Efficient Zero-Knowledge Range Proofs in Ethereum

Tommy Koens, Coen Ramaekers and Cees van Wijk

ING,
blockchain@ing.com

1 Introduction

Two months ago I (Tommy) joined ING's blockchain team. This job change was motivated by two ideas. The first being able to combine my PhD research on blockchain technology with my work at ING. Second, I was told that ING's blockchain team is one of the best teams out there. Curious about the team, I decided to go for lunch with Coen and Cees, co-authors of this paper.

To get further acquainted with them, as well as to gain insight in their many blockchain related projects, I asked: 'What is it that you're currently working on?'. The passionate answer was: 'We have created an efficient zero-knowledge range proof for Ethereum'.

Considering the knowledge, skills and experience required to successfully create and implement such a proof, my motivation for changing jobs was confirmed. Ever since, that motivation has been confirmed, many times.
Back on topic: we continued talking about how the Zero Knowledge Range Proof (ZKRP) works, the benchmark results, and why it is of value. This and more, I would like to share with you as well.

1.1 Background and Our Contribution

ING's blockchain team was initiated in 2014 by ING's management, and a hand-selected group of technically highly skilled developers that are interested in cryptocurrencies and distributed ledger technology (DLT). DLT, which includes blockchain technology, is an interesting topic for Financial Institutions. This technology allows for, for example, decentralization of trust, in some cases is faster then existing processes, and offers new business models that include a component of privacy sensitivity. In other words, existing processes can significantly be improved, and new processes can be created that are beneficial to society, customers, our peers and ING. These benefits make that ING contributes to DLT. In this paper, our contribution consists of the following:

- We built upon the work by Peng and Bao [19], by making their work non-interactive, see Section 5.
- We implemented a Zero-Knowledge Range Proof Section 6 in Ethereum and we open-sourced our solution (as of November 16, 2017), see [1].

1.2 Outline of this Paper

First we introduce the concept of zero-knowledge proof (without the math) and also introduce some basic terminology in Section 2. Then, we discuss ZKRP and why they are relevant, in Section 3. Having provided a foundation of knowledge, we introduce the theoretical foundations of ING's zero-knowledge range proof in Section 4. Then we discuss some of the hairy details of ZKRP in Section 5. We follow up with a description of what we did in practice in Section 6. Here we describe how we implemented a ZKRP in Ethereum's testnet, and provide you with the code. In Section 7 we follow up with Frequently Asked Questions section. Finally, we conclude in Section 8 that ZKRP can be used in practice, and that ZKRP are more efficient than current zero-knowledge proofs.

2 Remind Me, What Is Zero-Knowledge Proof Again?

A zero-knowledge proof (ZKP) is a cryptographic method that allows a party (the prover) to prove to another party (the verifier) that a given statement is true, without conveying any additional information. For example, a ZKP allows you to proof that you're of a certain age, without revealing your actual age.

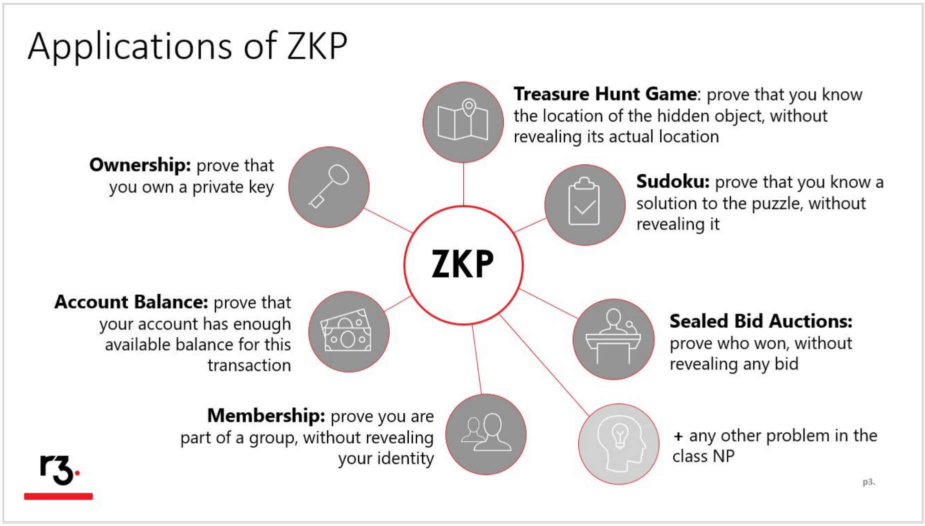
To make it less abstract, there is an easy to understand demonstration given by Chalkias and Hearn [2]. The idea is as follows. Imagine you have a color-blind friend. Your friend has a red and a green ball that are otherwise identical. Your friend is not sure if these actually differ in color. Your job is to convince your friend that these balls are different in color, and reveal nothing else.

The proof works as follows. Your friend puts these balls behind his back. Then, he brings the balls forward, shows them to you, and returns the balls to his back. You now know in which hand is the red, and in which hand is the green ball. Then your friend may switch the balls behind his back, and shows you a single ball (after putting it back again), and asks you: Did I switch the balls?

Remember, your job is to convince your friend that the balls are different in color. Now, consider for a moment that the balls are of the same color. Then you have a 50% of guessing correctly if the balls were switched. After, say 1000 switches, the Law of Large Numbers [3] states that you have guessed approximately 50% of the switches correct. However, if the balls are of different color, you would end up with a very high number of correct statements (over 99%), since you can determine if the switch has been made. Since obtaining such a high score is highly unlikely when the balls are of the same color, your friend can assume that you are stating the truth: the balls are indeed of a different color. From this example we learn that there are ZKP has three properties:

1. Completeness. If the statement is true, an honest verifier will be convinced by an honest prover.
2. Soundness. If the statement is false, no dishonest prover can convince an honest verifier.
3. Zero-knowledge. If the statement is true, no dishonest verifier learns anything other than the fact that the statement is true.

So, why are ZKP relevant? ZKP allow you to make statements about secret values without revealing anything other than that statement [4]. In other words, ZKP allow for many applications in which your privacy can be preserved. Especially in distributed ledger technology ZKP proof useful. Indeed, in most cases, in DLT all participating nodes validate all transactions. This means that, in principle, all nodes see all transactions in plain text - there is no privacy. With ZKP, all nodes can validate, and privacy is guaranteed. Furthermore, some practical examples for ZKP are shown in Figure 1 by R3 [5].



Applications of ZKP

ZKP

Treasure Hunt Game: prove that you know the location of the hidden object, without revealing its actual location

Ownership: prove that you own a private key

Sudoku: prove that you know a solution to the puzzle, without revealing it

Account Balance: prove that your account has enough available balance for this transaction

Sealed Bid Auctions: prove who won, without revealing any bid

Membership: prove you are part of a group, without revealing your identity

+ any other problem in the class NP

Fig. 1. Practical Use Cases for ZKP. Source:[5]

2.1 Interactive vs. Non-interactive ZK-proofs

Note that this demonstration containing two colored balls is interactive. Only you and your friend are convinced by the outcome of the demonstration. If we were to proof to others that, indeed, the color of the balls differ, we would have to perform the demonstration again. Therefore, interactive zero-knowledge proof is limited in transferability. This means that others are unable to verify the outcome of the demonstration. Repeated interaction of a proof makes interactive ZKP impractical for a distributed network. Therefore, we would like to make the ZKP non-interactive, where we (the prover) can show the result, and another party (the verifier) can verify the proof themselves. Note that non-interactiveness is achieved by the Fiat-Shamir heuristic, as we will discuss in Section 4.

2.2 I've also heard about ZK-SNARKS...

There already exist several good, detailed explanations of Zero-Knowledge Succinct Non-interactive ARgument of Knowledge, or ZK-SNARKS. For example, sorted by technical difficulty, [6], [7], and [16]. Therefore, we will not discuss the details of ZK-SNARKS here. However, it is important to understand that ZK-SNARKS, are a ZKP-variant, that requires no interaction necessary between the prover and verifier. The most appealing example of ZK-SNARK is its application in Zcash [8]. Also, Ethereums Metropolis will introduce ZK-SNARKS. This shows that ZK-SNARKS can, indeed, be applied in practice.

ZK-SNARKS are generic; they can verify any function which is particularly relevant for Ethereum which provides the Turing-complete EVM, allowing developers to build any type of logic. Generic ZKPs like ZK-SNARKS, however come at a price. They are notably less efficient than specific zero-knowledge proofs. This efficiency is particularly important in a Blockchain context where all nodes need to verify all transactions. That is why ING has developed and implemented a specific ZKP that can verify a range proof. This ZKP is more efficient than ZK-SNARKS since it requires less computations.

3 Zero-Knowledge Range Proofs

Zero-Knowledge Range Proofs (ZKRP) allow for proofing that a number lies within a certain range. Note that ZKRP holds the same three properties as ZKP, which are completeness, soundness and zero-knowledge.

3.1 Efficiency of a ZK-SNARK

ZK-SNARKs verification are, to some extent, computational intensive. Given current processor capacity, this is not an issue. However, in Ethereum each node verifies the proof. This requires a certain amount of fee for ZK-SNARK verification. For example, this transaction [9] (and the contract source [10]) in Ethereum's testnet Ropsten would cost about 1.9 million gas. Considering an Ether price of 283 dollar per Ether and a gas price of 0,000 000 021, we can calculate the cost of transaction verification: $1.900.000 * 0,000.000.021 * 283 \approx 11$ dollars. Indeed, one of the main concerns about Ethereum is that of efficiency [11].

3.2 A Significant Efficiency Gain with ZKRP

We benchmarked our ZKRP implementation against several available functions, including the SHA256 hashing function. Our implementation is a factor 10 more efficient then the ZK-SNARKS solution currently used in Ethereum's test network. Our ZKP implementation costs 180.000 gas, so we can calculate: $180.000 * 0,000.000.021 * 283 \approx 1$ dollar. This is a significant gain, compared to the 11 dollars in Paragraph 3.1.

In the following sections we provide details on the building blocks of our protocol, as well as a more in-depth description of how our protocol operates.

4 ZKRP: The Foundation

In this section we describe the theoretical foundation of ING's ZKRP. ING's ZKRP is an improvement on a paper title An Efficient Range Proof Scheme, by Peng and Bao [19]. Their work describes an interactive ZKRP. Our work improves on the work of Peng and Bao [19] by making it non-interactive. Also, the ZKRP protocol consists of several building blocks from earlier publications and properties. These are:

1. Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations [18]
2. Efficient Proofs that a Committed Number Lies in an Interval [17]
3. Discrete Logarithm problem [12]
4. Fiat-Shamir heuristic [13]
5. Integer Factorization problem [14]
6. The Secret Order principle [19]
7. Chinese Remainder theorem [15]

We will now briefly discuss these building blocks and properties to gain a better understanding of how ING's ZKRP implementation works.

4.1 The Discrete Logarithm problem

In real numbers, taking the logarithm is a trivial operation. For example

$$\begin{aligned} b^n = x \text{ then } \log_b x = n \\ 2^3 = 8 \text{ then } \log_2 8 = 3 \end{aligned} \tag{1}$$

However, if we take a cyclic group, calculating the discrete logarithm appears to be very hard. A cyclic group is a group of numbers generated by a single element g, called the generator. The Discrete Logarithm problem can more formally be described as follows: Given a group G, a generator g of the group and an element h of G, to find the discrete logarithm to the base g of h in the group G.

There are no polynomial time algorithms that solve the Discrete logarithm problem. Therefore, if we choose large enough numbers solving, the discrete logarithm becomes impractical to solve. It can always be solved given enough computation time.

4.2 The Fiat-Shamir heuristic

The purpose of this cryptographic technique is that a fact (e.g. knowledge of a certain number) can be proven without revealing the actual fact. For example, Alice wants to proof that she knows the solution to $b^n = x$. She picks a random integer v and calculates $s = b^v$. Then, Alice computes $c = H(b, x, s)$, where H is a hash function. Also, she computes $r = v - (n * c)$, and publishes the proof (s,r). Then, anyone can verify that $s = b^r x^c = b^{v-nc} b^{nc} = b^v$. Since the outcome of the hash function H was unpredictable, Alice must have known n to be able to compute r, because otherwise Alice was able to solve the Discrete Logarithm problem. Therefore, by only providing (s, r), Alice proofs the fact that she knows the number n.

4.3 The Integer Factorization problem

Integer factorization is the decomposition of any composite number (i.e. a positive integer) into a product of smaller integers. An example is: $20 = 4 * 5$. If these smaller integers are restricted to prime numbers (e.g. $15 = 3 * 5$), and the composite number is sufficiently large (preferably with only large prime factors), then there currently exists no algorithm to solve this in polynomial time.

4.4 The Secret Order principle

The secret order principle states that it is difficult to compute the order of an element in a group where the prime factors of the modulus are unknown. This principle relies on the Integer Factorization problem. The order of an element in a cyclic group is the exponent which reduces the element in the group to the identity element. As an example, in the integer modulo 7, 3 has the order 6 (7-1):

$$\begin{aligned} &3 * 3 * 3 * 3 * 3 * 3 \bmod 7 = \\ &2 * 2 * 2 \bmod 7 = \\ &8 \bmod 7 = 1 \end{aligned} \tag{2}$$

4.5 Chinese Remainder theorem

As stated by Wikipedia, if one knows the remainders of the Euclidean division of an integer n by several integers, then one can determine uniquely the remainder of the division of n by the product of these integers, under the condition that the divisors are pairwise coprime [15]. For example, what is the smallest number x that divided by 3 has a remainder of 2, dividing by 5 a remainder of 3, and divided by 7 a remainder of 2? The correct answer here can be provided by the theorem, and is 23.

4.6 Fujisaki-Okamoto Commitment

Let N be a large composite number of which the prime factorization is unknown to Alice. Let g be a large element in the group $\mathbb{Z}_N^*$, and let h be an element of the group generated by g, such that both discrete logarithms are unknown. Let $E(x, r) = g^x h^r \bmod n$ be a commitment to x in base (g, h) where r is a random value. Then, it is computationally infeasible to compute x_1, x_2, r_1, r_2 where $x_1 \neq x_2$ such that $E(x_1, r_1) = E(x_2, r_2)$. Finally, $E(x, r)$ statistically reveals no information of x to Bob.

5 ING's ZKRP: In Theory

Our ZKRP implementation efficiently proofs that a secret number lies within a given range. More precise, we wish to prove that a secret integer m is in the range of $\{a, a + 1, \ldots, b\}$. This is true if and only if $(m - a + 1)(b - m + 1)$ is positive. Again, this is true and only if $w^2(m-a+1)(b-m+1)$ is positive, for any random integer w. To prove that $w^2(m-a+1)(b-m+1)$ is positive, we choose random non-negative integers m_1, m_2, m_3, such that $m_1+m_2+m_3 = w^2(m-a+1)(b-m+1)$, and $m_3 = m_4^2$.

If $x = s * m_1 + m_2 + m_3$ and $y = m_1 + t * m_2 + m_3$ are positive, for large positive random integers s and t, then $w^2(m-a+1)(b-m+1)$ is also positive with an overwhelmingly large probability. Note that the information revealed about m by x and y is negligible, since only x and y are revealed and no value m_i. is revealed.

However, publishing x and y does not provide sufficient information to convince a verifier, since x and y could be random numbers. Therefore, the verifier needs to be convinced that the proofer did actually use the correct commitment. This can be done by the use of secret order groups.

A secret order group is used throughout the ZKRP protocol, the generation of which requires finding two large primes P and Q for which $(P-1)/2$ is also a prime. The secret order group is formed by taking $N = PQ$, and from generators of the subgroups mod P and mod Q, using the Chinese remainder theorem, generators g and h of the secret order group can be computed.
Note that the factorization of N remains unknown to Alice. The secret integer m is committed to $c = g^m h^r \bmod N$, where r is a random integer. Then, the prover publishes $c_1 = c/g^{a-1} \bmod N, c_2 = g^{b+1}/c \bmod N,$ and $c' = c_1^{b-m+1}h^{r'} \bmod N$, where r' is a random integer and gives proof that c_2 and c' hide the same secret. The prover then publishes $c'' = c'^{w^2}h^{r''} \bmod N$ and proofs that the hidden number is a square with base (c', h).

The prover then chooses random non-negative integers m_1, m_2, m_3 such that $m_1+m_2+m_3 = w^2(m-a+1)(b-m+1)$ and $m_3 = m_4^2$, and also chooses random integers r_1, r_2, r_3 such that $r_1 + r_2 + r_3 = w^2((b-m+1)r + r') + r''$ and publishes $c'_1 = g^{m_1}h^{r_1} \bmod N$, $c'_2 = g^{m_2}h^{r_2} \bmod N$, and $c'_3 = g^{m_3}h^{r_3} \bmod N = g_4^{m^2}h^{r_3} \bmod N$. The prover then gives the proof that c_3 hides a square [17]

Finally $x = sm_1 + m_2 + m_3$, $y = m_1 + tm_2 + m_3$, $u = sr_1 + r_2 + r_3$ and $v = u = r_1 + tr_2 + r_3$ are published. A verifier can now verify that $c_1 = c/g^{a-1} \bmod N, c_2 = g^{b+1}/c \bmod N, c'' = c'_1c'_2c'_3 \bmod N, c'^S_1c'_2c'_3 = g^xh^u \bmod N, c'_1c'^t_2c'_3 = g^yh^v \bmod N$, and that x and y are both positive numbers.

This, in principle concludes the ZKRP. To further clarify, from these statements we can conclude that indeed m lies within an interval $\{a, \ldots, b\}$. Earlier, the prover proved that c'' is a commitment to a square w^2 with base (c', h) so then we get:

$$\begin{aligned} c'' &= c'^{w^2} h^{r''} \bmod N && \text{we know that } c' = c_1^{b-m+1} h^{r'} \bmod N \\ &= (c_1^{b-m+1} h^{r'})^{w^2} h^{r''} \bmod N && \text{we know that } c_1 = c/g^{a-1} \bmod N \\ &= ((cg^{(1-a)b-m+1} h^{r''} \bmod N && \text{we know that } c = g^m h^r \bmod N \\ &= (g^{(m-a+1)(b-m+1} h^{(b-m+1)r+r'})^{w^2} h^{r''} \bmod N && \text{rewriting} \\ &= (g^{(m-a+1)(b-m+1} h^{(b-m+1)r+r'})^{w^2} h^{r''} \bmod N && \text{definition of } m_i \text{ and } r_i \\ &= g^{m_1+m_2+m_3} h^{r_1+r_2+r_3} \bmod N = c'_1 c'_2 c'_3 \bmod N \end{aligned}$$

The proof that m lies within a range is sketched as follows. The verifier knows that x, y, s, t are all positive. From that, given the above equation, the verifier knows with a overwhelming probability that $w^2(m-a+1)(b-m+1)$ is positive, leading to the fact that m is in the interval $\{a, a+1, \ldots, b\}$.

For the full article including mathematical proofs, we reference to [19].

6 ING's ZKRP: In Practice

How can we practically apply a ZKRP? Here are some examples:

- Proof that my salary is sufficient to rent a house, or obtain a mortgage, without revealing the exact number.
- Proof that a payment amount is within a limit, but it does not show the exact amount.
- Location based proof. I can proof that I am in a country, without revealing my exact location.

6.1 Implementing the ZKRP

ING built a precompiled contract in Geth, that allows the entire network to verify a secret number in a known range. A precompiled contract is a specific, predefined, optimized function that runs outside the EVM but can be called from a regular solidity smart contract that runs inside the EVM. A precompiled contract is part of the Ethereum client. We decided not to implement the ZKRP verifier in a regular solidity smart contract because precompiled contracts execute much faster than regular Solidity smart contracts in the EVM. Additionally, the proof is an array of numbers which is too large for the largest Solidity numeric type (int256).

7 Frequently Asked Questions

- **Is the code available?**
  Yes. The code can be found here [1]
- **Under what license is the code published?**
  GPL
- **What if the interval (range) is too small, or multiple intervals have a small overlap, can you then determine the secret number (e.g. age)?**
  Yes. Though the range proof is cryptographically strong the intervals should be chosen large enough to preserve confidentiality.
- **If you require a commitment from a trusted party stating your age why use a range proof and not just a boolean-statement stating that the number is between 18 and 200?**
  The commitment needs to be provided once and can be re-used for multiple intervals.
- **Will this ZKRP be part of Ethereum or Quorum?**
  Preferably both, but both are open source so we need community consensus to include it. We are working with the EEA quorum working group to make this part of Quorum. Also, we consider applying for the Ethereum Improvement Process to add it to Ethereum.

8 Conclusion

ZKP is very promising and becomes more practical to use in distributed ledgers. Current implementations, however, are resource demanding and there is a need for efficient zero-knowledge range proof solutions. Indeed, in this article we described ING's blockchain team proposal for a more efficient Zero-Knowledge Range Proof. Benchmarking this ZKRP with current ZKP in Ethereum (test environment) shows that the ZKRP implementation is 10 times more efficient. Also, as this article discussed, there are practical use cases in which a ZKRP is useful. Finally, the source code is publicly available at [1].

We would gladly receive any feedback on this work, share experiences, or discuss improvements. If you have any questions/remarks, please contact ING's blockchain team at: blockchain@ing.com

References

1. `https://github.com/ing-bank/zkrangeproof`,.
2. `https://www.linkedin.com/pulse/demonstrate-how-zero-knowledge-proofs-work-without-using-chalkias`, (visited on November 2, 2017).
3. `https://en.wikipedia.org/wiki/Law_of_large_numbers`, (visited on November 2, 2017).
4. `https://isp.cs.ru.nl/2016/danezis.pdf`, (visited on November 2, 2017).

5. https://www.linkedin.com/pulse/demonstrate-how-zero-knowledge-proofs-work-without-using-chalkias, (visited on November 2, 2017).
6. https://www.linkedin.com/pulse/demonstrate-how-zero-knowledge-proofs-work-without-using-chalkias, (visited on November 2, 2017).
7. https://medium.com/@VitalikButerin/zk-snarks-under-the-hood-b33151a013f6, (visited on November 2, 2017).
8. https://z.cash/, (visited on November 2, 2017).
9. https://ropsten.etherscan.io/tx/\0x15e7f5ad316807ba16fe669a07137a5148973235738ac424d5b70f89ae7625e3, (visited on November 2, 2017).
10. https://ropsten.etherscan.io/address/0xa1f11d83a5222692c0eff9eca32254a7452c4f29#code, (visited on November 2, 2017).
11. https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/57506f387da24ff6bdecb3c1/1464889147417/Ethereum_Paper.pdf, (visited on November 3, 2017).
12. https://en.wikipedia.org/wiki/Discrete_logarithm, (visited on November 2, 2017).
13. https://en.wikipedia.org/wiki/Fiat%E2%80%93Shamir_heuristic, (visited on November 2, 2017).
14. https://en.wikipedia.org/wiki/Integer_factorization, (visited on November 2, 2017).
15. https://en.wikipedia.org/wiki/Chinese_remainder_theorem, (visited on November 2, 2017).
16. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112. ACM, 1988.
17. Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In *Advances in CryptologyEUROCRYPT 2000*, pages 431–444. Springer, 2000.
18. Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology-CRYPTO'97: 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 1997. Proceedings*, page 16. Springer, 1997.
19. Kun Peng and Feng Bao. An efficient range proof scheme. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 826–833. IEEE, 2010.