Discussion 4A Recap

Tyler Zhu

March 1, 2020

1 RSA

The setup is that Alice is trying to send a message to Bob securely so eavesdroppers can't figure out the message. RSA is a public-key cryptography scheme, meaning that anyone can encrypt a message and send it to Bob with his public key, but only he has the private key to decrypt the message.

Setup: Bob picks two large primes p, q and another (small) number e which is relatively prime to $(p-1)(q-1)$. He releases $(N = pq, e)$ to the world as his public key. He also computes $d = e^{-1} \pmod{(p-1)(q-1)}$ but keeps it private.

Encryption: If Alice wants to send a message x, she transmits $E(x) = x^e \pmod{N}$ to Bob.

Decryption: Bob receives $y = E(x)$ and decrypts it by doing $D(y) = y^d \pmod{N} = x^{ed} \equiv x \pmod{(p-1)(q-1)}$.

The proof of correctness relies on Fermat's Little Theorem (really Euler's Totient Theorem) and the proof of security relies on the hardness of the discrete logarithm problem, i.e. there is no efficient way to undo exponents.

2 Remarks

- The message x needs to be $< N$, otherwise Bob can't recover the original message.
- There's lots of bad cases that can occur in RSA schemes, such as...
  - Using $N_1 = pq_1$ and $N_2 = pq_2$ with the same e (falls to Euclidean Algorithm)
  - Using $e = 3$ and sending the same message in multiple different N's (see Discussion)
  - Using $e = 3$ and picking p, q that aren't $2 \pmod{3}$
  - Using a small N or limited values (bypass discrete log problem by bashing through all possible x).
- Think hard about what you know and using loose bounds (think practically!)