4 Public key algorithms

Questions & Answers

1. Symmetric ciphers rely on shared secrets and thus lack the convenience of key distribution that comes with public key algorithms. Can you think of problems that still remain unsolved? Maybe read through Wikipedia, Public-key Crypto, Alteration of Public Keys first.
    - Man in the middle is still possible. (in the web we use certificates that can be checked using preinstalled certificates of a CA, that we (have to) trust)
2. The class of public key algorithms make use of so-called "hard problems" to assure that private keys cannot be calculated from public keys. The problems are basically a kind of one-way functions. Easy to calculate in one direction but hard to reverse. Describe the two popular "hard problems" in simple words and give an example of an algorithm or protocol that makes use of it.
    - Integer factorization (used by RSA)
        - e.g: 2 large prime numbers are multiplied to the result "LP". It is hard to find out which 2 prime numbers "LP" is based on.
    - Discrete log problme: (Diffie-Hellman key exchange) - given a + b, find integer k so a = $(b powered by k)
3. RSA is by far the most popular public key algorithm. That said, you are expected to describe the key setup (components and criteria) as well as the encryption and decryption formulas. Make a simple example with very small numbers. Follow the guide at Wikipedia, RSA (cryptosystem).
    - *I wrote it down in word and inserted it as image as markdown support for math is limited in my tool*

RSA·public/private·key·generation¶

Choose·2·random·prime·numbers··(should·be·large)·and·compute·produce:·n·=·p*q·=>·3*11·=·33¶

Choose·random·integer·e··with···e·<·(p-1)*(q-1):···20¶

Compute·the·unique·inverse:···d·=·e^{-1}·mod(p-1)(q-1))¶

- → d·*·e···mod·((p-1)(q-1)))·=·1···*mathematical·operations·in·finite·fields:··y·=·x^{-1}·mod(n)·=>··x*y·mod(n)·=·1*¶
- → (d·*·e)···mod·(20)·=·1¶
  - ➢ → Multiple·possibilities·for·d·and·e¶
    - → 3·*·7···(and·always·the·reverse)¶
    - → 9·*·9¶
    - → Etc.¶

¶

Public·key:··modulus·n,·e:···33,···3·¶

Private·key·d:···7¶

¶

RSA·Encryption/Decryption·example¶

Plaintext·x:····16¶

Encryption:··y·=·x^e·mod(n)··=>···16^3·mod(33)·=·4096·mod(33)·=·4¶

Decryption:·x·=·y^d·mod(n)···=>····4^7·(mod33)·=>·16384·mod(33)·=>·16¶

- ←
- rsa calculator: https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html

4. OPTIONAL Do the same for ElGamal. Follow the guide at Wikipedia, ElGamal(encryption).
   - skipped 😟)
5. Alice sent Bob an encrypted number: 587. Mallory has intercepted the message and also holds a copy of Bob's RSA public key (n,e 2773,17). Proof that small hard problems are not hard and calculate the plaintext number.
   - Answer: plaintext: "31" and d is "157"

     y·=·x^e·mod(n)···=>···x^{17}·mod(2773)·=···587¶

     n·->·is·the·product·of·2·prime·numbers:···Found·via·trial·and·error:·47·+·59¶

     d·*·e···mod·((p-1)(q-1)))···->·····d·*·17·mod(46*58)·=·1·=>···(d·*·17)·mod(2668)·=·1¶

     Possibilities·for·d*17·->·¶

     - → 1¶
     - → 2669···->·157···->·d¶
     - → 5337·->·313.94...·(not·a·candidate)¶

     Decryption:·x·=·y^d·mod(n)···=>····587^{157}·mod(2773)·=>·31···(calculation·large·numbers·done·with·a·little·program:·see·https://stackoverflow.com/questions/2177781/how-to-calculate-modulus-of-large-numbers·)·¶

     ¶

     Encrypte·again·to·check·x·=·y^d·mod(n)···=>····1609^{17}mod(2773)·=>·587···(verified·with·https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html·)¶
6. OPTIONAL The Diffie-Hellman key agreement protocol is an early follower of the public key algorithm idea.

1. How does the protocol work? Use Alice and Bob as parties.
    - skipped as I'm a bit short of time 😦
2. What is the public and what is the private key composed of?
    - skipped as I'm a bit short of time 😦
3. Assuming Mallory is in the position to rely and alter communication. Could you describe an attack that may allow interception of traffic?
    - man in the middle