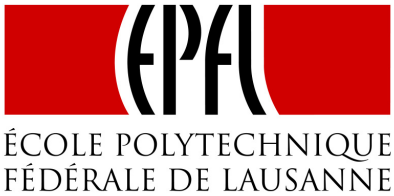

Distributed Identity Based Short Linkable Ring Signature

Kasra EdalatNejadKhamene

School of Computer and Communication Sciences
Ph.D. Semester Project

Prof. Bryan Ford
DEecentralized and DIstributed Systems

January 2018

Abstract

Linkable Ring Signature is a very valuable method to provide a combination of anonymity and accountability. The major drawback of LRS is that the size of signature grows linearly with the size of anonymity set. This prevents LRS to scale to large systems like a big election. In this report, we use an accumulator to create a constant size LRS. Using accumulator forms a trusted authority for the signature. We use distributed trust and secret sharing to handle this issue. Distributed Identity-Based Short Linkable Ring Signature provides a short signature without relying on a central authority.

Introduction

The demand for privacy and anonymity is growing every day. The need for privacy ranges from simple matters like posting in a forum or updating a wiki to topics of national interest like the presidential election. Service providers need to detect and prevent misbehavior. Hence most of them are not willing to provide anonymity without having some accountability. This accountability can take different forms. Some service providers want to ensure that only a set of authorized users can do specific actions. Another form of accountability is the ability to link different actions of a user together which we call linkability. This can reduce the anonymity of the user or result in deanonymization but in some cases like an election we need to know whether a user has voted more than once and having full anonymity for an honest user is neither useful nor practical.

One of the methods to provide both anonymity and accountability is Linkable Ring Signatures(LRS). LRS has been broadly studied and used in both academia and industry. One of the significant drawbacks of the LRS is that the size of the signature linearly increases with the size of the anonymity set (ring). The size of the signature has a negative impact on the usability of LRS and limits the size of the anonymity set and the number of signatures that a system can support. For example, we cannot use LRS for a big election. The necessary time and memory for an LRS can limit its use cases to a small system and prevent broad acceptance due to performance issues. In this report, we try to improve the robustness and usability of accountable anonymous systems.

Accumulators provide an opportunity to securely show membership in a set with constant size. Therefore, we can use them to create Short Linkable

Ring Signatures(SLRS). Having short signatures can reduce the communication cost and memory size which leads to better scalability. In this project, we study the various construction of accumulator and how we can use them to improve the usability of LRSs.

Our primary goal here is usability and reducing size. Therefore, we mostly focus on approaches based on elliptic curves over RSA approaches.

1 Accumulator

1.1 definition

Benaloh et al.[1] introduced the accumulator in 1994. Here we use Mashatan and Vaudenay[2] formal definition of accumulator.

An accumulator is a set of algorithms, a domain for the input D, and a set of values $X \subseteq D$. Accumulator represents the set X as one element in D

$Setup(i^k) \rightarrow (K_s, K_p)$: Setup algorithm gets a security parameter k and generates a secret key for the authority and a public key for the accumulator. Some accumulators may not need the trapdoor. In this case the secret key will be empty $K_s = \perp$. We will mention the secret for operations which need it in a trusted setup. If there is no trusted autority the K_s will be empty.

$Accumulate(K_s, K_p, X) \rightarrow V$: accumulates all members to an element V.

$WitnessGeneration(K_s, K_p, V, x, X) \rightarrow W_x$: generates a proof of membership W_x for user x.

$Verify(K_p, x, W_x, X)$: checks the proof for x.

A dynamic accumulator define two additional algorithms to allow efficient adding/removing members to the set.

$UpdateSet(K_s, K_p, V, x, op) \rightarrow (V', record)$ where $op \in \{add, remove\}$: adds or removes an element to the set. This algorithm writes a record of this operation to allow efficient witness update in WitnessUpdate.

$WitnessUpdate(K_p, x, W_x, V, x', op)$ where x' is the chnaged elemelt and $op \in \{add, remove\}$: Using the record we update the witness for element x. This operation should not need the secret key. Hence, every client can update his own witness with downloading a list of new records.

There are two major categories of accumulators: RSA based, Bilinear Pairing(BP) based. We provide a brief introduction of BP accumulators then we compare different accumulators.

1.2 Bilinear pairing accumulator

Nguyen[3] proposed the first accumulator based on bilinear pairing. We provide a brief description of the accumulator here. For further information refer to [3].

$Setup(1^k) \rightarrow (e, G_1, G_2, P, u, K_s = s, K_p = (P, sP, s^2P, ..., s^qP))$: We generate the pairing $e(G_1, G_1) \rightarrow G_2$ and point $P \in G_1$. Here s is the secret key, $u \in \mathbb{Z}_p$ is a random number, and q is the upper bound on the nmber of accumulated elements. We need a trusted setup to generate s and K_p. The auxilary information s can be used to improve the efficiancy of the accumulator or we can delete it after the setup. Users can publicly verify K_p using $e(P, s^qP) = e(sP, s^{q-1}P) = e(s^2P, s^{q-2}P) = ...$

$Accumulate(K_s, K_p, X) \rightarrow V$: for set $X = \{x_1, x_2, ...x_n\}$ we compute the accumulation as $V = \prod_{i=1}^{n} (x_i + s)uP$. We can directly compute this with the knowledge of auxilary information in linear time. If we don't have access to the auxilary information we can write V as a polynomial based on s^iP and we can compute it using the public informatio K_p.

$WitnessGeneration(K_s, K_p, V, x, X) \rightarrow W_x$: witness generation is similar to accumulate. For element x_j the witness is $W = \prod_{i=1 \wedge i \neq j}^{n} (x_i + s)uP$.

$Verify(K_p, x, W, X)$: accept if $e(W_x, xP + sP) = e(V, P)$. The value of $e(V, P)$ is common for all elements and we can publish it to reduce the number of necessary pairing for verifying.

$UpdateSet(K_s, K_p, V, x, op) \rightarrow (V', record)$: We need to have access to the auxilary information to do this operation efficiently. Without the auxilary information we need to compute everything from scrath and we lose dynamicity. Adding: $V' = (x + s)V$. Removing: $V' = \frac{1}{x+s}V$

$WitnessUpdate(K_p, x, W, V, x', op)$: We can update the witness using published record regardless of having auxilary information. We use folowing formula to update the new witness W'_x. Adding: $W'_x = V + (x' - x)W$. Removing: $W'_x = \frac{1}{x'-x}(W - V')$.

The major problem of the Nguyen's accumulator is the authority. Without the authority, the accumulator is no longer dynamic, and we need to spend quadratic time with respect to the number of members to accumulate or generate a witness. Furthermore, we have to use a trusted setup to generate the public key.

1.3 Integraating with Merkle tree

Our first goal was to integrate an accumulator with a Merkle tree to provide short proofs. This would increase the robustness and dynamicity of the accumulator and allow faster witness updates. Users would no longer need to follow the record updates and spend time linear to the number of changes to update the witness. To enable this, we need two properties: union, no trusted authority.

A Merkle node is the union of its children. Hence, we require to compute the union of node's childs to add an accumulator to each node. Merkle tree is publicly verifiable and doesn't need a trusted authority to operate. The accumulator should not add a trusted authority. We can reduce this requirement to allow a distributed trust, but a central trust is not acceptable.

For both RSA and BP accumulators, the union of two accumulators $V_1 = xP$ and $V_2 = yP$ would be in the form of $V_{union} = xyP$. Unfortunately, this is the Computational Diffie-Hellman (CDH) problem which is assumed to be hard. On the other hand, storing or using x or y would remove the discrete logarithm problem from the accumulator, which makes generating a fake proof easy for someone with this knowledge. We cannot integrate eaither RSA or BP accumulators with a Merkle tree.

1.4 Comparision

In this section we provide a comparision between different accumulators in table 1. The full report would be available on [4].

A universal accumulator can generate proof of non-membership in addition to the proof of membership. A $q - bounded$ accumulator can have at most q members.

A week dynamic only support adding members and cannot remove a member efficiently. An auxilary dynamic needs secret key to efficiently add and remove. An Eff authority needs auxilary info for efficiency, but it can operate without it.

Scheme	Year	Model	Bounded	Authority	Dynamic	Union
Benaloh et al.[1]	1994	RSA	No	No	No	No
Camenisch et al.[5]	2002	RSA	No	Yes	weak	No
Dodis et al.[6]	2004	RSA	No	Yes	weak	No
Lan Nguyen[3]	2005	BP	Yes	Yes	Yes	No
Li et al.[7]	2007	RSA	No	Eff	Yes	No
Au et al.[8]	2009	DH/BP	Yes	Eff	Yes	No
Camenisch et al[9].	2009	BP	Yes	Yes	Yes	No
Mashatan et al.[2]	2013	RSA	No	Eff	Yes	No
Miers et al.[10]	2013	RSA	No	Yes	Yes	No
Ghosh et al.[11]	2016	BP	No	Yes	Yes	No
Sun et al.[12]	2017	DH/BP	Yes	Eff	Yes	No
Kuo et al.[13]	2017	RSA	No	Yes	Yes	No

Table 1: Accumulator comparision

2 Short Linkable Ring Signature

2.1 Comparision

We can use an accumulator to create a Short Ring Signature. In all of the following schemes, the accumulator and the signature use the same cryptosystem. Hence, we categorize the SLRS to RSA based and Bilinear Pairing based. We will provide a brief introduction to the RSA based approaches then we give a more detailed explanation of BP based approaches. Finally, we provide a comparison in table 2. The full report would be available on [14].

The RSA schemes use an RSA accumulation to achieve the small size and they show a big number's factorization to prove the knowledge of the private key. In this approach, a maliciously created public key can create multiple linkage tags. Therefore, RSA based approaches either use a trusted authority to generate public/private keys or use a Certificate Authority(CA) to ensure the correctness of the public key. Elliptic curves and bilinear pairing can provide the same security as RSA with fewer bits. Considering the drawbacks mentioned above, we focus on BP based approaches.

A Non-Authority anonymity scheme is anonymous against ordinary users, but the authority can deanonymize it.

In Zhang et al.[20] the number of necessary pairing grows linearly with

Scheme	Year	Model	Size	Linkable	Anonymity	Bounded	Authority
Tsang et al. [15]	2005	RSA	$O(1)$	Yes	Non-Authority	No	Yes
Au et al. [16]	2006	RSA	$O(1)$	Yes	Yes	No	Yes
Au et al. [17]	2013	BP/IBC	$O(1)$	Yes	Non-Authority	Yes	Yes
Chow et al. [18]	2006	BP/IBC	$O(1)$	Yes	Non-Authority	Yes	Yes
Wu et al. [19]	2006	RSA	$O(1)$	No	Yes	No	No
Zhang et al.[20]	2004	IBC	$O(\log(n))$	No	Yes	No	No

Table 2: SLRS comparision

respect to the size of the signature($\log(n)$), which has a negative impact on performance.

2.2 Bilinear pairing SLRS

Nguyen[3] proposed a Short Ring Signature alongside his accumulator. In this ring, the public keys are based on Sakai-Kasahara Identity Based Cryptosystem (SK-IBC)[21]. SK-IBC is efficient and Chen et al.[22] proved it secure in 2005. Afterward, Chow et al.[18] expands this Ring Signature with a linkage tag to create a Short Linkable Ring Signature.

$Setup(1^k)$: We initialize the Accumulator $Acc = (G_1, G_2, e, P, K_s = sk_{acc}, K_p = pk_{acc})$ and the Identity based system $SK{-}IBC = (G_1, G_2, e, H, Q, Q_{pub} = sk_{IBC}Q, K_s = sk_{IBC})$ where H is a collision-free hash function, Q is a point on G_1, and sk_{IBC} is the master secret key of IBC.

$MakeRingPubKey(X = \{id_i\}_{i=1}^n) \rightarrow RPK$: we accumulate all ids to generate the ring's public key. $RPK = Accumulate(\{H(id_i)\}_{i=1}^n)$

$MakeRingSecKey(id, X) \rightarrow RSK_{id}$: the secret key includes the witness in the accumulator and the SK-IBC's private key. $RSK_{id} = (h_{id} = H(id), R_{id} = \frac{1}{h_{id}+sk_{IBC}}Q, W_{id} = WitnessGeneration(X, h_{id}))$

$Sign(RPK, RSK_{id}, M) \rightarrow \sigma$: where M is the message and σ is the signature. We use a Non-Interactive Zero Knowledge procedure to prove the folowing properties and form the signature. For further information check [18].

$$e(h_{id}Q + Q_{pub}, R_{id}) = e(Q, Q)$$

$$e(h_{id}P + pk_{acc}, W_{id}) = e(V, P)$$

$$Link = e(R_{id}, tag)$$

$Verify(M, RPK, \sigma) \rightarrow \{Accept, Reject\}$: check the correctness of zero knowledge properties.

The Nguyen Signature is based on an Identity Based approach and relies on a trusted authority. Due to IBC inherent key escrow problem, not only the authority can deanonymize the users, but also it can forge signatures.

3 Secret Sharing

Shamir [23] has introduced the secret sharing in 1979. We choose an arbitrary polynomial p of degree $t-1$ and take $p(0)$ as the secret. The user i receives point $(i, p(i))$ as his secret share. Having t share, one can interpolate the polynomial and compute the secret $p(0)$. Nobody can determine the polynomial with $t-1$ share or less. We call this a (t,n)-threshold cryptosystem.

In Shamir's algorithm a dealer chooses the polynomial and shares the secret. Hence, the dealer knows the secret. In many use cases, nobody should know the secret. In these cases, we use a dealerless approach or a Distributed Key Generation (DKG) to share the secret. In a DKG, all of the users collaborate to generates the secret shares and none of them will know the secret.

3.1 Direct

We want to compute sP from P where s is the secret key, and we do not want to have a trusted authority. This problem has been studied thoroughly and we can solve it by using a secret sharing algorithm in a distributed trust setting. The Feldman [24] and Pedersen [25] are known algorithms for this problem.

3.2 Inverse

We want to compute $\frac{1}{s+x}P$ from P where s is a secret key, x is an arbitrary value, and we do not want to have a trusted authority. This problem is more complicated than the previous one, and we can no longer solve it with a simple secret sharing. We need to use a Secure Multiparty Computation between servers to compute the inverse for each request. Using inverse secret sharing requires having at least t servers to collaborate on each request. Hence, it

is not efficient, and the computation and communication cost is higher than the direct secret sharing. Kate[26] and Geisler et al.[27] provide two solution for this problem.

4 Distributed signature authority

We used an accumulator to reduce the size of LRS, but it led to a trusted authority. In many cases, we cannot have a trusted authority. For example, in a presidential election, the government needs to show that it has not meddled in the result or change/fabricate votes. Here we propose using a distributed trust and share the secret between a set of servers.

4.1 Distributed Identity-Based Cryptosystem

In SK-IBC the identity authority has a secret key sk_{IBC}. This key is used for computing users private key $R_{id} = \frac{1}{sk_{IBC}+h_{id}}Q$. We can share this key using a (t,n)-threshold secret sharing to distribute the trust.

Using a direct distributed key generation, we cannot compute the $\frac{1}{Sec+x}$, which is required for generating user's private key. Hence, we need to use an inverse DKG. We can use either Kate[26] or Geisler et al.[27] to distribute the SK-IBC.

4.2 Distributed Accumulator

The Nguyen's accumulator requires either a trusted setup or a trusted authority to work properly. This accumulator is dynamic and efficient with authority, but without authority, it is not dynamic, and it requires quadratic time for each operation. We are using a distributed trust for the SK-IBC. Since we are using a distributed trust to form the signature, it will not be a heavy burden to use it for the accumulator too. Here, we study three level of trust in the accumulator and compare them.

4.2.1 Trusted setup

We use a direct DKG to create a secret s and generate the public key $(P, sP, s^2P, ..., s^qP)$. Afterward, we delete all shares of the secret s. We no longer can increase the bound q to have more than q members in the ring, and all operations would require quadratic time.

4.2.2 Weak dynamic

We run a trusted setup, but we will not delete the secret share. A threshold of t servers can compute sQ for an arbitrary Q. Hence we can increase the upper bound whenever we want. Additionally, a threshold of t servers can accumulate, and generate witness in linear time and dynamically add a new user in constant time.

4.2.3 Full dynamic

We use an inverse secret sharing in addition to the direct DKG. This allows us to dynamically remove users. Furthermore t servers can create a fast witness using the trapdoor instead of the accumulator structure. Due to distributed trust, we do not consider the possibility of cheating with at least t server as an issue.

4.2.4 Comparision

Using an inverse secret sharing requires more computation, communication, and collaboration from the servers than a direct secret sharing. This holds true between a direct secret sharing and having no secret. There is a tradeoff between the complexity of each operation in secret sharing and the number of needed operation for each request. We provide a comparison in table 3.

	None	SS Direct	SS Inverse
Accumulate	$O(n^2)$	$O(n)$	$O(n)$
Witness generation	$O(n^2)$	$O(n)$	$O(1)^*$
Add member	$O(n^2)$	$O(1)$	$O(1)$
Remove member	$O(n^2)$	$O(n^2)$	$O(1)$

Table 3: Complexity of distributed accumulator

* Possibility of cheating in fast witness generation.

Conclusion

We used an accumulator to decrease the size of a Linkable Ring Signature. Furthermore, we proposed a secret sharing to handle the issue of key escrow and improve the efficiency of the accumulator.

References

[1] J. Benaloh and M. D. Mare, "One-way accumulators: A decentralized alternative to digital signatures," *Advances in Cryptology—EUROCRYPT'93*, vol. 765, pp. 274–285, 1994.

[2] A. Mashatan and S. Vaudenay, "A Fully Dynamic Universal Accumulator," *Romanian Academy*, vol. 14, no. EPFL-ARTICLE-188657, pp. 269–285, 2013.

[3] L. Nguyen, "Accumulators from Bilinear Pairings and Applications to ID-based Ring Signatures and Group Membership Revocation," *Cryptology ePrint Archive*, pp. 1–32, 2005.

[4] K. EdalatNejad, "Accumulator Survey." https://goo.gl/TM6di1, 2017.

[5] Jan Camenisch and Anna Lysyanskaya, "Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials," *Crypto*, p. 16, 2002.

[6] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup, "Anonymous identification in ad hoc groups," *Procedings of Eurocrypt*, vol. 3027, pp. 609–626, 2004.

[7] J. Li, N. Li, and R. Xue, "Universal Accumulators with Efficient Nonmembership Proofs," *Proceedings of the 5th international conference on Applied Cryptography and Network Security (ACNS)*, pp. 253–269, 2007.

[8] M. H. Au, P. P. Tsang, W. Susilo, and Y. Mu, "Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5473, pp. 295–308, 2009.

[9] J. Camenisch, M. Kohlweiss, and C. Soriente, "An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials," vol. 5443, pp. 481–500, 2009.

[10] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," *Proceedings - IEEE Symposium on Security and Privacy*, pp. 397–411, 2013.

[11] E. Ghosh, O. Ohrimenko, D. Papadopoulos, R. Tamassia, and N. Triandopoulos, "Zero-knowledge accumulators and set algebra," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10032 LNCS, pp. 67–100, 2016.

[12] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "RingCT 2.0: A Compact Accumulator-Based (Linkable Ring Signature) Protocol for Blockchain Cryptocurrency Monero," pp. 456–474, 2017.

[13] T. M. Kuo, S. M. Yen, and M. C. Han, "Dynamic reversed accumulator," *International Journal of Information Security*, pp. 1–9, 2017.

[14] K. EdalatNejad, "Short Ring Signature Survey." https://goo.gl/8QP6Kk, 2017.

[15] P. Tsang and V. Wei, "Short linkable ring signatures for e-voting, e-cash and attestation," *Information Security Practice and Experience*, pp. 48–60, 2005.

[16] M. H. Au, S. S. M. Chow, W. Susilo, and P. P. Tsang, "Short Linkable Ring Signatures Revisited," pp. 101–115, 2006.

[17] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, "Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction," *Theoretical Computer Science*, vol. 469, pp. 1–14, 2013.

[18] S. S. Chow, W. Susilo, and T. H. Yuen, "Escrowed linkability of ring signatures and its applications," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4341 LNCS, pp. 175–192, 2006.

[19] Q. Wu, W. Susilo, Y. Mu, and F. Zhang, "Ad Hoc Group Signatures," no. 60403007, pp. 120–135, 2006.

[20] F. Zhang, R. Safavi-Naini, and W. Susilo, "An Efficient Signature Scheme from Bilinear Pairings and Its Applications," *In ASIACRYPT'03, LNCS 2894, pp. 452-473, Springer-Verlag, 2003*, vol. 2947, pp. 277–290, 2004.

[21] R. Sakai and M. Kasahara, "ID based Cryptosystems with Pairing on Elliptic Curve," *2003 Symposium on Cryptography and Information Security–SCIS*, 2003.

[22] L. Chen and Z. Cheng, "Security Proof os Sakai-Kasahara's Identity-Based Encryption Scheme," *IMA CC 2005: Cryptography and Coding: 10th IMA International Conference, Cirencester, UK, December 19-21, 2005. Proceedings*, vol. 3796, pp. 442–459, 2005.

[23] A. Shamir, "How to share a secret," *Algorithms Unplugged*, pp. 159–168, 1979.

[24] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pp. 427–437, 1987.

[25] T. P. Pedersen, "A threshold cryptosystem without a trusted party," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 547 LNCS, pp. 522–526, 1991.

[26] A. Kate, "Asynchronous Distributed Private-Key Generators for Identity-Based Cryptography," 2010.

[27] M. Geisler and N. P. Smart, "Distributing the key distribution centre in Sakai-Kasahara based systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5921 LNCS, pp. 252–262, 2009.