Realization of ZKRP Algorithm via Homomorphic Encryption Methods

Sergey Abrahamyan
Institute for Informatics and Automation Problems
Yerevan, Armenia
e-mail: serj.abrahamyan@gmail.com

***Abstract*—Zero-knowledge Range Proof(ZKRP) has gained increasing interest due to its applications in blockchain and cryptocurrencies in particular. ZKRP provides a mechanism to prove that a hidden integer belongs to a given interval without revealing any information about hidden integers. Some ZKRP depends on an honest and reliable third party. Others avoid from the third party. Currently, one of the famous ZKRP is the so-called Bulletproofs proposed by Bunz et al. Applying well-known homomorphic encryption methods in realizing ZKRP is a prospective direction. This paper proposes a new ZKRP based on one of the well-known homomorphic encryption methods- the order-preserving encryption method.**

***Keywords*— Zero Knowledge Proof, Range Proof, Order Revealing Encryption**

I. Introduction

Zero-knowledge proof is a protocol that allows one part, the so-called prover, to convince another part, the so-called verifier, of an assertion without revealing any further information beyond the fact that the assertion is true. ZKP is applied in various applications in which a computation's correctness must be verified by many other parties. It can be applied to a variety of real-world use cases, including identity protection, authentication, autonomous payments, scalability, and decentralized voting systems. One of them is a Zero-knowledge range proof (ZKRP). ZKRP scheme allows to prove that a secret integer belongs to a certain interval without revealing any information about secret integer. The first ZKRP protocol was presented in 1995 by Damgard [6] and in 1997 by Fujisaki and Okamoto [7]. The first practical construction was proposed by Boudot in 2001 [3]. In 2016, Bunz et al. [4] proposed a new idea for constructing ZKRP with a very small proof size, which they called Bulletproof. The idea, similar to some other schemes, is to decompose the secret into the bit representation and, using the "Inner product proof" method, to prove that it belongs to the interval.

The next approach for ZKP is ZK-SNARK(Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge). The idea of ZK-SNARKs is transforming the statement to be proved, into an arithmetic circuit and building algebraic equations from it. ZK-SNARKs are not quantum resistant. Once quantum computing is largely available, the privacy technology behind SNARKs will be broken.

There is another interesting cryptographic primitive called Order Preserving Encryption-OPE(or Order Revealing Encryption-ORE). ORE is a deterministic symmetric encryption scheme the encryption algorithm of which produces ciphertexts that preserve numerical ordering of the plaintexts. OPE was proposed by Agrawal et al. [1] in 2004 as a tool to support efficient range queries on encrypted data. The first formal cryptographic treatment of OPE scheme was given by Boldereva et al. [2]. A number of OPE schemes have been proposed in recent years [1, 2, 10, 5, 8]. Unfortunately, all these ORE schemes are not efficient to be used in practice. Concurrent with these works, Lewi and Wu [9] presented a new and efficient ORE scheme, which is based on the work of Chenette et.al. [5]. The ORE construction proposed by Lewi and Wu leaks less information about the encrypted numbers,which is an important advantage. In this paper, the author proposes to transmute [9] into an efficient ZKRP scheme. As the proposed scheme is based on [9], its short description is given below. In [9], the large-domain ORE scheme consists of three parts: Setup, Encryption(left, right) and Compare. Right encryption is used for encrypting values stored on the server side. During the right encryption process, for each digit x_i of the value to be encrypted, d digits (where d is the radix) of $\mathbb{Z}_3$ are generated, which are the comparison output of x_i and every element of radix. These numbers are then permuted via the permutation function. Thus, each value is represented as a $d \times n$ table of elements of $\mathbb{Z}_3$, where n is a maximal number of digits of upper endpoint. Each element of the table is encrypted, and the table is stored in the database. Left encryption is used only for making a search query. During the left encryption, each digit of the encrypting value is permuted via the permutation function and encrypted. The server via the "Compare" algorithm compares the left encrypted value with the right encrypted value without revealing both of them.

It is easy to see that the structure of the "Large ORE Scheme" in [9] allows us to modify the ORE scheme to ZKRP in the following manner: encrypt the endpoints of the range interval via the right encryption algorithm on the verifier's side, encrypt the secret value via the left encryption in the prover side. Then the prover sends the secret value to the verifier. The latter verifies if the secret value is smaller than the upper endpoint and bigger than the lower endpoint. In order to provide completeness and soundness for the new ZKRP it is important to design a new key management system. Recall that completeness means that the verifier accepts the proof if the

https://doi.org/10.51408/csit2023_15

statement or assertion is true. In other words, an honest verifier will always be convinced of the true statement by an honest prover. Soundness means that in the case of false fact the verifier rejects the proof, which indicates that a cheating prover can cheat an honest verifier with a negligible probability. The other important property of ZKRP is a trusted setup. Many ZKRP constructions depend on a trusted party. A trusted party generates and provides necessary parameters for both prover and verifier. Some ZKRP algorithms avoid the trusted setup[8], which is an obvious advantages. The proposed ZKRP scheme assures completeness and soundness and minimizes the dependence on a trusted setup. The data concerning the algorithm's performance, security parameters and other details will be presented in a full paper to be prepared.

II. New approach for ZKRP protocol

In this chapter, a novel non-interactive ZKRP- a modification of the Levi and Wu ORE scheme, is proposed. The proposed ZKRP algorithm is a tuple of three algorithms (Setup, Prove, Verification).

The Setup algorithm (setuper) samples the pair of secret keys $SK\{k_1, k_2\} \xleftarrow{R} \{0:1\}^{\lambda}$ and sends it to Prover. After that, pads 0's on the left at the upper (lower) endpoint in such a way that the number of digits of the upper (lower) endpoint is equal to n where n is the maximal number of digits of the upper endpoint. Next, $2n$ length d-ary array P is generated. Recall that d is a radix of the system. Then randomly samples n numbers $p_i \in [0; d]$ $i = 1 \dots n$ and these numbers are inserted into randomly chosen n places in array P. The Setuper also sends array P to Prover. After that, the upper (lower) endpoint's digits are sequentially inserted into the free places. The next setup algorithm encrypts the $2n$ length array in which the upper(lower) endpoint is embedded.

The next setup algorithm encrypts the upper and lower endpoints and sends them to the Verifier. The upper and lower endpoints are encrypted via the Right encryption scheme.

The Prover generates a proof, i.e., encrypts the hidden value and sends it to the Verifier for verification. For this end, The Prover pads 0's from the left in the hidden value in such a way that the number of digits of the hidden value is equal to n. Then The Prover sequentially inserts the hidden value's digits into the array P (recall that in array P there are n free places) which he has received from the Setuper. The hidden value is encrypted via the left encryption scheme [**?**].

The verification algorithm compares two $2P$ length encrypted arrays (upper and lower endpoints) with a $2P$ length hidden value and verifies the statement if the hidden value is less than to the upper endpoint and greater than to the lower endpoint. It is easy to see that randomly added numbers cannot impact on verification algorithm because in both arrays they are the same.

III. Performance and Memory requirements

To encrypt a hidden value, one should encrypt all $2n$ digits one by one. To encrypt separate digits, it is necessary to do 2 encryption operations and one permutation operation. Thus, to encrypt a single hidden value, it is necessary to do $4n$ encryption and $2n$ permutation operations. To verify the hidden value, the Verifier should compute the hash value up to $2 \times 2n$ times. Processing encryption of a 32-bit value, presented as a $d = 8$-ary string, on a laptop (16GB RAM and 2.3GHz Intel Core i7GPU), requires $\approx 800\mu s$ of computation. This parameter is also called proving time. And for processing verification, it requires $\approx 4\mu s$ of computation. The volume of encrypted endpoints is $\approx n \times d$ bytes, which is a quite acceptable volume for practical applications.

One of the desired parameters of ZKRP is the proof size. The proof size refers to the number of bytes the proof takes. For the proposed ZKRP, the proof size i.e., the length of the hidden value is 17 bytes when $d \leq 256$.

IV. Conclusion

In this paper, the concept of a novel non-interactive ZKRP scheme, which is based on the ORE scheme [9], is given. The proposed ZKRP concept requires an honest and reliable third party, which is a big disadvantage. However, the proposed method offers pretty good performance and can be useful for systems, where a trusted party is necessary. The follow-up research is planned to concentrate on the complete exclusion of a trusted third party and security proof of the proposed scheme.

Acknowledgment

This work was supported by the RA Science Committee, in the frame of the research project 20TTAT-RBe016.

References

[1] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, "Order-preserving encryption for numeric data", *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Paris, France, pp. 563–574, June 13–18, 2004.

[2] A. Boldyreva, N. Chenette, Y. Lee and A. O'Neill, "Order-preserving symmetric encryption", *Advances in Cryptology - EUROCRYPT 2009*, Berlin, Heidelberg: Springer, pp. 224–241, 2009.

[3] F. Boudot, "Efficient proofs that a committed number lies in an interval", *Preneel, B. (ed): Advances in cryptology - EUROCRYPT 2000*, Berlin, Germany: Sp ringer, pp. 431–444, 2000.

[4] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more", *2018 IEEE symposium on security and privacy (SP)*, pp. 315–334, May 21–23, IEEE, 2018.

[5] N. Chenette, K. Lewi, S.A. Weis and D.J. Wu, "Practical order-revealing encryption with limited leakage", **Fast Software Encryption - 23rd International Conference**, FSE 2016, Bochum, Germany, Revised Selected Papers, pp. 474–493. March 20–23, 2016. [Online], Available: https://www.overleaf.com/project/6227a2f5d5c64893ad6209ac/

[6] I. Damgård, "Practical and provably secure release of a secret and exchange of signatures J. Cryptol", **8**(4), pp. 201–222, 1995.

[7] E. Fujisaki and T. Okamoto, "Statistical zero knowledge protocols to prove modular polynomial relations", *ski, B.S. (ed)*, Advances in cryptology - CRYPTO '97, pp. 16–30, Berlin, Germany: Sp ringer, 1997.

[8] M. Lacharite, B. Minaud and K.G. Paterson, "Improved reconstruction attacks on encrypted data using range query leakage", *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 297–314, IEEE, 2018.

[9] K. Lewi and D.J. Wu, "Order-revealing encryption: New constructions, applications, and lower bounds", *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, pp. 1167–1178, October 24–28, 2016.

[10] I. Teranishi, M. Yung, and T. Malkin, ”Order-preserving encryption secure beyond one-wayness”, *Advances in Cryptology - ASIACRYPT 2014*, pp. 42–61, Berlin, Heidelberg: Springer, 2014.