Zero knowledge proof for sign of message value

Asked 5 years, 4 months ago Modified 2 years, 10 months ago Viewed 1k times

4

I am using ElGamal encryption to encrypt an integer message m as,

$E[m] = (g^x, g^m . h^x)$

Can I write a zero-knowledge proof to prove to a verifier that $m > 0$?

3

I can create the bit representation of m and encrypt each bit to prove this, but I have a lot of integers and am trying to avoid this approach.

zero-knowledge-proofs elgamal-encryption

Share Edit Follow

edited Dec 4, 2016 at 5:33

asked Dec 4, 2016 at 1:00

btan
143 4

Do you mean $m \neq 0$, or do you mean $0 < m < q/2$? – poncho Dec 4, 2016 at 4:13

The latter, i.e. $0 < m < q/2$. I am not concerned about including the upper bound in the proof since I am going to homomorphically add a bunch of these integers and prove a bound on the sum. I just need to prove that each integer, individually, is a greater than zero. – btan Dec 4, 2016 at 5:31

This paper is related link.springer.com/chapter/10.1007%2F978-3-662-53015-3_18 – redplum Dec 4, 2016 at 7:10

1 Answer

Sorted by: Highest score (default)

11

What you are looking for is called a range proof. There has been a vast body of research on the topic recently - so vast, in fact, that it can be quite hard to know what is the state of the art, and what solution is the most appropriate in a given situation. Therefore, to let you evaluate by yourself what might best suit your exact situation, I'll attempt at providing a quick survey of the existing methods and their constraints.

The most natural method is the naive method that you mentioned: commit to the bits $(m_i)_{i \leq \log q/2}$ of the plaintext m, prove that $\sum m_i 2^i = m$ and that $m_i(1 - m_i) = 0$ for every i (id est, each m_i is a bit). Unfortunately, this takes $O(\log q)$ group elements, hence it is very costly. An improvement to this naive method has been suggested in this article. The idea is the following: rather than decomposing m in base 2, decompose it in base B. You

now simply need to commit to the $\log_B(q/2)$ elements m_i of the decomposition in base B. To prove efficiently (in constant time) that $m_i \in \{0, \cdots, B-1\}$, the idea is to let the verifier add signatures of the integers $0, \cdots, B-1$ to the public key, and to use known methods to let the prover show that he can produce a signature for the committed value. As he cannot forge a signature for any value outside $\{0, \cdots, B-1\}$, it is sufficient to complete the proof. This gives you a proof which communicates $O(\log q / \log \log q)$ group elements, but it has a downside: it requires to use an elliptic curve with an appropriate pairing operation, as such a pairing is required for the signature part. The constant in this $O()$ has been improved in this article.

Now, you might also be fine with a proof that does not hold for *arbitrary* provers, but only for computationally bounded provers - it won't make much difference in practice. This is called a zero-knowledge *argument*, and there are more efficient methods in this case. The one I will describe is not explicitely mentioned in an article, but is a natural consequence of a serie of works by Groth and Bayer (and can therefore be seen as folklore). The core idea is that one can commit to many numbers at once using a scheme called "generalized Pedersen commitment scheme": the public key contains $n+1$ random group elements, and to commit to n values $(a_1, \cdots, a_n)$ with random coin r, compute $c = h^r \cdot \prod_i g_i^{a_i}$. This is perfectly hiding, and computationally binding under the discrete logarithm assumption.

To prove that m is in $[0, q/2]$, decompose m in base 2: $m = \sum_i m_i 2^i$. Group these bits by packs of $\sqrt{\log q}$ consecutive bits (you'll have $O(\sqrt{\log q})$ such packs). Commit to each pack, with a single group element per pack, using the generalized Pedersen scheme. Now, perform a batch product proof to show that all the committed values are bits (ie, they satisfy $m_i^2 - m_i = 0$). This kind of batch proofs are non-trivial, but have been studied in detail in e.g. this article. Essentially, they are all constructed by carefully and repeatedly applying the Schwartz-Zippel Lemma, which is used to say (informally) that if someone knowns a random linear combination of some values, he must know all the values. Applying this lemma allows to shrink the dimensions of the problem.

The prover must also prove that $m = \sum_i m_i 2^i$; the previously mentioned article also shows how such proofs can be constructed. The techniques developed essentially allow to perform an efficient (batched) proof for any statement over committed values that can be expressed with linear algebra, see this article. Overall, this gives you a range proofs with a communication of $O(\sqrt{\log q})$ group elements. It involves many rounds (I'd say 7 to 9), but can be made non-interactive if you are fine with using the Fiat-Shamir transform, which is secure in the random oracle model (hence only "heuristically secure").

If you want an even shorter proof, then you can consider again using an elliptic curve with a pairing, to use the improved version of the above method: this article does essentially what I described above, but gains additional efficiency by using a pairing, which allows (informally) to *commit to commitments*, hence to get one more level of "packing". It gives a protocol which communicates $O(\log^{1/3} q)$ group elements.

And finally... You can have a proof of constant size. But then, you must rely on a completely different method: first, commit to m in a group of unknown order, and prove that you've committed the same value in two different groups, using for example this article. To have

such a group, the usual method is to have an RSA modulus $N = pq$, product of two safe primes, and to take a generator of the group of squares, whose order $\phi(N)/2$ is unknown unless one can factor N. Note that this means that the constant will be large: factorization-based methods need very large groups (say, $|N| = 2048$). This is therefore suitable for range proofs on large intervals. The trick is the following: by a theorem from Lagrange, an integer is positive if and only if it can be decomposed as a sum of four squares. Therefore, one simply commits to the elements of this decomposition (which can be efficiently computed) and prove that the sum of their squares is m. The group of unknown order is required to ensure that we indeed work over the integers: the prover cannot perform modulo reductions if he does not know the modulus. This method was described in this article, and improved several time. To my knowledge, the most efficient constructions of constant size range proofs are given in this article, which is one of my recent works. In particular, we review the existing optimized range proofs and give new constructions, while weakening the necessary security assumptions.

To choose an appropriate scheme is now up to you, depending on the following:

- Are you willing to use an elliptic curve with pairings?
- Are you Ok with using the random oracle model to get less interactive arguments?
- Would you be fine with using additional assumptions, and a group of unknown order, which might require an additional setup phase?
- How many range proofs will you have to perform? Range proofs can be batched: using a combination of this article and my article allows to perform M range proofs simultaneously, incurring only a $\sqrt{M}$ overhead. As you seem to have many such range proof to perform, it would probably be better to perform a global batched range proofs.
- What is the exact size of the intervals you will have? (having q of size 256 bit, if using elliptic curves, or, say, 1024 bits if using subgroups of prime order fields, can change the appropriate method)

I do not provide detailed explanations of the methods in my answer, it would be too long; you can either refer to the articles I've included, or ask for precisions on a particular method if one of them seems to suit your requirements.

Share Edit Follow

edited Jun 5, 2019 at 15:14

Squeamish Ossifrage
45.4k 3 98 200

answered Dec 4, 2016 at 12:42

Geoffroy Couteau
16.5k 2 39 56

Thanks for the treasure trove of information. I'm going to look at the articles mentioned and decide. – btan Dec 4, 2016 at 22:27