RSA, Polynomials, Secret Sharing, Erasure Errors 4

Computer Science Mentors 70

February 19 - 21, 2018

1 RSA

1.1 Introduction

Main idea: find an encryption function E and an decryption function D such that $D(E(x)) = x$. In other words, two people can encrypt and decrypt a message x (an integer) if they know E and D.

Mechanism:

Public key: (N, e)

Private key: d

Assumption: Given N, there is no efficient algorithm to determine $(p-1)(q-1)$.

Encryption function: $E(x) = x^e$ mod N

Decryption function: $D(x) = x^d$ mod N, where $d = e^{-1}$ mod$(p-1)(q-1)$

1.2 Proof

Goal: To show $(x^e)^d \equiv x \bmod (N)$, $\forall x \in \{0, 1, \cdots, N-1\}$.
Since $ed \equiv 1 \bmod (p-1)(q-1)$ (that's how we define d), $ed = 1 + k(p-1)(q-1)$ for some integer k. Therefore, $x^{ed} = x^{1+k(p-1)(q-1)}$

$\implies x^{ed} - x = x^{1+k(p-1)(q-1)} - x = x(x^{k(p-1)(q-1)} - 1)$.

Claim: $p \mid x^{ed} - x$
Case 1 ($x \not\equiv 0 \bmod p$):
According to Fermat's Little Theorem, $x^{p-1} \equiv 1 \bmod p$.
$\implies (x^{p-1})^{k(q-1)} \equiv 1^{k(q-1)} \mod p$
$\implies x^{k(p-1)(q-1)} - 1 \equiv 0 \bmod p$
$\implies x^{k(p-1)(q-1)+1} - x \equiv 0 \bmod p$
$\implies p \mid x^{ed} - x$
Case 2 ($x \equiv 0 \bmod p$):
$p \mid x(x^{ed-1} - 1)$
$\implies p \mid x^{ed} - x$
By symmetry, $q \mid x^{ed} - x$. Thus, $pq \mid x^{ed} - x$, which means $(x^e)^d \equiv x \bmod (N)$.

1.3 Questions

1. **How does RSA work?**

   a. Alice wants to send Bob a message $m = 5$ using his public key ($n = 26$, $e = 11$). What cipher text E(m) will Alice send?

Solution:

$$
\begin{aligned}
5^1 &= 5 \mod 26 \\
5^2 &= 5 \mod 26 \\
&= -1 \mod 26 \\
5^4 &= (-1)^2 \mod 26 \\
&= 1 \mod 26 \\
5^8 &= 1 \mod 26 \\
5^{11} &= 5^8 * 5^2 * 5^1 \mod 26 \\
&= 1 * -1 * 5 \mod 26 \\
&= -5 \mod 26 \\
&= 21 \mod 26
\end{aligned}
$$

b. What is the value of d (Bobs private key) in this scheme? Note that traditional RSA schemes use much larger prime numbers, so its harder to break n down into its prime factors than it is in this problem.

Solution: $n = 26 \rightarrow$ because $26 = pq$ and $p \neq a * q$ for all a within integers, $p = 13$, $q = 2$

$$d = e^{-1} \mod (13-1)(2-1)$$

$$d = 11^{-1} \mod 12$$

$$d = 11$$

2. In RSA, if Alice wants to send a confidential message to Bob, she uses Bobs public key to encode it. Then Bob uses his private key to decode the message. Suppose that Bob chose $N = 77$. And then Bob chose $e = 3$ so his public key is (3, 77). And then Bob chose $d = 26$ so his private key is (26, 77).
   Will this work for encoding and decoding messages? If not, where did Bob first go wrong in the above sequence of steps and what is the consequence of that error? If it does work, then show that it works.

   Solution: e should be co-prime to $(p-1)(q-1)$.
   $e = 3$ is not co-prime to $(7-1)(11-1) = 60$, so this is incorrect, since therefore e does not have an inverse $\mod 60$.

1.4 Extra Practice

1. **Coin tosses over text messages**
   You and one of your friends want to get your hands on the new gadget thats coming out. One of you has to wait in line overnight so that you have a chance to get the gadgets while they last. In order to decide who this person should be, you both agree to toss a coin. But you wont meet each other until the day of the actual sale and you have to settle this coin toss over text messages (using your old gadgets). Obviously neither of you trusts the other person to simply do the coin toss and report the results.

   How can you use RSA to help fix the problem?

   Solution: Firstly, we need some way to create two events of equal probability. One way to do this is have both of you flip a coin, and if you both get the same coin (i.e. "HH" or "TT") then one of you has to wait in line, and if you both get different coins (i.e. "HT" or "TH") then the other one of you needs to wait in line. Since each combination of two flips has probability $\frac{1}{4}$, each of you has a $2 \cdot \frac{1}{4} = \frac{1}{2}$ chance of having to wait in line.

   Some form of encryption is necessary in this question, because otherwise, you could send your non-encrypted result to your friend and they could simply lie and say that they got the result that would cause you to have to go wait in line. Consider the following abstraction:

   1. I flip a coin, write down my result on a piece of paper, and turn over the piece of paper
   2. I give you the paper, you flip your coin, and then turn over my paper

The main idea of the above scheme boiled down to the fact that I was able to give you my result without you being able to see it. After flipping my coin and writing down the result, I'm unable to change my flip, therefore I was forced to "commit" to my answer. Since you weren't able to see my coin, you had no choice but to actually flip the coin, as you had no additional information.

Let's implement this with RSA.

1. You generate a public key (N, e), and flip your coin. If you got heads, encrypt some random word beginning with the letter "H". If you got tails, encrpyt some random word beginning with the letter "T". The reasoning for this will be evident in the next step.
2. Send your encrypted message to your friend, along with your public key. Sending the public key is important, since it forces you to commit to the result that you got (if you were to send your public key after your friend flipped their coin, you could calculate a public key and private key such that your encrypted message decrypts to whatever you want it to).
3. Since your friend currently has no information about your flip, as you can't decrypt with just the public key, they have no choice but to just flip their coin and relay their result back to you.
4. Now, you send your friend your private key. With your public key and private key, they can decrypt your message to see what your original word was, telling them whether you got heads or tails.

2 Polynomials

2.1 Introduction

We can use two representations for polynomials: value representation and coefficient representation.

1. Value representation: if polynomial $P(x)$ has degree $n-1$ then we can uniquely reconstruct it from any n distinct points
2. Coefficient representation: if a polynomial $P(x)$ has degree $n-1$ then it can be uniquely described by its n coefficients

2.2 Questions

1. Define the sequence of polynomials by $P_0(x) = x+12$, $P_1(x) = x^2-5x+5$ and $P_n(x) = xP_{n-2}(x) - P_{n-1}(x)$. (For instance, $P_2(x) = 17x - 5$ and $P_3(x) = x^3 - 5x^2 - 12x + 5$.) Show that $P_n(7) \equiv 0 \pmod{19}$ for every $n \in N$.

   Solution:

   1. Prove using strong induction.

      Base Case There are two base cases because each polynomial is defined in terms of the two previous ones except for P_0 and P_1.

      $$P_0(7) \equiv 7 + 12 \equiv 19 \equiv 0 \pmod{19}$$
      $$P_1(7) \equiv 7^2 - 5 \cdot 7 + 5 \equiv 49 - 35 + 5 \equiv 19 \equiv 0 \pmod{19}$$

      Inductive Hypothesis Assume $P_n(7) \equiv 0 \pmod{19}$ for every $n \leq k$.

      Inductive Step Using the definition of P_{k+1}, we have that

      $$\begin{aligned} P_{k+1}(7) &\equiv xP_{k-1}(7) - P_k(7) \pmod{19} \\ &\equiv x \cdot 0 - 0 \pmod{19} \\ &\equiv 0 \pmod{19} \end{aligned}$$

   Therefore, $P_n(7) \equiv 0 \pmod{19}$ for all natural numbers n.

2. Let $P(x)$ be a polynomial of degree 2 over GF(7).

   1. Suppose $P(0) = 3$ and $P(1) = 2$. How many values can $P(5)$ have? How many distinct polynomials are there?

Solution: 7 polynomials, one for each different possible value of $P(5)$.

2. Suppose we only know $P(0) = 3$. How many possible pairs of $(P(1), P(5))$ are there? How many different polynomials are there?

Solution: 49 polynomials because there are 7 possible values for each of $P(1)$ and $P(5)$

3. If we only know k different points, where $k \leq d$, of a degree d polynomial over GF(p), how many possible polynomials are there?

Solution: There are p^{d+1-k} polynomials. Since we need $d+1$ points to uniquely determine the polynomial, we need to choose values for the $d+1-k$ remaining points that we don't know, and each point can be one of p possible values. For $k = d+1$ there is only 1 possible polynomial.

3 Secret Sharing

3.1 Introduction

1. Set up: n officials indexed from 1 to n. Secret s is an integer. Use $GF(q)$: q is a prime number such that $q > n$ and $q > s$.
2. Scheme: Pick a random $p(x)$ of degree $k-1$ such that $p(0) = s$. Thus,
   (a) Any k officials can use Lagrange Interpolation to find $p(x)$, therefore x.
   (b) Any $k-1$ officials have no information about s.

3.2 Questions

1. Suppose the Oral Exam questions are created by 2 TAs and 3 Readers. The answers are all encrypted and we know that:
   (a) Together, both TAs should be able to access the answers
   (b) All 3 Readers can also access the answers
   (c) One TA and one Reader should also be able to do the same

   Design a secret sharing scheme to make this work.

Solution: Use a 2 degree polynomial which requires at least 3 shares to recover the polynomial. Generate a total of 7 shares, give each Reader a share, and each TA 2 shares. Then, all possible combinations will have at least 3 shares to recover the answer key. Basically the point of this problem is to assign different weights to different classes of people. If we give one share to everyone, then 2 Readers can also recover the secret and the scheme is broken.

4 Erasure Errors

4.1 Introduction

We want to send n packets and we know that k packets could get lost. We use a polynomial under GF(q).

How many more points does Alice need to send to account for k possible errors? __

Solution: k

What degree will the resulting polynomial be? ___

Solution: $n - 1$

How large should q be if Alice is sending n packets with k erasure errors, where each packet is a integer between 0 and m?

Solution: Modulus should be larger than $n + k$ and larger than m and be prime

What would happen if Alice instead send $n+k-1$? Why will Bob be unable to recover the message?

Solution: Bob will receive $n - 1$ distinct points and needs to reconstruct a polynomial of degree $n - 1$. By Fact #3 this is impossible. There are q polynomials of at most degree $n - 1$ in $GF(q)$ that go through the $n - 1$ points that Alice sent.

4.2 Questions

1. Suppose $A = 1$, $B = 2$, $C = 3$, $D = 4$, and $E = 5$. Assume we want to send a message of length 3. Recover the lost part of the message, or explain why it can not be done.
   1. C_AA

      Solution: $P(0) = 3, P(2) = 1, P(3) = 1$. Once we interpolate the polynomial over $\mod 7$, as E is 5, we get $5x^2 + 3x + 3$. Now, once we evaluate this at 1, we get 4. So, in the end, its CDAA.

   2. CE_ _

Solution: Impossible. In order to get the original degree 2 polynomial, we need at least $3 > 2$ points.

2. Suppose we want to send n packets, and we know $p = 20\%$ of the packets will be erased. How many extra packets should we send? What happens if p increases (say to 90%)?

Solution: We want to have $(1-p)*(n+k) = n$, where k is the number of additional packets we send. Solving for k, we get $\frac{n}{1-p} - n$. When p is large, we have to send many times the number of original packets. (fraction packets not erased)*(how many packets are sent) = (number packets in original message)