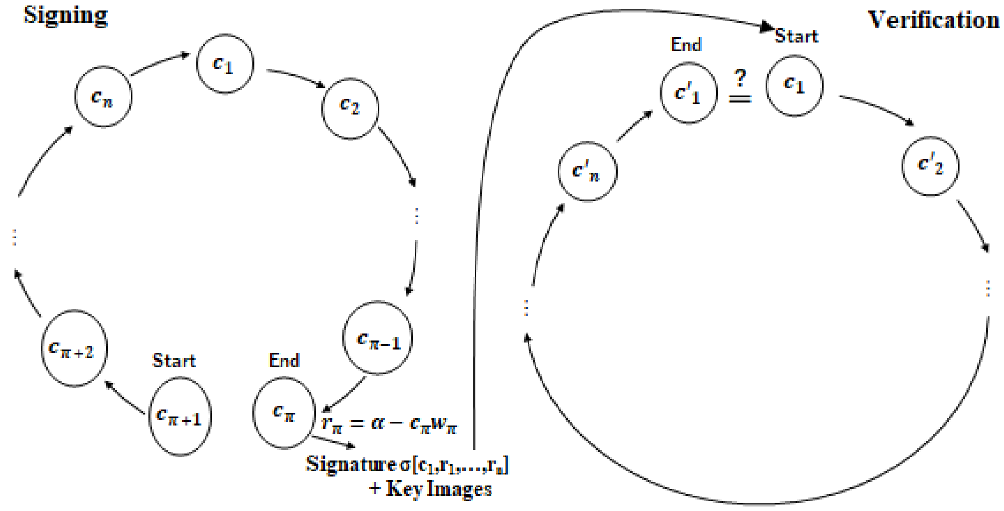

Fig. 1. PLSAG Algorithm

4. Results and Comparison

Table 1 compares signature size between existing LSAG and proposed PLSAG. n is the number of anonymity set, and m is the number of inputs. The number of the random numbers is used for calculation ring signatures, and the number of Key Images is used for avoiding Double spending. Basically, the sum of the size of random numbers and key images is the signature size. Firstly, the size of the CLSAG signature is perfectly smaller than that of the MLSAG signature. Secondly, the CLSAG signature is smaller than the PLSAG signature for the number of key Images but larger than the PLSAG signature for the number of random numbers.

The signature size of PLSAG with existing linkable ring signatures (MLSAG and CLSAG) and Triptych, which will be implemented in Monero, is compared with N Anonymity set size and 3 inputs, in Figure 2. For large anonymity set size ($N > 64$), the signature size of Triptych is the smallest, and PLSAG is smaller than that of MLSAG and CLSAG. On the other hand, for small anonymity set size ($N < 64$), the signature size of PLSAG is smaller than that of Triptych.

Figure 3 illustrates signature sizes for four LSAG with 10 Anonymity set sizes and M inputs. It is obvious that the signature size of Triptych is smaller than that of CLSAG and MLSAG in all ranges. The signature size of PLSAG is the smallest among 4 LSAG for the small number of input($M < 5$), however that of PLSAG increases with the square of M. Thus, the signature size of PLSAG is larger than other signatures for the large number of input($M > 10$).

Table 1. size and signature

Ring Signature	Random Numbers (F)	Key Images (G)
MLSAG	$(2n+1)m$	$2m$
CLSAG	$(n+1)m$	$2m$
PLSAG	$n+1$	$2m^2$
Triptych	$(\lg(n)+3)m$	$(2\lg(n)+6)m$

5. Discussion

Anonymity set size is often around 10 in Monero, and thus it suggests that PLSAG is able to reduce the signature size the most in the limited range. comparing between CLSAG and PLSAG, focusing on figure1 and table1, the reason why the signature size of CLSAG is smaller than that of CLASG is that the size of key images in PLSAG increases with the square of m, and the size of random numbers in CLSAG increases with the square of n, since basically the number of anonymity set is much larger than the number of inputs. However, if there are inputs to one transaction, it becomes a big disadvantage for PLSAG signatures. According to table 1, it is clear that

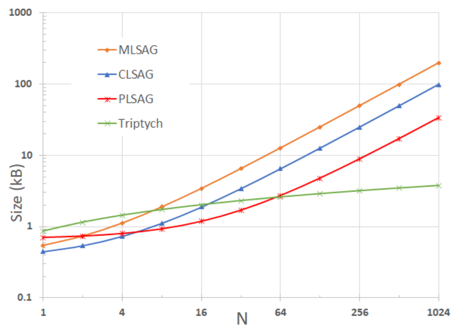

Fig. 2. anonymity set size N with 3 inputs

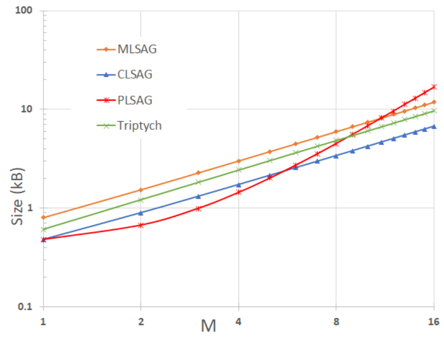

Fig. 3. 10 anonymity set size with M Input

the size of CLSAG and PLSAG are the same size in the case that there is only one input to the transaction. In this study, PLSAG, which gives Linkability to odd-numbered rows, is found to be an intermediate relationship between CLSAG, which gives Linkability only to the first row, and MLSAG, which gives Linkability to all rows. In addition, PLSAG can add Linkability to any row, such as even-numbered rows and the first 3 rows, with only a slight modification of the proposed algorithm. However, as the number of rows to which linkability is to be added increases, the required key image space also increases.

The size of Triptych is efficient even if there are many inputs and a large anonymity set size, on the other hand, Triptych has the disadvantage that the computational complexity is more expensive than the other signatures due to a large number of multi-scalar multiplications with the assumption that multi-scalar multiplications need more computational power than hash calculations.

6. Summary

We propose a new signature algorithm, PLSAG, for Monero's ring signatures which can reduce the signature size in a limited range ($6 < N < 64$) compared to the existing ring signatures, every transaction need one PLSAG signature regardless of the number of inputs, since PLSAG signatures can add linkability to odd-numbered key images of the private key in the public key set. But the size for key images increases with the square of M and the size of signatures also increases linearly with anonymity set size, so PLSAG signatures have a demerit in the case that there are many inputs to the transaction. In future work, I would like to investigate an improved PLSAG that the signature size increases with logarithmic sized without limited range by reference to aggregation techniques in Triptych.

References

1. KOE, Kurt M. Alonso, Sarang Noether, "Zero to Monero: Second Edition", April 2020.
2. Sarang Noether, Brandon Goodell, "Triptych: logarithmic-sized linkable ring signatures with applications", DPM/CBT@ESORICS 2020, 2020.
3. Tsz Hon Yuen, Shifeng Sun, Joseph K. Liu, Man Ho Au, Muhammed F. Esgin, Qingzhao Zhang, Dawu Gu, "RingCT 3.0 for Blockchain Confidential Transaction: Shorter Size and Stronger Security ", Financial Cryptography 2020, 2020.
4. Benedikt Bunz, Jonathan Bootley, Dan Bonehz, Andrew Poelstrax, Pieter Wuille, and Greg Maxwellk "Bulletproofs: Short Proofs for Confidential Transactions and More ", IEEE Symposium on Security and Privacy 2018, IEEE, pages 315-334, 2018.
5. Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen "A survey on the security of blockchain systems ", Future Generation Computer Systems Volume 107, Elsevier, Pages 841-853, June 2020.
6. Shen Noether, Adam Mackenzie, and Monero Core Team "Ring Confidential Transactions ", Cryptology ePrint Archive, Report 2015/1098, 2015
7. Brandon Goodell, Sarang Noether, and Arthur Blue "Concise Linkable Ring Signatures and Forgery Against Adversarial Keys ", Cryptology ePrint Archive, Report 2019/654, 2019