An Improved Range Proof with Base-3 Construction

Esra Günsay[1][0000−0001−6672−4253], Cansu Betin Onur[1][0000−0002−3691−1469], and Murat Cenk[1][2222−−3333−4444−5555]

Institute of Applied Mathematics,
Middle East Technical University, Ankara, Turkey
{gunsay,cbetin,mcenk}@metu.edu.tr

Abstract. Zero-knowledge protocols (ZKPs) allow a party to prove the validation of secret information to some other party without revealing any information about the secret itself. Appropriate, effective, and efficient use of cryptographic ZKPs contributes to many novel advances in real-world privacy-preserving frameworks. One of the most important type of cryptographic ZKPs is the zero-knowledge range proofs (ZKRPs). Such proofs have wide range of applications such as anonymous credentials, cryptocurrencies, e-cash schemes etc. In many ZKRPs the secret is represented in binary then committed via a suitable commitment scheme. Though there exist different base approaches on bilinear paring-based and RSA-like based constructions, to our knowledge there is no study on investigating the discrete logarithm-based constructions. In this study, we focus on a range proof construction produced by Mao in 1998. This protocol contains a bit commitment scheme with an OR-construction. We investigate the effect of different base approach on Mao's range proof and compare the efficiency of these basis approaches. To this end, we have extended Mao's range proof to base-3 with a modified OR-proof. We derive the number of computations in modulo exponentiations and the cost of the number of integers exchanged between parties. Then, we have generalized these costs for the base-u construction. Here, we mainly show that comparing with other base approaches, the base-3 approach consistently provides approximately 12% efficiency in computation cost and 10% efficiency in communication cost. We implemented the base-3 protocol and demonstrated that the results are consistent with our theoretical computations.

Keywords: Zero knowledge proof · Range proof · OR proof · Commitment schemes.

1 Introduction

The zero-knowledge proofs have recently gained the utmost importance in many systems, especially in the context of privacy. These proofs are used to prove the accuracy of a specific information in a secret way. In many privacy-preserving systems, ZKPs are used as the building blocks. A particular and notable example of ZKPs is the zero-knowledge range proofs, which can be seen as a special

case of zero-knowledge set-membership proofs (ZKSMPs). The basic principle of the ZKSMP is to prove to a third party that a member of a public/private set (countries, names, nationalities, etc.) without revealing any information about the secret itself. When we take this set as a list of integers, such proofs are called as ZKRPs. The corresponding hidden knowledge is often the output of a cryptographic function. There are numerous usage areas and various cryptographic systems of ZKRP in the real world such as e-voting [14], age validation (prove someone is over 18), e-cash [7,18], risk assessments/credit score systems [12] for banking and financial institutions (prove salary of an individual is above some threshold), investment grading (rating companies due their financial status), e-auction [11], group signature schemes [5] and verifiable secret sharing. Camenisch et. al. [4] used range proofs to prove knowledge of a signature on a committed integer in anonymous credentials and group signatures. Recently, with the fast development of the blockchain based distrubuted ledger technology ZKRPs become even more popular since they used to validate the transactions of a cryptocurrency. Monero [19], Zcash [20], and Zerocoin [18] are just a few examples of cryptocurrencies using ZKRPs on the validation of the transaction process.

The foundations of the range proofs go back to the 80's. Brickell, Chaum, Damgård & van de Graaf [2] proposed the first elements to construct a range proof in 1987. They achieved to send a secretly disclosed bits to other participants. Their construction was based on discrete logarithm using a *bit commitment*. To check the accuracy of the proof they use $\Sigma-$protocol. This construction has many negative features especially in ranging. In 1995, Damgard [9] proposed a ZKRP construction. Soon later, in 1997, Fujisaki & Okamoto [10] proposed another construction. Although these proposed constructions work properly, they were inefficient to use in the real-world cases. In 1997, Bellare and Goldwasser [6] presented *the binary decomposition range proof*. In this construction, the secret s is represented in binary, i.e., on modulo-2 basis. In 1998, Chan et al. [7] constructed a scheme, known as CFT proof, using the algorithm by Brickell et al. [2]. However, their construction was only succeeded on the non-negatives ranges and the order of the used group must be unknown. In 2000, Boudot [1] proposed a scheme depending on the strong RSA problem. To show the secret x lies in the interval of the form $[a, b]$, it is sufficient to prove both $x - a$ and $b - x$ are positive.

Almost in all ZKRP constructions the secret is committed at the beginning of the scheme. In the literature, there exist three main approaches to commit the secret. These are *integer*, *binary*, or *the $u-$ary method*. Hence, these methods are listed as:

1. *binary method* This method allows to check that the binary representation of a committed value is in the interval $[0, 2^k - 1]$.
2. *integer method* In this method, it is enough to check whether a committed number belongs to an interval $\mathcal{I} = [a, b]$ or not. Usually, $\mathcal{I}$ is chosen as a much larger interval space.

3. *$u-$ary method* This way allows to check that the u-ary representation of a committed value is in the interval $[0, u^k - 1]$. As it is stated in the literature [3], this method itself does not reduce the proof size.

Camenish et. al. [3] proposed improvements on the bilinear-group assumption based set-membership proofs which also can be used for range proofs. In their study, it is recommended to represent the secret in base-u, instead of the folklore base-2 approach. However, it is stated that this idea does not reduce the proof size, thus it does not bring any efficiency alone. Therefore they constructed a scheme that enables to reuse the list of u signatures sending by the verifier. In the range proof instantiation of this approach, the verifier sends the list of u-signatures, and the prover use this list to check the accuracy. In their approach, I denotes a range of integers such as $[1, u^n]$, where u is the representation base. Elements of I are signed using a digital signature by the verifier. These signatures are considered as common inputs. The prover proves that she knows a signature under the verification key for the element committed to C, while C is a commitment. Moreover, they showed that their approach also can be applicable to the Strong RSA-like assumption with a significant level of improvement.

Though different base approaches investigated on bilinear groups and RSA-like schemes, as far as we know, there is no generic study on discrete logarithm based schemes such as Mao's range proof. Due to its wide range of usage areas, we investigate the efficiency of different base approaches on discrete logarithm based settings. Our aim is to find the most efficient base approach in both computation and communication costs.

1.1 Our Contribution

In 1998, Mao [13] proposed a binary multi-party secret sharing construction that can be corrected by a single verifier. In this construction to encrypt corresponding primes, *a proof of bit length* is used. We call this proof of bit length scheme as the classical range proof. In the classical range proof proposed by Mao, the secret x is represented in binary, and the binary commitment scheme is used afterward. To our knowledge, there is no known investigation on Mao's range proof in different base approaches. In this paper, we investigate the usage of Mao's range proof, with different base approaches. In this protocol, an OR-Proof proposed in [8] is used as a sub-protocol. For this purpose, we decompose the secret in the $u-$ary method, with an adapting sub-protocol. The results show that this approach in base-3 is more efficient than other base choices in both computation and communication costs.

1.2 Outline of the Paper

In Section 2, we present the details of the underlying cryptographic primitives such as commitment schemes, zero knowledge proofs, and Σ-protocols. The classical range proof proposed by Mao is described in Section 3. In Section 4, we

represent the details of base-3 approach to the existing construction. Generalizations to different bases are analyzed and compared in Section 5, where we show that it yields the best efficiency in base-3 both in computation cost and communication cost. We implement the base-2 and base-3 methods and compare the implementation results in Section 6.

2 Preliminaries

Some of the basic primitives, notations, and definitions are introduced in this section. ZKRPs contain many sub-protocols in their construction. Since we only examine Mao's construction in this paper, we only define related primitives. Commitments have an important role in almost every range proof scheme. The secret value is expressed as a committed value. Therefore, after giving a short notations part we explain commitment schemes and the Pedersen commitment scheme more precisely. Then, we explain $\Sigma-$protocols and their OR-composition due to their significant role in Mao's construction.

2.1 Notations

Over the ring of integers $\mathbb{Z}$, let s, p, q be large primes such that $q = 2s + 1$ and $p = kq + 1$ hold where k is an even positive integer. Let $g \in \mathbb{Z}_p^*$ be an element of order q, and G be a group generated by g. We know that $g^q = 1 \bmod p$. Let $f \in G$ be a fixed element generated by a pseudo-random generator which is seeded by g, and its discrete logarithm in base g, $\log_g f$ is unknown. Throughout the paper $r \in_R \mathbb{Z}_p^*$ denotes that r is randomly chosen in $\mathbb{Z}_p^*$. For randomly chosen r, we say that $E = \mathrm{Com}(x, r) = g^x f^r$ is a commitment to hide x.

2.2 Commitment Schemes

Commitments are important building blocks due to their major role in many cryptographic applications. A commitment scheme is a deterministic polynomial-time algorithm, which has two stages, namely *committing* and *revealing*. In this study, we will focus on bit commitment schemes. A bit commitment scheme is a cryptographic primitive satisfying hiding and binding properties.

- *perfectly hiding* It should be computationally infeasible to reveal x for a given $Com(x, t)$.
- *binding* It should be infeasible to find two different openings from one committed value which guarantees the committer cannot forge the system even if she changes her mind. This property directly holds for the Pedersen commitment scheme due to the DLP assumption.

Pedersen Commitment Schemes The idea of the Pedersen commitment with perfectly hiding and computationally binding properties is presented for the first time in [16,17]. The security of the scheme is based on the hardness of the discrete logarithm problem (DLP).

In the setup phase, the reciever picks uniformly ramdom primes q and p with $q|(p-1)$. Suppose G is the cyclic subgroup of $\mathbb{Z}_p^*$ of order q, and $G =< g >$. The reciver picks an element f in G randomly where $\log_g f$ is unknown.

In the committing phase, for a randomly chosen $t \in_R \mathbb{Z}_q^*$ to commit a secret $x \in \mathbb{Z}_q^*$, the committer computes $Com(x,t) = g^x f^t$. The opening phase is quite similar. The committer reveals x and the corresponding t for the opener to compute $Com(x,t) = g^x f^t$ to check its correctness. A Pedersen commitment scheme should satisfy the perfectly hiding and binding properties.

2.3 Zero Knowledge $\Sigma-$protocol and OR-composition

$\Sigma-$protocols are special types of interactive 3-move honest verifier zero-knowledge proofs (HVZK). The first movement is usually a committed value sent by the prover. The second movement is by the verifier, and it is usually a large enough uniformly random challenge. The third movement comes from the prover to aim that the verifier will be able to run a proof of knowledge with some specific steps.

Definition 1 (Σ-Protocol). *Let P and V refer the probabilistic polynomial time machines. For the protocol system pair (P,V), where $\mathcal{R}$ is a binary relation, a Σ-Protocol for the relation $\mathcal{R}$, is of the 3 movement form, namely message, challenge, and response.*

Let both P and V have x as a common input. P has a private input w, where $(x,w) \in \mathcal{R}$. A typical Σ-protocol needs to achieve three security parameters. These are (perfect) completeness, special soundness, and special honest-verifier zero-knowledgeness (sHVZK).

Combining the existing protocol for achieving different aims results in compositions. One of these compositions is the OR-composition, which establishes the correctness of one of the given two statements. In Figure 1, we show the workflow of the OR-composition of $\Sigma-$protocol for the Schnorr protocol. It is a 3-movement protocol which is used as a sub-protocol in the context of range proofs. The details of the workflow will be explained in detail in the next section.

3 Classical Range Proof

In the classical range proof protocol, the main idea is to prove a secret integer x belongs to some interval of the form $[0, 2^{k+1}-1]$. To this end, we decompose the secret in base two, then prove that this decomposition truly occurs by 0's and 1's.

The length of the secret x equals to $\lfloor \log_2 x \rfloor + 1$ in bit-wise representation, and since $x \in [0, 2^{k+1}-1]$, x can be written as:

$$x = x_0 2^0 + x_1 2^1 + \ldots + x_k 2^k \text{ for } x_i \in \{0,1\} \text{ and } i = 0,1,\ldots,k. \quad (1)$$

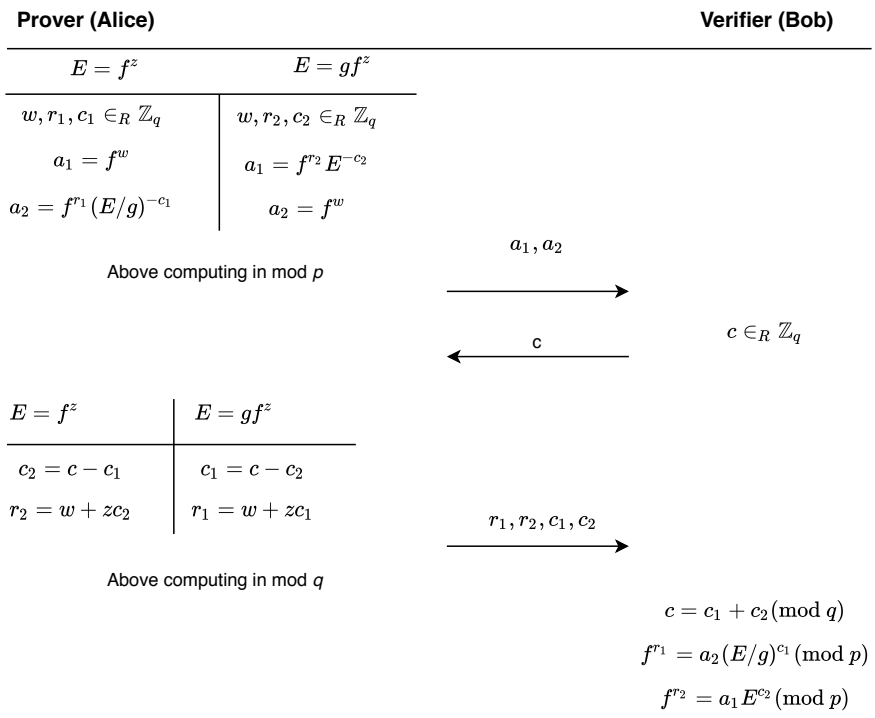


Fig. 1: OR-composition of $\Sigma-$protocol for Schnorr protocol.

The prover chooses $t_0, t_1, \ldots, t_k \in_R \mathbb{Z}_q$, and computes t as:

$$t = \sum_{i=0}^{k} t_i 2^i \bmod q. \tag{2}$$

Then, she computes the following bit commitment scheme:

$$E_i = E(x_i, t_i) = g^{x_i} f^{t_i} \bmod p \text{ for } i = 0, 1, \ldots, k. \tag{3}$$

After that, the prover proves that, in each step, the value committed by $E(x_i, t_i)$ is whether 0 or 1. For this purpose, one can use a zero-knowledge sub-protocol, namely the OR-composition of Σ-protocol, and shows that she knows whether E_i is in base f or E_i/g is in base f as shown in the Figure 1.

In this protocol, the prover proves that a commitment E hides a value of 0 or 1. If the secret value is 0, the commitment equals $E = f^z$, if the secret is 1 then it becomes $E = gf^z$. Depending on the secret, the prover computes a_1 and a_2 and sends them to the verifier. The verifier randomly selects a challenge c. Using the corresponding challenge the prover sends a response to the verifier. Lastly, some equality checks are done, and this ends up the OR-proof.

Finally, the verifier gets E_i and t values. Then, he requires to check (4) using homomorphic proporty.

$$g^x f^t \stackrel{?}{=} \prod_{i=0}^{k} E_i^{2^i} \bmod p. \tag{4}$$

In each iteration of this proof, both the prover and the verifier compute four exponentiations, and seven integers each length $k^{'}$ are exchanged between them. At the end of the proof, the cost of the exponentiations equals $4k$. Similarly, the cost of exchanging numbers equals $7k^{'}k$.

4 Classical Range Proof with Base-3 OR-Construction

In this section, we prove the secret integer x belongs to some interval of the form $[0, 3^{\tilde{k}+1} - 1]$. To do so, this time we decompose the secret in ternary representation and prove that decomposition truly occurs in 0's, 1's, and 2's. First, we use the ternary-length-based representation instead of the bit-length-based one. The length of the secret x equals to $\lfloor \log_3 x \rfloor + 1$ in ternary representation and $x \in [0, 3^{\tilde{k}+1} - 1]$. Hence, we denote the secret we want to prove in a ternary representation as follows:

$$x = x_0 3^0 + x_1 3^1 + \ldots + x_{\tilde{k}} 3^{\tilde{k}} \text{ for } x_i \in \{0, 1, 2\} \text{ and } i = 0, 1, \ldots, \tilde{k}. \tag{5}$$

The prover chooses $t_0, t_1, \ldots, t_{\tilde{k}} \in_R \mathbb{Z}_q$, and computes t as:

$$t = \sum_{i=0}^{\tilde{k}} t_i 3^i \bmod q. \tag{6}$$

After that, we compute the commitments as:

$$E_i = E(x_i, t_i) = g^{x_i} f^{t_i} \bmod p \text{ for } i = 0, 1, \ldots, \tilde{k}. \tag{7}$$

Then, we need to use the OR-proof. However, by using the classical base-2 OR-proof, we cannot check each commitment for once in our case. Hence, instead of base-2 OR-proof, we provide the base-3 OR-proof to prove that the committed value is whether equal E_i, or E_i/g, or E_i/g^2 as seen in Figure 2.

Similar to base-2 construction, the Figure 2 illustrates that the prover proves that a commitment E, hides a value of 0 or 1 or 2. If the secret value is 0, the commitment equals $E = f^z$. If the secret is 1 then it becomes $E = gf^z$, and if the secret is 2 it becomes $E = g^2 f^z$. Depending on the secret, the prover computes a_1, a_2 and a_3 and sends them to the verifier. The verifier randomly selects a challenge c. Using the corresponding challenge, the prover sends a response to the verifier. Lastly, some equality checks are done, and this ends up the OR-proof.

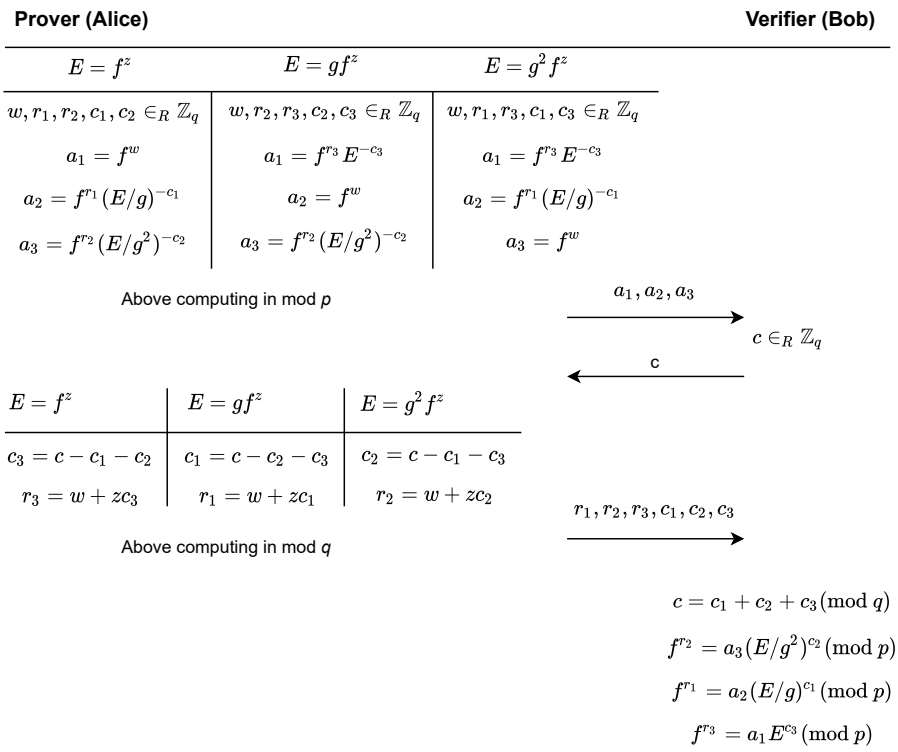


Fig. 2: Base-3 OR-proof.

After getting E_i and t values, the verifier needs to check the equality of the following property:

$$g^x f^t \stackrel{?}{=} \prod_{i=0}^{\tilde{k}} E_i^{3^i} \bmod p. \tag{8}$$

In each step, both the prover and the verifier need to compute six exponentiations. This time there exist 10 integers to exchange which costs $10k^{'}$. At the end of the protocol, the overall cost for exponentiation is $6\tilde{k}$ and the cost of exchanging integers is $10k^{'}\tilde{k}$.

5 Generalizations to Base-u and Comparisons

In this section, we analyze the performance of the base-u representation of the secret x. Clearly, the base-3 scheme succeeds in our case. Recall that, for the complexity of the base-3 construction, both the prover and the verifier need to compute six exponentiations in each step. In total, we need $6\tilde{k}$ exponentiations. In addition to this computational comparison, we consider the number of exchanged integers between the prover and the verifier. In this scenario, there

exist 10 integers to exchange between the prover and the verifier in each step. We denote the length of the integers by $k^{'}$. The cost of the exchange then is equal to $10\tilde{k}k^{'}$ for the base-3 case.

From now on, we use $\mathcal{E}$, $\mathcal{I}$, and $\mathcal{M}$ to denote the cost of the exponentiation, inversion, and multiplication operations in $\mathbb{Z}_p^*$, respectively. Before generalizing this scheme to the base-u, it is enough to compute g^{-1} only once. Since we work in $\mathbb{Z}_p^*$, the cost of the inversion can be considered as $\mathcal{I} \approx \mathcal{E}$ [15]. We may also assume that $\mathcal{E} > 1000\mathcal{M}$ using the square and multiply algorithm for cryptographic applications [15]. That is why other operations can be seen as negligible. Hence, it is enough to take into account the exponentiations and inverses. In general, when we work in base-u, both the prover and the verifier need to compute $2u$ exponentiations in each step. Furthermore, there are $3u+1$ numbers to exchange in each step in base-u. The required exponentiations are given in Table 1.

Table 1: Comparisons of the required operations for OR-proof in each step

Cost type \ Basis	Base-2	Base-3	Base-4	Base-5	Base-6	...	Base-u
operations	$4\mathcal{E}+\mathcal{I}$	$6\mathcal{E}+\mathcal{I}$	$8\mathcal{E}+\mathcal{I}$	$10\mathcal{E}+\mathcal{I}$	$12\mathcal{E}+\mathcal{I}$	...	$2u\mathcal{E}+\mathcal{I}$
numbers to exchange	7	10	13	16	19	...	$3u+1$

The above computations are valid only for one iteration of OR-proof. In the context of range proof, these exponentiations and the number exchanges repeat as many times as the secret length in base u, since we call the OR-proof as many times as the secret length. Let $k = \lfloor \log_2 x \rfloor$ and $\tilde{k} = \lfloor \log_3 x \rfloor$. Since $\log 2 = 0.30102$ and $\log 3 = 0.47771$, in the base-3 representation $\tilde{k} = \frac{\log 2}{\log 3} k \approx 0.63k$. So, we can compare these two worst case complexities of the operations executed by the prover as follows:

$$\frac{0.63k(7\mathcal{E})}{k(5\mathcal{E})} = \frac{4.41\mathcal{E}}{5\mathcal{E}} \approx 0.88. \tag{9}$$

This corresponds to approximately 12% efficiency in the proof generation when we use the ternary representation instead of the binary representation.

We also analyze other basis complexities in the same way and generalize this analysis for base-u. For base-4, since $\frac{\log 2}{\log 4} \approx 0,5$, remember we have 8 exponentiations and one inverse operations are required. In total $4.5k$ operations are required, which means we have 10% efficiency in base-4 representation. Continuing similar computations, we get the results for the other basis. For example we observed that we have 6% efficiency in base-5 representation. In base-6 total operation cost equals $5.02k$, and no more efficiency comes with comparing to base-2. In general, for the base-u representation, the number of required exponentiations can be formalized as follows:

$$\frac{\log 2}{\log u}(2u)k. \tag{10}$$

This means after achieving improvements on base-3, base-4, and base-5 representations, the number of operations increases starting by base-6. We also observe that the base-3 approach is the most efficient approach with respect to computation cost.

Table 2: Table that compares the required operations for range proof in total

Cost type \ Basis	Base-2	Base-3	Base-4	Base-5	Base-6	...	Base-u
operations	$5k$	$4.41k$	$4.5k$	$4.7k$	$5.02k$	...	$\frac{\log 2}{\log u}(2u)k$
numbers to exchange	$7k^{'}$	$6.309k^{'}$	$6.5k^{'}$	$6.88k^{'}$	$7.22k^{'}$	...	$\frac{\log 2}{\log u}(3u+1)\ k^{'}$

Similar computations can be done for total numbers exchanged data between the prover and the verifier. In the binary representation approach, $7kk^{'}$ bits are exchanged. In the ternary representation, $10\tilde{k}k^{'}$ bits are exchanged and as we mentioned before:

$$\frac{0.63k(10k^{'})}{k(7k^{'})} = \frac{6.309\mathscr{E}}{7\mathscr{E}} \approx 0.90. \tag{11}$$

Hence, we have approximately 10% efficiency in the base-3 approach comparing with the base-2 approach. It can be formalized for a general base-u as $\frac{\log 2}{\log u}(3u+1)$, and the most efficient computations come in base-3. In Table 2, cost of requirements tabulated for different basis.

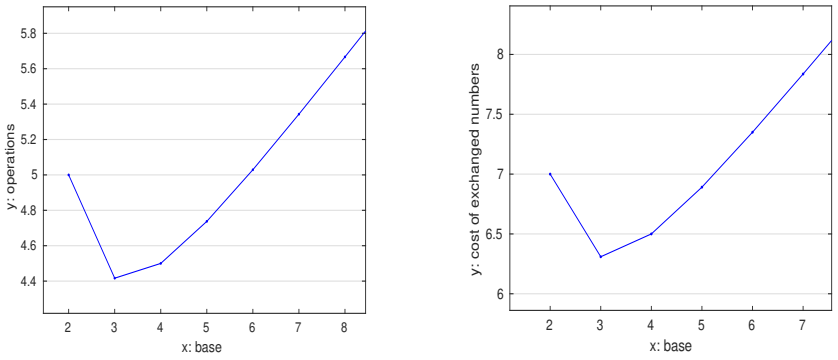
We also sketch the total required exponentiations among different basis in the first of the following graphs of Figure 3. The graph shows that maximum efficiency can be observed when the base is selected as three. In the first graph 3a, you may find the number of required exponentiations for the different basis. The graph has its minimum value in $(3, 4.41)$, which is our most efficient point.

In the second graph 3b, we can see the number of bits exchanged. Although in folklore bit-representation it equals 7, in base-3, base-4 and base-5 it has better results. Still, it has its minimum value in $(3, 6.31)$, which is our most efficient point. As a result of both comparisons, the construction achieves the most efficiency when using the base-3 approach.

6 Implementation

For the implementation of protocols C++ language was selected due to its portability to the lower level languages and wide range of library options. All experiments were carried out using a single core of an Intel Core i7-8565U Processor CPU running at 1.8 GHz. We made the codes available as open-source on the GitHub. [1] [2]

[1] `https://anonymous.4open.science/status/RangeProof_Base2_ZKRP-7499`
[2] `https://anonymous.4open.science/status/RangeProof_Base3_ZKRP-F4C3`



(a) Required exponentiation (b) Number of exchanged integers

Fig. 3: Comparison of different base approaches.

We used The MPIR - Multiple Precision Integers and Rationals Library to handle the arithmetic operations on big integers. MPIR is a highly optimized open-source multi-precision integer library, which is forked from the GMP - The GNU Multiple Precision Arithmetic Library.

The parameters p, q, g, f are defined at the very beginning of the algorithm. The primes up to 1024-bit are generated separately using another implementation. The inverse of g is pre-computed at the beginning in both base-2 and base-3 versions. In the provers part, since the exponentiation on modular function $'mpz\_pown'$ also accepts minus variables, $-c_1, -c_2$, and $-c_3$ are directly used in the function so that no extra inverses are computed. Although the number of operations is increased in the base-3 version, the number of calls is decreased since we call the for loop of OR-proof less than the base-2 version. This brings the proven efficiency to the overall result. We compare the computation

Table 3: Table that compares the average computation times in milliseconds for different prime sizes.

Primes \ Basis	Base-2	Base-3	Efficiency in Base-3
512-bit	223.49 ms	203.04 ms	9.15%
1024-bit	984.37 ms	875.52 ms	11.05%

times in milliseconds. The implementation can be run for different prime sizes. In the Table 3, one may find the computation times (in milliseconds) for different prime length trials. Overall, we obtain that the base-3 version works 11.05% more efficiently than the base-2 version. This is consistent with the theoretical computations in Section 5.

7 Conclusion

We have presented a base-3 OR-proof that has soundness and zero knowledgeness properties. We introduced a modified OR-composition of Shnorr protocol to base-3. We computed the overall complexity in the context of required exponentiations and the cost of the number of integers that are exchanged. After our derivations, we showed that the cost has a pattern so that we have generalized and formalized the proof for different bases. At the end of these comparisons, we have obtained that the computational cost of the base-3 representation is 12% more efficient than the other base representations. Moreover, comparing in the context of the communication cost, we showed that the base-3 approach is 10% more efficient with respect to the other base approaches. We have also implemented the protocols and it has been observed that the base-3 protocol is 11.05% faster than the base-2 protocol.

References

1. Boudot, F.: Efficient proofs that a committed number lies in an interval. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000. LNCS, vol. 1807, pp. 431–444. Springer Berlin Heidelberg, Berlin, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-$6_3$1
2. Brickell, E.F., Chaum, D., Damgård, I.B., van de Graaf, J.: Gradual and verifiable release of a secret (extended abstract). In: Pomerance, C. (ed.) Advances in Cryptology - CRYPTO '87. LNCS, vol. 293, pp. 156–166. Springer Berlin Heidelberg, Berlin, Heidelberg (1988). https://doi.org/10.1007/3-540-48184-$2_1$1
3. Camenisch, J., Chaabouni, R., shelat, a.: Efficient protocols for set membership and range proofs. In: Pieprzyk, J. (ed.) Advances in Cryptology - ASIACRYPT 2008. LNCS, vol. 5350, pp. 234–252. Springer Berlin Heidelberg, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-$7_1$5
4. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings. LNCS, vol. 3152, pp. 56–72. Springer (2004). https://doi.org/10.1007/978-3-540-28628-8_4
5. Camenisch, J., Michels, M.: Separability and efficiency for generic group signature schemes. In: Wiener, M. (ed.) Advances in Cryptology - CRYPTO' 99. LNCS, vol. 1666, pp. 413–430. Springer Berlin Heidelberg, Berlin, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-$1_2$7
6. Canard, S., Coisel, I., Jambert, A., Traoré, J.: New results for the practical use of range proofs. In: Katsikas, S., Agudo, I. (eds.) Public Key Infrastructures, Services and Applications. LNCS, vol. 8341, pp. 47–64. Springer Berlin Heidelberg, Berlin, Heidelberg (2014). https://doi.org/10.1007/978-3-642-53997-8_4
7. Chan, A.H., Frankel, Y., Tsiounis, Y.: Easy come - easy go divisible cash. In: Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding. LNCS, vol. 1403, pp. 561–575. Springer (1998). https://doi.org/10.1007/BFb0054154

8. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings. LNCS, vol. 839, pp. 174–187. Springer (1994). https://doi.org/10.1007/3-540-48658-5$_1$9
9. Damgård, I.B.: On the existence of bit commitment schemes and zero-knowledge proofs. In: Brassard, G. (ed.) Advances in Cryptology — CRYPTO' 89 Proceedings. LNCS, vol. 435, pp. 17–27. Springer New York, New York, NY (1990). https://doi.org/10.1007/0-387-34805-0$_3$
10. Fujisaki, E., Okamoto, T.: Statistical zero knowledge protocols to prove modular polynomial relations. In: Kaliski, B.S. (ed.) CRYPTO '97. LNCS, vol. 1431, pp. 16–30. Springer Berlin Heidelberg, Berlin, Heidelberg (1997). https://doi.org/10.1007/BFb0052225
11. Hahn, A., Singh, R., Liu, C.C., Chen, S.: Smart contract-based campus demonstration of decentralized transactive energy auctions. In: 2017 IEEE Power & energy society innovative smart grid technologies conference (ISGT). pp. 1–5. IEEE (2017). https://doi.org/10.1109/ISGT.2017.8086092
12. Lin, C., Luo, M., Huang, X., Choo, K.K.R., He, D.: An efficient privacy-preserving credit score system based on noninteractive zero-knowledge proof. IEEE Systems Journal (2021). https://doi.org/10.1109/JSYST.2020.3045076
13. Mao, W.: Guaranteed correct sharing of integer factorization with off-line shareholders. In: Public Key Cryptography, First International Workshop on Practice and Theory in Public Key Cryptography, PKC '98, Pacifico Yokohama, Japan, February 5-6, 1998, Proceedings. LNCS, vol. 1431, pp. 60–71. Springer (1998). https://doi.org/10.1007/BFb0054015
14. McCorry, P., Shahandashti, S.F., Hao, F.: A smart contract for boardroom voting with maximum voter privacy. In: Kiayias, A. (ed.) Financial Cryptography and Data Security. LNCS, vol. 10322, pp. 357–375. Springer International Publishing (2017). https://doi.org/10.1007/978-3-319-70972-7$_2$0
15. Menezes, A., Katz, J., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. Discrete Mathematics and Its Applications, CRC Press (1996). https://doi.org/10.1201/9780429466335
16. Metere, R., Dong, C.: Automated cryptographic analysis of the pedersen commitment scheme. In: Rak, J., Bay, J., Kotenko, I., Popyack, L., Skormin, V., Szczypiorski, K. (eds.) Computer Network Security. LNCS, vol. 10446, pp. 275–287. Springer International Publishing (2017). https://doi.org/10.1007/978-3-319-65127-9$_2$2
17. Micali, S., Rabin, M., Kilian, J.: Zero-knowledge sets. In: 44th Annual IEEE Symposium on Foundations of Computer Science 2003 – FOCS 2003., Proceedings. pp. 80–91 (2003). https://doi.org/10.1109/SFCS.2003.1238183
18. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from bitcoin. In: 2013 IEEE Symposium on Security and Privacy. pp. 397–411. IEEE (2013). https://doi.org/10.1109/SP.2013.34
19. van Saberhagen, N.: Cryptonote v 2.0. In: Tech. Rep. [Online] (2013), `Available: https://cryptonote.org/whitepaper.pdf`
20. Sasson, E.B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy. pp. 459–474 (2014). https://doi.org/10.1109/SP.2014.36