## Math for dh parameters

* [A Computational Introduction to Number Theory and Algebra (Version 2) (shoup.net)](https://shoup.net/ntb/ntb-v2.pdf)
  + Euler’s totient
  + Primitive root
  + Quadratic residues
  + Legendre symbol
* [BigNum Math | ScienceDirect](https://www.sciencedirect.com/book/9781597491129/bignum-math)
  + Miller rabin primality test
  + Exponentiation

## Bernardo email about DH params

[Generation of a cyclic group of prime order - Cryptography Stack Exchange](https://crypto.stackexchange.com/questions/22716/generation-of-a-cyclic-group-of-prime-order)

[diffie hellman - How to get the order of a group generator in DH? - Cryptography Stack Exchange](https://crypto.stackexchange.com/questions/87137/how-to-get-the-order-of-a-group-generator-in-dh)

1. If q is prime, then p=2q+1 is \*not\* automatically prime. This is easy to see with a counter-example. Let q = 7, then p = 2\*7+1 = 15 which is not a prime.  
   Thus, a safe prime p is a prime number p=2q+1 such that q is also prime.
2. P will indeed be the modulus for your DH
3. This is what I was missing last time, when your modulus is a safe prime it’s easier to find a generator. However you are not quite correct in what you said here.  
   A generator in this group mod p is any quadratic non-residue, i.e., any number with Legendre value of -1 (exc. So to find a generator in this group you sample a random value and compute its Legendre value; if it’s -1 then you are done, otherwise you try another one. You will find a generator in the first try with about 50% probability.

I’m adding some extra references here so you can understand better the problem. This is a nice discussion to add to the report, because most of the times those details are swept under the rug.

<https://en.wikipedia.org/wiki/Legendre_symbol>

<https://en.wikipedia.org/wiki/Quadratic_residue>

<https://math.stackexchange.com/questions/145578/what-are-the-generators-for-mathbbz-p-with-p-a-safe-prime>

<https://en.wikipedia.org/wiki/Safe_and_Sophie_Germain_primes>

## DH secret key size

* Secret key needs to be similar size as modulus (2048 bits)
* Look at NIST and ISO or something like that to get guidelines on DH key sizes
* Reference them. Say I am using these key sizes based on this website

## KDF (scrypt) in EKE

* I used KDF scrypt function to derive key from DH common secret key
* Say in the report that this is an improvement
* Say why I need it
  + After the key exchange, I get the same common secret key.
  + But I need to verify that both parties have agreed on the same key
  + Hence, I send challenges
  + The generated key is too large to be used for AES encryption
  + Hence, I use KDF to get the key for AES encryption
* It is okay to have a fixed salt since both parties need the same salt.
* Maybe check the salt part

## Proof : quadratic-non residue is generator

* [ntb-v2.pdf (shoup.net)](https://shoup.net/ntb/ntb-v2.pdf)–
  + Also, note that alpha is a generator for Zn if and only if alpha is a primitive root modulo p. [Page 175 (page 157 on book]
  + Let p be an odd prime. Then |(Z\*p)^2| = (p-1)/ 2 [Theorem 2.20]
    - This means for every odd prime p, exactly half the elements of Z\*p are squares and half are non-squares
* [elementary number theory - Can an element be a quadratic residue and a generator (mod p)? - Mathematics Stack Exchange](https://math.stackexchange.com/questions/105978/can-an-element-be-a-quadratic-residue-and-a-generator-mod-p)
* [elementary number theory - Relationship between primitive roots and quadratic residues - Mathematics Stack Exchange](https://math.stackexchange.com/questions/588774/relationship-between-primitive-roots-and-quadratic-residues)
* Stanford notes on this-
  + [lecture19.pdf (stanford.edu)](https://theory.stanford.edu/~trevisan/cs276/lecture19.pdf)

## Montgomery Exponentiation

[Notes7-Montgomery.pdf (ucsb.edu)](http://koclab.cs.ucsb.edu/teaching/cs154/docx/Notes7-Montgomery.pdf)

[A Low Latency Montgomery Modular Exponentiation - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S1877050920310565)

[Montgomery Arithmetic | SpringerLink](https://link.springer.com/referenceworkentry/10.1007/978-1-4419-5906-5_38)

[Full details and actions for The art of computer programming.: (Seminumerical algorithms.) (vlebooks.com)](https://www.vlebooks.com/Product/Index/1023355?page=0&startBookmarkId=-1)