Week 9

Lecture 1

Polynomial Rings

Lecture 2

ElGamal Encryption

Workshop 9: Workshop based on Lectures in Week 8

Quiz 9

ElGamal Signature

COMP90043

Lecture 2

Udaya Parampalli

Lecture 1

1.1 ElGamal Sigantures

- Main ideas
- Direct Signature Scheme
- Ideas behind ElGamal signature
- ElGamal signature Scheme.
- Other Schemes

Goal: Main ideas

- In RSA, signature and encryption functions are complementary operations.
- Can we find similar feature for ElGamal encryption?
- We looked at the ideas behind the construction of ElGamal Encryption last week.

- How to create a signature algorithm based on the hardness of discrete logarithms and Computational DH problems?

Recap: Requirements of Digital Signatures

- A user's public key should be related his secret using a one way function. We may need to assume some hardness assumption.
- A signature or tag should be able to be created using an efficient algorithm.
- A verification algorithm should be able to be implemented efficiently using the public key by anyone else.
- The signature should have non-repudiation property with a property that once created by a signer he cannot then deny creating the signature.
- Created signatures should be unforgeable by anyone.

More details from Number theory

- Consider q a prime number and a generator "a".
- There are q-1 integers less q which are relatively prime to q.
- The generator "a" is chosen such that
- $a^{(q-1)} = 1 \bmod q$
- a is a generator of the group under multiplication.
- Because of the above property we have:
- $a^{(q-1)} = 1 \bmod q$
- Also, $a^{t(q-1)} = 1 \bmod q$, , for any integer "t". In fact,

$a^{(m)} = 1 \bmod q$, for any integer satisfying $m = 0 \bmod (q-1)$

- We can have a stronger result:
- $a^{(m)} = 1 \bmod q$, if and only if $m = 0 \bmod (q-1)$

More details

- Now consider any integer “i” in the range $1 \leq i < q-1$.

- Consider q (a prime number) and a generator “a” as before: $a^{(q-1)} = 1 \bmod q$
- Because of the above property we have:
- $a^{i+(q-1)} = a^{(i)} = 1 \bmod q$,
- In fact adding any multiple of (q-1) to the exponent does not change the result:
- $a^{i+t(q-1)} = a^{(i)} = 1 \bmod q$, for an integer “t”.

- So, we can have a stronger result:
- $a^{(i)} = a^{(j)} \bmod q$, if and only if $i = j \bmod (q-1)$

.

Direct Digital Signature

- What is the direct digital signature?
- This is attributed to a signature scheme involving only the sender and receiver.
- Here the authenticity of the public key of the source is assured for the destination.
- The scheme is valid depending on the security of the private key. Hence there is a threat that sender could claim that the key is compromised. Such risks could be avoided by having a tighter control on the keys. For example, a requirement of reporting key compromise to a central authority could be included in the policy.

How does ElGamal Signature work?

- As before, each user (eg. Alice) generates their key:
  - chooses a secret key (number): $1 < x_A < q-1$
  - compute her **public key**: $y_A = a^{x_A} \bmod q$
- We would like signature generation depends on the secret x_A and possibly some new secret.
- The signature should depend on the message:
- It needs to embed to exponents (modulo q-1) as others need verify through public address.
- The verification should be effected using only public parameters.
- Everyone known the finite field (modulo q) on which the scheme is defined.
- Before going through the scheme, we will look at the basic ideas behind the scheme.

An Essential Idea

- Idea: If $x' = (x1 + x2) \bmod (q-1)$, (a^{x1}) and (a^{x2}) are given by a user, without revealing x1 and x2,
- then this can be verified by checking the following equation
- $a^{x'} = (a^{x1})(a^{x2}) \bmod (q)$

- Note that , only the person who knows x1 and x2 could have constructed this sequence: x' , (a^{x1}) and (a^{x2}).

- Next, we give the ElGamal Signature idea.

ElGamal Signature Idea

- Given, Alice's public key $y_A = a^{x_A}$ mod q, and the prvate key $1 < x_A < q-1$ and a message M,
- Let m = H(m) = Hash of the message M.
- We would like m to be related to private and public information.
- m = Function(x_A ,y_A, S1, S2)
- When LHS ad RHS are applied as exponents of x, the verification should involve only public parameters.
- $S_1 = \quad a^k \bmod \quad q$
- $k\, S_2 \quad + \quad x_A\, S_1 = m \bmod \quad (q-1)$
- Take a th power of LHS and RHS:
- $(S_1)^{S2} \; (y_A)^{S1} = (a^m) \bmod \quad q$
- So, we have arrived at a verification equations only involving public parameters. This is an ElGamal signature!

ElGamal Digital Signatures

- We will give now a version given in the textbook.
- Each user (eg. A) generates their key
  - chooses a secret key (number): $1 < x_A < q-1$
  - compute her **public key**: $y_A = a^{x_A} \bmod q$
- Alice signs a message M to Bob by computing:
- the hash $m = H(M)$, $0 <= m <= (q-1)$ and gcd(k, q-1) =1.
- compute temporary key: $S_1 = a^k \bmod q$
- compute k^{-1} the inverse of k mod (q-1)
- compute the value: $S_2 = k^{-1}(m - x_A S_1) \bmod (q-1)$
- signature is: (S_1, S_2)
- any user B can verify the signature by computing:
- $V_1 = a^m \bmod q$; $V_2 = y_A^{S1} (S_1)^{S2} \bmod q$
- Signature is valid if $V_1 = V_2$

Example

- Consider GF(19), q =19 and a =10.
- Alice chooses x_A=16 and and hence y_A=10^16 = 4.
- Show the calculations for m =14.

Other related Signature Algorithms

- The textbook gives some related signature algorithms: Schnorr and Digital Signature Algorithm (DSA).
- Schnorr is an efficient signature algorithm.
- DSA is a standard signature algorithm introduced by NIST. I will provide details from the textbook as a additional material.

- Works in subgroup of a larger finite field.
- Works over a large finite field Z_p. p: 1000 bits long.
- Maximum size of the cyclic group = p-1.
- We will ensure that p-1 has a large prime factor q (160 bit long). Hence q divides (p-1).
- We will choose a generator of the subgroup (g).
- Then $g^{(q)} = 1 \bmod p$.
- Now we can redefine ElGamal idea over the subgroup:
  - Signing equations involve modulo q
  - Verifications are over mod p;
- DSA follows a similar strategy with some modifications.

Week 9

Lecture 1

Polynomial Rings

Lecture 2

ElGamal Encryption

Workshop 9: Workshop based on Lectures in Week 8

Quiz 9