Public Key Cryptography

Indivar Gupta
SAG, DRDO, Delhi

INTERNATIONAL CRYPTO-WEBINAR, 2020
26th - 30th August, 2020

Outline

1. Background
   - Basic Concepts in Complexity Theory
   - Some Number Theoretic & Algebraic Algorithms
   - Some Computational Hard Problems and their Application
2. Cryptography
3. Public Key Cryptography
   - Diffie Hellman Key Exchange Protocol
   - Public Key Cryptosystems
   - RSA & ElGamal
   - Elliptic Curve Cryptography
   - Other Public Key Cryptosystems: Post Quantum
4. Elliptic Curve Cryptography
5. Open Source Libraries

Outline

Indivar Gupta
SAG, DRDO, Delhi

Representation of Numbers

Indivar Gupta
SAG, DRDO, Delhi

- Each number can be represented in the form of different basis.
- Any number n between b^{k-1} and b^k is a k-digit number to the base b.
- Number of digits = $[\log_b n] + 1$ (basis b)
- Number of bits (size of number) $[\log_2 n] + 1$

Complexity Theory: I

Indivar Gupta
SAG, DRDO, Delhi

Definition (Running Time)

The number of basic (primitives) operations (steps) to execute an algorithm (computational complexity). *Running time of an algorithm is depend on the size of the input.*

Definition (Size of an Input)

In bits, in digits, in bytes, in words etc.....

Definition (Space Complexity)

It measures the amount of temporary storage used when performing a computational task.

Definition (Big- O)

Complexity Theory: II

Indivar Gupta
SAG, DRDO, Delhi

$f(n) = O(g(n))$ if there exists a positive constant c and a positive integer $n_\circ$ such that $0 \leqslant f(n) \leqslant cg(n)$ for all $n \geqslant n_\circ$. Note: g is simpler function than f and it does not increase much faster than f.

Some Examples of Big-O

- Let $f(n) = 2n^3 + 3n^2 + 4n + 5$ & $g(n) = n^3$. Then $f = O(g)$,for take $n_0 = 5, c = 3$.
- Let $f(n) = a_k n^k + a_{k-1} n^{k-1} + \cdots + a_0$ with $a_k > 0$, $f(n) = O(n^k)$.
- $x^n = O(e^n)$ for any positive power n

Complexity Theory: III

Indivar Gupta
SAG, DRDO, Delhi

Note: The notation Big-O is used to represents an upper bound of the computational complexity of an algorithm in the *worst-case scenario*.

Definition (Small-o)

$f(n) = o(g(n))$ if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} \to 0.$$

Note: $g(n)$ is upper bound of $f(n)$ i.e $f(n) << g(n)$.

Some Examples of small-o

- For any positive integer a, $a^n = o((n!))$
- $n! = o(n^n)$

Remark 1: : Other notations : Ω, ω, Θ.

Orders of common functions[2]

$O(1)$	Constant
$O(\log \log n)$	Double-Logarithmic
$O(\log n)$	Logarithmic
$O(\log^c n), c > 1$	Poly-Logarithmic
$O(n^c), 0 < c < 1$	Fractional
$O(n)$	Linear
$O(n^c), c > 1$	Polynomial
$L_n(\alpha, c)$[1]	Sub-exponential
$O(c^n)$	Exponential
$O(n!)$	Factorial

[1] $O(exp(c + O(1))(n)^{\alpha}(\log n)^{1-\alpha})$

[2] http://en.wikipedia.org/wiki/Big_O_notation

Complexity of an Algorithm I

Indivar Gupta
SAG, DRDO, Delhi

- Complexity of an algorithm is said to be in polynomial time if its complexity is $O(n^c)$, where n is the bit length of the input, & $c > 1$.
- Algorithms with complexity of form $c^{f(n)}$, where $c > 1$ & f is a polynomial in n are called exponential time algorithm.
- A sub-exponential algorithm is one for which the time complexity is in between of polynomial and exponential $(L_n(\alpha, c))$

Complexity of an Algorithm II

Indivar Gupta
SAG, DRDO, Delhi

Definition

A decision problem is said to be in class P if it can be solved in polynomial time.

Example

Instance: $n \in Z^+$
Question: Is n prime?
Answer: Yes, $[O(\log n)^6$ using AKS algo]

Definition

A decision problem is said to be in class NP if 'yes' answer can be verified in polynomial time.

Complexity of an Algorithm III

Indivar Gupta
SAG, DRDO, Delhi

Definition

A decision problem is said to be in class *co-NP* if 'no' answer can be verified in polynomial time.

Example

Instance: $n \in Z^+$
Question: Is n composite?

Definition

L_1 and L_2 be two decision problems. L_1 is said to poly-time reduce to L_2, written $L_1 \leqslant_p L_2$, if there is an algorithm that solves L_1 using only polynomial calls to an algorithm for solving L_2 as a subroutine. This means a polynomial time algorithm for L_2 implies a polynomial time algorithm for L_1.

Complexity of an Algorithm IV

Indivar Gupta
SAG, DRDO, Delhi

Example

$QRP \leqslant_P IFP$

Definition

A decision problem L is said to be NP-complete if

- $L \in NP$
- $L_1 \leqslant_P L$ for every $L_1 \in NP$.

Example

Subset Sum Problem is **NP** complete problem: given a set of positive integers $\{a_1, a_2, \cdots a_n\}$ and a positive integer s, determine whether or not there is a subset of the a_i that sum to s.

Complexity of an Algorithm V

Indivar Gupta
SAG, DRDO, Delhi

Definition

A decision problem is said to be *NP*-hard if any *NP*-complete problem polynomially reduces to it.

Examples

- Computational version of subset sum problem is **NP**-hard problem.

Important Topics of Number Theory

Indivar Gupta
SAG, DRDO, Delhi

- The Euclidean Algorithm
- Modular Arithmetic
- Arithmetic functions and their Properties
- Solution of Linear and Quadratic Congruences, CRT
- Primality testing and Factorization

The Euclidean Algorithm

Indivar Gupta
SAG, DRDO, Delhi

The Euclidean Algorithm finds the greatest common divisor of two integers a and$b$.
For example, if we want to find $gcd(287, 91)$, we divide 287 by 91:

$$287 = 91 * 3 + 14 .$$

We have if $a|b$ and $a|c$, then $a|(b + c)$.

$$gcd(287, 91) = gcd(91, 14)$$

$$gcd(91, 14) = gcd(14, 7)$$

$$gcd(14, 7) = 7$$

Therefore, $gcd(287, 91) = 7$.

Indivar Gupta
SAG, DRDO, Delhi

- **Theorem** The $gcd(a, b)$ is the least positive value of $ax + by$, where x and y range over all integers.
- **Theorem** An integer solution (x, y) of equation $ax + by = c$ exists if and only if c is divisible by $gcd(a, b)$.

- $gcd(a, b, c) = gcd(gcd(a, b), c)$

Number Theoretic Algorithms[3]

Indivar Gupta
SAG, DRDO, Delhi

Complexity of Basic Operations in $\mathbb{Z}_n$

Operations		Complexity
Modular Addition	$(a+b) \mod n$	$O(\log n)$
Modular Subtraction	$(a-b) \mod n$	$O(\log n)$
Modular Multiplication	$(a.b) \mod n$	$O((\log n)^2)$
Modular Inversion	$a^{-1} \mod n$	$O((\log n)^2)$
Modular Exponentiation	$a^k \mod n, k < n$	$O((\log n)^3)$

[3] Chapter 2, [2]

Solution of the Congruence

Indivar Gupta
SAG, DRDO, Delhi

Consider $f(x) \equiv 0 \pmod{m}$, $f(x)$ being a polynomial with integer coefficients. Any $n \in \mathbb{Z}$ is called a solution of the congruence if $f(n) \equiv 0 \pmod{m}$.
The solution is not unique as for any k, such that $n \equiv k \pmod{m}$,

$$\begin{aligned} & f(k) \equiv f(n) \pmod{m} \\ \Longrightarrow\ & f(k) \equiv 0 \pmod{m} \end{aligned}$$

Thus every congruence having one solution has infinitely many solutions.
The congruence $ax \equiv b \pmod{m}$ is called a linear congruence.

Euler Fermat Theorem

Indivar Gupta
SAG, DRDO, Delhi

We define Euler's totient function $\phi(m)$ as

$$\phi(m) = \begin{cases} 1 \text{ when } m = 1, \\ \text{number of positive integers that are } \leq m \text{ and co-prime to } m \end{cases}$$

Euler Fermat Theorem For $(a, m) = 1$, we have

$$a^{\phi(m)} \equiv 1 \mod m.$$

Fermat Little Theorem

Indivar Gupta
SAG, DRDO, Delhi

Fermat Little Theorem is a corollary to Euler-Fermat Theorem.
Let p be a prime number. If $(a, p) = 1$ then from Euler Fermat theorem, we have $a^{\phi(p)} \equiv 1 \pmod{p}$ but $\phi(p) = p - 1$, therefore $a^{p-1} \equiv 1$.
Multiplying both side by a, we get $a^p \equiv a \pmod{p}$.

Otherwise, if $(a, p) \neq 1$, then $p|a$ and $a \equiv 0 \pmod{p}$.
$\implies a^p \equiv 0 \pmod{p}$
$\implies a^p \equiv a \pmod{p}$
Which is Fermat Little Theorem.

The Chinese Remainder Theorem (CRT)

Indivar Gupta
SAG, DRDO, Delhi

Let $m_1, m_2, \cdots, m_r$ be pairwise mutually prime positive integers. For arbitrary integers $b_1, b_2, \cdots, b_r$, the system of linear congruences

$$\begin{aligned} x &\equiv b_1 \pmod{m_1} \\ x &\equiv b_2 \pmod{m_2} \\ &\vdots \\ x &\equiv b_r \pmod{m_r} \end{aligned}$$

has a unique solution $x = \sum_{k=1}^{r} b_k M_k M_{k'}$ modulo m,
where $M = m_1 m_2 \cdots m_r$ and $M_k = M/m_k$, $M_k M_{k'} \equiv 1 \pmod{m_k}$.

CRT Problem

Indivar Gupta
SAG, DRDO, Delhi

To find the least value of x which satisfy,

$$x \equiv 2 \pmod 3$$
$$x \equiv 3 \pmod 5$$
$$x \equiv 2 \pmod 7$$

The calculations are given in the following table.

b_k	m_k	M_k	$M_k(modm_k)$	$M_{k'}$	$b_k M_k M_{k'}$
2	3	35	2	2	140
3	5	21	1	1	63
2	7	15	1	1	30
				sum	233

The least number of things are obtained by reducing 233 (mod 105) i.e. 23.

Quadratic Residues

Indivar Gupta
SAG, DRDO, Delhi

Consider the quadratic congruences of the type

$$x^2 - n \equiv 0 \pmod{p} \qquad \ldots(1)$$

where p is a prime. We see that the congruence can have at most 2 solutions.

If x is a solution, then $(-x)$ is also a solution. Thus, (1) has either two solutions or none. When (1) has solutions, we say that n is a quadratic residue modulo p and we write nRp, otherwise if (1) has no solution then we say that n is a quadratic non-residue modulo p and we write $n\bar{R}p$.

The Legendre's symbol $\left(\frac{n}{p}\right)$ is defined as

$$\left(\frac{n}{p}\right) = \begin{cases} 0 \text{ if } p|n, \\ 1 \text{ if } nRp, \\ -1 \text{ if } n\bar{R}p. \end{cases}$$

Jacobi Symbols

Indivar Gupta
SAG, DRDO, Delhi

If P is an odd positive integer with prime factorization

$$P = \prod_{i=1}^{r} p_i^{a_i}$$

The Jacobi symbol $\left(\frac{n}{P}\right)$ is defined as,

$$\left(\frac{n}{P}\right) = \prod_{i=1}^{r} \left(\frac{n}{p_i}\right)^{a_i}, \quad if\ P > 1$$

and

$$\left(\frac{n}{1}\right) = 1$$

Primitive Roots

Indivar Gupta
SAG, DRDO, Delhi

Let a and m be relatively prime and $m > 1$. Then a is called a primitive root modulo m if

$$a^k \not\equiv 1 \; mod \; m$$

whenever $k < \phi(m)$.

Theorem:
There are exactly $\phi(p-1)$ primitive roots modulo p.

Computing Euler's function

Indivar Gupta
SAG, DRDO, Delhi

- If p is a prime, then for integers $k \geq 1$

$$\phi(p^k) = p^{k-1}(p-1)$$

- If m and n are co-prime, then

$$\phi(mn) = \phi(m).\phi(n)$$

- Therefore, the value of ϕ(n) for any positive integer n can be computed by writing prime factorization of n, for $n = p_1^{a_1}.p_2^{a_2}.....p_t^{a_t}$

$$\phi(n) = (p_1-1).p_1^{a_1-1}.(p_2-1).p_2^{a_2-1}....(p_t-1).p_t^{a_t-1}$$

And so,

$$\phi(n) = n.\prod_{p|n}(1-\frac{1}{p})$$

Some Primality Testing Algorithms [4]

Indivar Gupta
SAG, DRDO, Delhi

Algorithms	Type	Complexity
Jacobi Some test	True Primality Test	$O(k(\log n)^{\log\log\log n})$
Elliptic Curve Primality Proving test (ECPP)	-do-	$O(\log n)^{6+\epsilon}$
Agrawal, Kayal, and Saxena Test	-do (deterministic)-	$O((\log n)^{10.5}$
Miller-Rabin test	Probabilist Primality Test	$O((t\log n)^3)$, (t modular exp)
Solovay-Strassen Test	Probabilist Primality Test	$O((t\log n)^3)$, (t modular exp)
Fermat's test	Probabilist Primality Test	$O((t\log n)^3)$, (t modular exp)

[4] Chapter 4, [2]

Fermat's Primality Test

Indivar Gupta
SAG, DRDO, Delhi

Fermat's Little theorem for p prime and an integer a such that $p \nmid a$ says that $a^{p-1} \equiv 1 \; mod \; p$.

If $a^{p-1} \not\equiv 1 \; mod \; p$ for some a then p is composite. Thus, if $a^{p-1} \equiv 1 \; mod \; p$ for one or more values of a, then we say that p is probably a prime.

Example: Fermat pseudo-prime: $n = 341$**,**

$$2^{340} \equiv 1 \mod 341 (11 \times 31)$$

Fermat's Primality Test: Algorithm

Indivar Gupta
SAG, DRDO, Delhi

Input: n - an integer to test for primality, $n > 3$, k - a parameter that determines the number of times to test for primality.

Output: Composite if n is composite.
Otherwise, probably prime.

Repeat k times.

- Pick 'a' randomly in the range $[2, n-2]$
- If $a^{p-1} \not\equiv 1 \; mod \; p$, then return composite.
- If composite is never returned: return probable prime.

Solovay-Strassen Test

Indivar Gupta
SAG, DRDO, Delhi

Concept:

- Euler proved that for an odd prime number p and any integer a, $a^{p-1/2} \equiv \left(\frac{a}{p}\right) \mod p$, where $\left(\frac{a}{p}\right)$ is Legendre symbol.
- Given an odd number n, we can contemplate weather or not the congruence $a^{n-1/2} \equiv \left(\frac{a}{n}\right) \mod n$, holds for various values of the base a. If n is prime, then this congruence holds for all values of a.
- So, if we pick a value of a at random and test the congruence then as soon as we find an a which does not fit the congruence we know that n is not prime.
- Example of Euler Pseudo-prime is $91 = (7 \times 13)$ to the base 9: $9^{45} \equiv (\frac{9}{91}) \equiv 1$

Solovay-Strassen Test: Continued –

Indivar Gupta
SAG, DRDO, Delhi

Pick a random integer $a < n$, and do the followings:

- If $gcd(a, n) > 1$, then return composite.
- If $\left(\frac{a}{n}\right)$ is not equal to $a^{n-1/2}$, then return composite.
- Else, return probable (pseudo) prime.

Indivar Gupta
SAG, DRDO, Delhi

Exercises

- Complexity for computation of Jacobi /Legendre Symbol.
- Complexity for finding square root of Quadratic Residue Modulo p.
- Complexity for getting solution the system of linear congruences using Chinese Remainder Theorem.

Important Topics of Finite Fields

Indivar Gupta
SAG, DRDO, Delhi

- Irreducible and Primitive Polynomials, and methods for testing them.
- Construction of Finite Fields and Basis
- Arithmetic over Finite Fields
- Factorization of polynomial over finite fields
- Combinatorial Problems over Finite Fields

Field

Indivar Gupta
SAG, DRDO, Delhi

Definition

A field $(\mathbb{F}, +, \cdot)$ *consists of a set* $\mathbb{F}$ *along with 2 binary operations* $+$ *&* $\cdot$ *on* $\mathbb{F}$ *satisfying the following conditions*

i. $(\mathbb{F}, +)$ *is a commutative group,*

ii. $(\mathbb{F} \setminus \{0\}, \cdot)$ *is also a commutative group and*

iii. *The operation* $\cdot$ *is distributive over* $+$.

Finite Fields

Indivar Gupta
SAG, DRDO, Delhi

- A **finite field** is a field $\mathbb{F}$ which contains a finite number of elements.
- If $\mathbb{F}$ is a finite field, then $\mathbb{F}$ contains p^m elements for some prime p and integer $m \geq 1$.
- For every prime power order p^m, there is a ! finite field of order p^m. This field is denoted by $\mathbb{F}_{p^m}$, or sometimes by $GF(p^m)$.
- For $m = 1$, $\mathbb{F}_p$ or $GF(p)$ is a field. If p is a prime then $\mathbb{Z}_p$ is a field.

$$\mathbb{F}_p \cong GF(p) \cong \mathbb{Z}_p.$$

Construction of Finite Field of Order p^m

Indivar Gupta
SAG, DRDO, Delhi

- First select an irreducible polynomial $f(x) \in \mathbb{Z}_p[x]$ of degree m.
- The ideal $< f(x) >$ is e a maximal ideal.
- Then $Z_p[x]/ < f(x) >$ is a finite field of order p^m.
- For each $m \geq 1$, $\exists$ a monic irreducible polynomial of degree m over $\mathbb{Z}_p$.
- Hence, every finite field has a polynomial basis representation.

Theorem

The number of monic irreducible polynomials in $\mathbb{F}_q[x]$ of degree n is given by

$$\frac{1}{n}\sum_{d|n}\mu(d)q^{n/d},$$

where μ is Möbius function.

Finite Fields $GF(2^3)$

Indivar Gupta
SAG, DRDO, Delhi

- Consider an irreducible polynomial $x^3 + x + 1$ over $GF(2)$
- $GF(2)[x]/ < x^3 + x + 1 >=$
  $\{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$
- One to one correspondence between $GF(2^3)$ and $\mathbb{Z}_8$.
  Similarly, $GF(2^3)$ maps all of the polynomials over $GF(2)$ to the 8 polynomials shown above.

Finite Field Basis I

Indivar Gupta
SAG, DRDO, Delhi

☞ $GF(p^m)$ is a vector space over $GF(P)$ of dimension m.
☞ Any set of m Linearly Independent elements can be used as a basis.

1. Polynomial Basis: If α is roots of generating polynomial $f(x) = 0$ with $GF(P^m)^\star =< \alpha >$ then $\{0, \alpha^0, \alpha^1, \cdots \alpha^{n-1}\}$ will be basis of $GF(p^m)$.
2. Normal Basis: : Basis of the form $\{\alpha, \alpha^p, \cdots \alpha^{p^{m-1}}\}$ is called normal basis, where α is a normal element.

Finite Field Basis II

Indivar Gupta
SAG, DRDO, Delhi

Polynomial Basis			Normal Basis			$\alpha^7 = 1$ Exp.
1	α	α^2	α^3	α^6	α^5	
0	0	0	0	0	0	0
1	0	0	1	1	1	$1 = \alpha^0$
0	1	0	0	1	1	α
0	0	1	1	0	1	α^2
1	1	0	1	0	0	α^3
0	1	1	1	1	0	α^4
1	1	1	0	0	1	α^5
1	0	1	0	1	0	α^6

Figure: Representation of $GF(2^3), f(x) = x^3 + x + 1$

Complexity of Arithmetic Operations in $\mathbb{F}_{p^m}$[5]

Indivar Gupta
SAG, DRDO, Delhi

Complexity wrt Polynomial Basis Representation

Operations		Complexity in terms of $\mathbb{Z}_p$-operations
Addition	$(g(x)+h(x)) \mod f(x)$	$O(m)$
Subtraction	$(g(x)+h(x)) \mod f(x)$	$O(m)$
Multiplication	$(g(x).h(x)) \mod f(x)$	$O(m^2)$
Inversion	$g(x)^{-1} \mod f(x)$	$O(m^2)$
Exponentiation	$g(x)^k \mod f(x), k < p^m$	$O((\log p)m^3)$

* $f(x)$ is irreducible polynomial of degree m.

Exercises

- Running time complexity of Extended Euclidean Algorithm in $\mathbb{Z}_p[x]$

[5] Chapter 2, [2]

Hard Problems and One-way Function I

Indivar Gupta
SAG, DRDO, Delhi

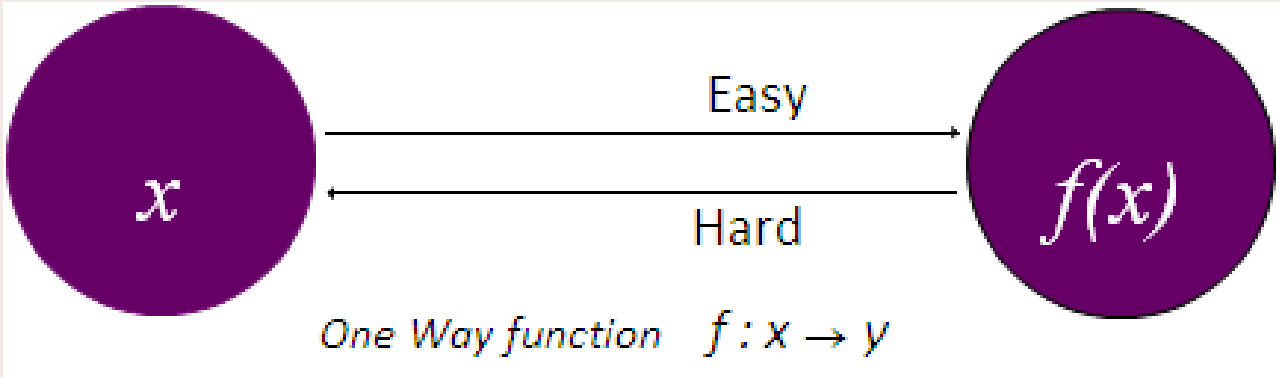
PKC is based on

- Mathematical problems believed to be *hard* to solve.
  Hard means computationally infeasible
  Hard=2^{128} or more operations : ECC-256,
- Trapdoor one-way function

Hard Problems and One-way Function II

Indivar Gupta
SAG, DRDO, Delhi

Definition (One-way Function)



Easy: There exists a polynomial-time algorithm that gives $y = f(x)$.

Hard: For given any $y = f(x)$ it is computationally infeasible to find x.

Hard Problems and One-way Function III

Indivar Gupta
SAG, DRDO, Delhi

Definition (trapdoor one-way function)

A trapdoor one-way function is a one-way function with the additional property that given some extra information (called the trapdoor information) it becomes feasible to find for any given $y \in Im(f)$, an $x \in X$ such that $f(x) = y$.

Some Computationally Hard Problems I

Indivar Gupta
SAG, DRDO, Delhi

- Integer factorization (RSA)
- Subset sum problem (Knapsack)
- Quadratic Residuosity Problem (Rabin)
- DLP in finite fields (El Gamal)
- DLP in elliptic curve over finite fields (ECC) (ElGamal, ECIES, ECDSA)
- CDHP, DDHP, GDHP, GDLP
- Conjugacy Search Problem
- Group action and Semi-group action problem
- BDHP (Joux Protocol)
- Factorization with Discrete Logarithm Problem (Over Group Ring)
- Discrete logarithm with conjugacy Search Problem

Some Computationally Hard Problems II

Indivar Gupta
SAG, DRDO, Delhi

- Solving system of multivariate polynomial equations over finite fields and isomorphism of polynomials (HFE, Sflash, Quartz)
- Decoding Problem: GBD/GD (Code based Cryptography: McEliece's Crypto-system)
- Lattice basis reduction (NTRU, FALCON)
- Hash Based PKC (Gravity-SPHINCS), PICNIC, SPHINCS+
- Computational Supersingular Isogeny Problem

Integer factorization (RSA)

Indivar Gupta
SAG, DRDO, Delhi

N = Large Composite Integer. IFP: Find factor of N.

Sub-exponential Algorithms

Methods	Running Time
Continued Fraction Method	$T(CFRAC) = O\big((L_N(1/2, \sqrt{2})\big)$
Lenstra's Elliptic Curve Method	$T(ECM) = O\big(L_p(1/2, \sqrt{2}).(\log N)^2)\big)$
Multiple Polynomial Quadratic Field Sieve Method	$T(MPQS) = O\big(L_N(1/2, 3/2\sqrt{2})\big)$
General Number Field Sieve Method	$T(GNFS) = O\big(L_N(1/3, \sqrt[3]{64/9})\big)$
Special Number Field Sieve Method	$T(SNFS) = O\big(L_N(1/3, \sqrt[3]{32/9})\big)$

Integer factorization (RSA)

Indivar Gupta
SAG, DRDO, Delhi

Other Integer Factorization Algorithms (Exponential)

Methods	Running Time
Lehman's Method	$O(N^{\frac{1}{3}+\epsilon})$
Shanks' Square Form Factorization Number	$O(N^{\frac{1}{4}})$
Shanks' Class Group Method	$O(N^{\frac{1}{5}+\epsilon})$

DLP in finite fields (El Gamal)

Indivar Gupta
SAG, DRDO, Delhi

Definition (Discrete Logarithm Problem (DLP))

- Given $y \in G$, The DLP in G is to find the unique $x \in Z_N$ such that $y = g^x$. Such an x is called the discrete log of y with respect to base g.

DLP in finite fields (El Gamal)

Indivar Gupta
SAG, DRDO, Delhi

Algorithm for Solving DLP

Methods	Running Time / Space
Shanks' Baby-Step Giant-Step Algorithm	$O(\sqrt{N}\log N)$ (group Operations) / $O(\sqrt{N})$
Silver-Pohlig-Hellman, $N = \prod_{i=1}^{i=k} p_i^{\alpha_1}$	$O(\sum_{i=1}^{i=k} \alpha_i(\log N + \sqrt{p_i}))$ (group Operations) /Comparable
Pollard's Rho	$O(\sqrt{N})$ (group Operations) /Negligible

DLP in finite fields (El Gamal)

Indivar Gupta
SAG, DRDO, Delhi

Algorithm for Solving DLP

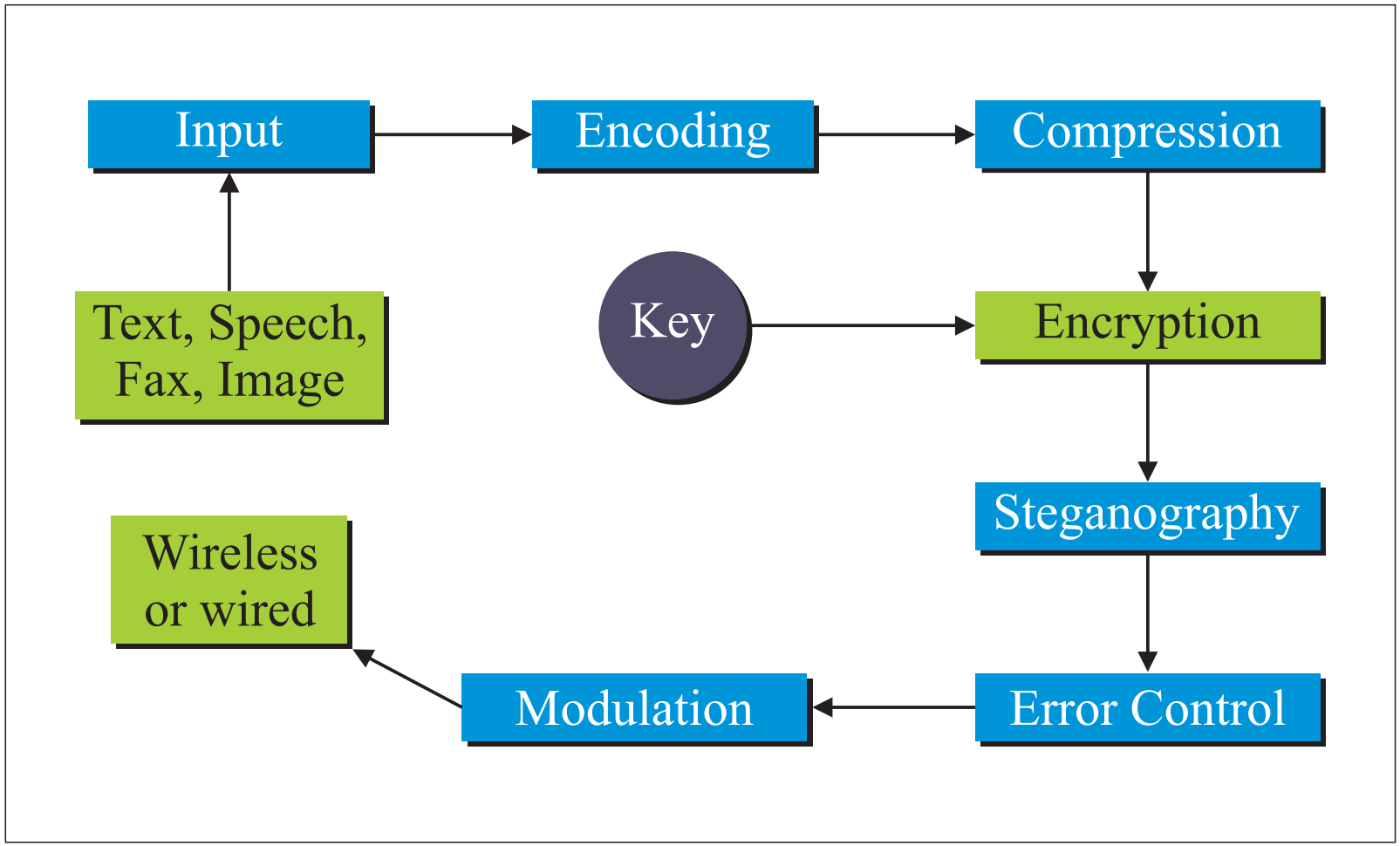
Adleman's Index Calculus Algorithm for $\mathbb{F}_p^\star$	$O\left(L_p(1/2,c)\right)$
Coppersmith et al Algo (for $\mathbb{F}_p^\star$)	$O\left(L_p(1/2,c)\right)$ (c is smaller than Adleman's Algo)
Index Calculus Algorithm for $GF(q)$: $q=p^m$	$O\left(L_q(1/2,c)\right)$
Coppersmith Index Calculus Algorithm for $GF(2^m)$	$O\left(L_{2^m}(1/3,c)\right)$
General Number Field Sieve Algorithm (for $GF(q)$)	$O\left(L_q(1/3,c)\right)$, $c=(64/9)^{1/3}$

Outline

Indivar Gupta
SAG, DRDO, Delhi

Cryptography for Secure Communication

Indivar Gupta
SAG, DRDO, Delhi



Cryptography for Secure Communication

Indivar Gupta
SAG, DRDO, Delhi

Input to Source

- Text
- Speech
- Picture/Movie
- Fax

Example

- Text $\Rightarrow$ ASCII
- Speech $\Rightarrow$ PCM
- Picture/Movie $\Rightarrow$ JPEG, MPEG
- FAX $\Rightarrow$ Modified Huffman Coding

Cryptology

Indivar Gupta
SAG, DRDO, Delhi

- **Cryptography** Design of algorithms, systems, protocols which are used to protect information against specific threats. PRIVACY, AUTHENTICATION, INTEGRITY & NON-REPUDIATION.
- **Cryptanalysis** Evolving mathematical methods to check if the specified cryptographic design achieves the desired security goal. It is a science which deals with how to defeat of achieving '*Security Goals*'.
- Cryptology = Cryptography + Cryptanalysis

Cryptology I

Indivar Gupta
SAG, DRDO, Delhi

- **Symmetric Key Crypto :** $\rightarrow$ Alice and Bob both agree on common key.
- **Public Key Crypto :** $\rightarrow$ PKC use a pair of keys $\longrightarrow$ public key, private key. Computing the private key form public key has to be intractable.
- **Hash Function:** $\rightarrow$ function that maps a bit string of arbitrary length into an output of fixed number of bits, called message digest or hash value.
- **Digital Signature:** A digital signature is a data string which associates a message (in digital form) with some originating entity.
- **Non-repudiation:** An entity should not be allowed to deny valid signatures made by him.

Cryptology II

Indivar Gupta
SAG, DRDO, Delhi

- **Data Origin Authentication /Message-Authentication:** techniques provide to receiver which receives a message assurance of the identity of the party which originated the message.
- **Entity Authentication/Identification:** Alice proves her identity to Bob. Alice demonstrates to Bob her knowledge of a secret piece of information.

Outline

Introduction

Indivar Gupta
SAG, DRDO, Delhi

- PKC developed by Diffie-Hellman and Merkle in the mid 1970s.
- In PKC, each user has pair of keys: Private Key and Public Key.
- Every one has access to the public key but private information can be accessed by only the owner.
- PKC depends on computationally hard problems that prevent inverting the public map.
- Computing the private key form public key has to be intractable.

Public-Key Cryptosystems

Indivar Gupta
SAG, DRDO, Delhi

Public Key Cryptosystem CS is a five-tuple

$$CS = (\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$$

- $\mathcal{P}$: Plaintext Space, $\mathcal{C}$: Ciphertext Space
- $\mathcal{K}$: Key Space
- $\mathcal{E}$: Encryption Rule, $\mathcal{D}$: Decryption Rule
- $E_e : \mathcal{P} \rightarrow \mathcal{C}, E_e(M) = C, D_d : \mathcal{P} \rightarrow \mathcal{C}, D_d(C) = M$ $(e,d) \in \mathcal{K}, E \in \mathcal{E}, D \in \mathcal{D}$.
- It is ***computationally infeasible*** to compute d from e.
- $D_d(E_e(M)) = M$.
- E and D operations should be efficiently computable.

Advantages and Disadvantages of PKC

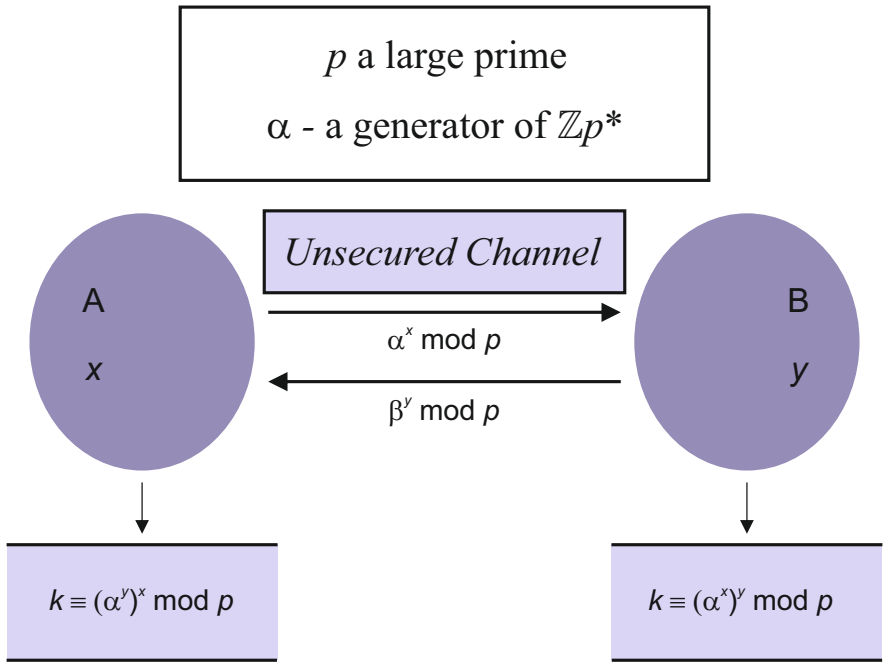
Indivar Gupta
SAG, DRDO, Delhi

- **Key Security:** Only private key needs to be kept secret.
- **Longevity:** Key pairs may be used without change in most cases over long period of time.
- **Key Management:** In a large network fewer private keys will be required.
- **Key-exchange:** No key exchange is required
- **Digital Signature:** The greatest advantage of PKC.
- **Performance:** It is slow, in general.
- **Dependency:** Role of CA, require PKI.
- **System Security:** Depends on well-defined computational hard problems.

Diffie Hellman Key Exchange Protocol

Indivar Gupta
SAG, DRDO, Delhi

Diffie Hellman Key Exchange Protocol I



- k is the shared secret key

Diffie Hellman Key Exchange Protocol

Indivar Gupta
SAG, DRDO, Delhi

- Knowing α, α^x and α^y(but neither x nor y) it is hard to find α^{xy}.
- This is as secure as discrete log is intractable.
- **Idea of this protocol:** The enciphering key can be made public since it is computationally infeasible to obtain the deciphering key from enciphering key.
- This protocol is the ***door-opener*** to Public Key Cryptography.

☞Security: CDHP

Public Key Encryption Schemes

Indivar Gupta
SAG, DRDO, Delhi

- Setup
- Key Generation Algorithm
- Encryption
- Decryption

RSA Cryptosystem

Indivar Gupta
SAG, DRDO, Delhi

Key Generation

- The first task is to select n. n is normally very large (approx 308 digits) which is a product of two large primes p and q.
- Next a large integer e is chosen such that e is relatively prime to $\phi(n)$. e is usually picked as a prime larger than both $(p-1)$ and $(q-1)$.
- Next d is selected in such a way that: $e.d \equiv 1 \mod \phi(n)$.
- n and e are made public.
- p, q and d are kept private.

Encryption and Decryption

Indivar Gupta
SAG, DRDO, Delhi

Encryption

- A obtains B's public key
- Message m is an integer in the interval $[1, n-1]$.
- Compute $c \equiv m^e \mod n$.
- Send the cipher text c to B.

Decryption

- To recover plaintext m from c, B uses his private key d to recover $m \equiv c^d \mod n$.

☞Security: IFP

Example

- Suppose A wants to send the following message to B

$$powera$$

- Let B chooses his $n = 1943 = 29.67$. Then $\phi(n) = 1848$.
- Suppose he picks $e = 701$, then $d = 29$. $\because 26^2 < n < 26^3$, therefore the block size of the plaintext $= 2$.
- $m_1 = po = 15.26 + 14 = 404, m_2 = we = 22.26 + 4 = 576, m_3 = ra = 17.26 + 0 = 42$.
- $c_1 = 404^{701} \equiv 1419(\mod 1943)$. Similarly, A can calculate $c_2 = 344 \& c_3 = 210$.
- Now $c_1 = 1419 = 2.26^2 + 2.26 + 15 = ccp$, $c_2 = 344 = 13.26 + 6 = ng$ & $c_3 = 210 = 8.26 + 2 = ic$ Therefore the cipher text is

$$ccpngic$$

RSA Signature Scheme

Indivar Gupta
SAG, DRDO, Delhi

- Setup
- Key Generation Algorithm
- Signature Generation
- Signature Verification

RSA Signature Scheme

Indivar Gupta
SAG, DRDO, Delhi

- Setup: common parameters: security level, hash $(h : \{0,1\}^* \longrightarrow Z_n)$.
- Key Generation Algorithm : Public Key of Signer (n, e), private key d.
- Signature Generation: $s = h(m)^d \mod n$.
- Signature Verification: $\tilde{m} = s^e \mod n$, verify that $\tilde{m} = h(m)$, if not reject the signature.

Discrete Log: ElGamal

Indivar Gupta
SAG, DRDO, Delhi

Key Generation

- First choose a large prime p such that the DLP is infeasible in $(\mathbb{Z}_p^*, .)$.
- Select a primitive element α of $(\mathbb{Z}_p^*, .)$.
- Pick a random integer $a(1 < a < p-1)$ and compute $\beta \equiv \alpha^a (mod p)$.
- Public Para $= (p, \alpha)$, Public Key$= \beta$ and Private key $= a$.

Encryption and Decryption

Indivar Gupta
SAG, DRDO, Delhi

Encryption

- A obtains B's public information (p, α, β).
- He represents the message m as an integer in $\mathbb{Z}_p^*$.
- He chooses a random integer k in $[2, p-2]$.
- Compute $c = (c_1, c_2) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$, where $c_1 \equiv \alpha^k(\mod p) \& c_2 \equiv m\beta^k(\mod p)$.

Decryption Using B's private key, he decrypts the ciphertext $c = (c_1, c_2)$ by calculating $m \equiv c_2(c_1^a)^{-1}(\mod p)$.

☞Security: DLP

Example of ElGamal

Indivar Gupta
SAG, DRDO, Delhi

- Let $p = 2579$ and $\alpha = 2$, α is a primitive element $\bmod\ 2579$.
- Let $a = 765$ then $\beta \equiv 2^{765}(\ \bmod\ 2579) \equiv 949(\ \bmod\ 2579)$.
- Therefore the public key of B $= (2579, 2, 949)$ and private key is 765.

$$m = 1299$$

- Now suppose A wants to send the message $m = 1299$ to B and A picks the random integer $k = 853$.
- Then A computes $c_1 \equiv 2^{853}\ \bmod\ 2579 \equiv 435\ \bmod\ 2579$ and $c_2 \equiv 1299.949^{853}\ \bmod\ 2579 \equiv 2396(\ \bmod\ 2579)$. Therefore, the ciphertext of A is

$$C = (435, 2396).$$

Digital Signature Algorithm (DSA)

Indivar Gupta
SAG, DRDO, Delhi

☞Adopted by NIST in 1994

- Setup: common parameters: security level, hash $(h : \{0,1\}^* \longrightarrow Z_q)$.
- Key Generation Algorithm:
  1. Select primes p, q with $q/(p-1)$
  2. select $g \neq 1 \in \mathbb{Z}_p^\star$ such that $g^q = 1 \mod p$
  3. Select $x \in \mathbb{Z}_q$ and compute $y = g^x \mod p$
  4. public key (p, q, g, y), secret key x

Digital Signature Algorithm (DSA)

Indivar Gupta
SAG, DRDO, Delhi

- Signature Generation
  1. Choose $k \in \mathbb{Z}_q^\star$,
  2. Compute $r = (g^k \mod p) \mod q$
  3. $s = k^{-1}(H(m) + xr) \mod q$
  4. Output the signature $\sigma = (s, r)$
- Signature Verification:
  1. Check if $r, s \in \{1, \ldots, q-1\}$, otherwise reject signature
  2. set $w = s^{-1} \mod q$, $u_1 = H(m)w \mod q$, $u_2 = rw \mod q$
  3. Compute $v = (g^{u_1} y^{u_2} \mod p) \mod q$
  4. Accept sign if $v = r \mod q$, and reject otherwise.

☞Security: DLP

RSA Factoring Challanges[7]

- Factorization of Giver Number
- Started by RSA Laboratories: March 1991, Ended: 2009
- RSA-768 factored in Dec 2009
- A lot of research have been carried out to factor RSA number beyond 768-bit. [6]

RSA-768=	123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985690214341
	p = 33478071698956898786044169848212690817704794983713768568912431388982883793878002287614711652531743087737814467999489 $\times q$ = 36746043666799590428244633799627952632279158164343087642676032283815739666511279233373417143396810270092798736308917

[6] Cryptology ePrint Archive: http://eprint.iacr.org/2010/006

[7] https://en.wikipedia.org/wiki/RSA_Factoring_Challenge

RSA Factoring Record: Feb 2020

Indivar Gupta
SAG, DRDO, Delhi

RSA-250 has 250 decimal digits (829 bits), and was factored in February 2020 by Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thome, and Paul Zimmermann

RSA-250 (829)=	2140324650240744961264423072839333563008614715144755017797754920881418023447140136643345519095804679610992851872470914587687396261921557363047454770520805119056493106687691590019759405693457452230589325976697471681738069364894699871578494975937497937
	p = 6413528947707158027879019017057738908482501474294344720811685963202453234463023862359875266834770873766192558569463979885336 7 $\times q$ = 3337202759497815655622601060535511422794076034476755466678452098702384172921003708025744867329688187756571898625803693206271 1

Post Quantum Cryptography

Indivar Gupta
SAG, DRDO, Delhi

- If quantum computers are available then Shor's algorithm runs in polynomail time to solve
  - Integer factorization problem
  - DLP in finite fields & DLP on elliptic curves
  - DLP in general class groups
- The following PKC will be dead: RSA, DLP /ECDLP based Cryptosystems
- Cryptography that will be resistant to attack by quantum computer is known as Post Quantum Cryptography

Indivar Gupta
SAG, DRDO, Delhi

Definition (Lattices)

A lattices is defined as the set of all integer combinations of n linearly independent vectors $\mathbf{b_1}, \cdots, \mathbf{b_n}$:

$$\mathfrak{L}(\mathbf{b_1}, \cdots, \mathbf{b_n}) = \left\{ \sum_{i=1}^{i=n} x_i \mathbf{b_i} : x_i \in \mathbb{Z} \text{ for } 1 \leq i \leq n \right\}.$$

The set of vectors $\mathbf{b_1}, \cdots, \mathbf{b_n}$ is called a *basis* for the lattice.
$\mathbf{B} = [\mathbf{b_1}, \cdots, \mathbf{b_n}]$

Lattice basis reduction (NTRU)

Indivar Gupta
SAG, DRDO, Delhi

Lattice Problems

- Shortest Vector Problem (SVP): Given a lattice basis $\mathbf{B}$, find the shortest non zero vector in $\mathfrak{L}(\mathbf{B})$.
- Closest Vector Problem (CVP): Given a lattice basis $\mathbf{B}$ and a target vector $\mathbf{t}$ (not necessarily in the lattice), find the lattice point $\mathbf{v}$ closest to $\mathbf{t}$.

Fact: SVP and CVP are NP-hard

Solving System of Multivariate Polynomial Equations Over Finite Field

Indivar Gupta
SAG, DRDO, Delhi

Definition ($\mathcal{MQ}$ Problem)

Solve the system $p_1(\mathbf{x}) = p_2(\mathbf{x}) = \cdots = p_m(\mathbf{x})$, where each p_i is a quadratic in $\mathbf{x}$. All coefficients and variables are in $\mathbb{F}_q$, the field with q elements.

- Multivariate Public Key Cryptography is based on hardness of $\mathcal{MQ}$.
- $\mathcal{MQ}$ is an NP-Hard Problem
- HFE, Sflash, Quartz

Decoding Problem: GBD/GD

Indivar Gupta
SAG, DRDO, Delhi

Definition (Goppa Bounded Decoding (GBD) Problem)

Instance: An $r \times n$ binary matrix H and word $s \in \mathbb{F}_2^r$.
Question: Is there a word e in $\mathbb{F}_2^n$ of weight $\leq r/\log_2 n$ such that $He^T = s$?

Definition (Goppa Code Distinguishing (GD) Problem)

Instance: An $r \times n$ binary matrix H.
Question: Does H belong to $\mathcal{G}_{n,t}$ for some t. Here $\mathcal{G}_{n,t}$ denotes the set of all parity check matrices of t-error correcting binary Goppa codes of length n.

Decoding Problem: GBD/GD

Indivar Gupta
SAG, DRDO, Delhi

- GBD and GD problem are NP-hard. In fact these problems are exponential.
- Security of Code based Cryptography are based on hardness of GBD/GD problem.
- Examples: McEliece's Cryptosystem and Niederreiter.
- Cryptanalytic Attacks: not susceptible to all known attacks
- Key Size: Very Large
- Encryption/Decryption Speed: Reasonable

Outline

Indivar Gupta
SAG, DRDO, Delhi

Introduction to Elliptic Curve I

Indivar Gupta
SAG, DRDO, Delhi

- An elliptic curve **E** over a field **K** is a non-singular cubic curve in two variables, $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ together with an extra point *the point at infinity*.
- The field **K** is usually taken to be the complex, real, rational, algebraic extensions of rational or a **finite field**.
- Elliptic curves groups for cryptographic applications are examined with the underlying finite fields of characteristic p(where $p > 3$ is a prime) i.e F_{p^m} and fields of characteristic 2 i.e. F_{2^m}.
- The "standard elliptic curve"has the form:

$$y^2 = x^3 + ax + b$$

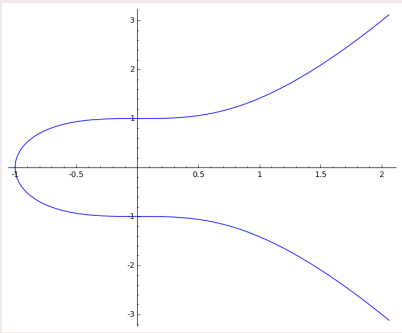
for some fixed a and b.

Introduction to Elliptic Curve II

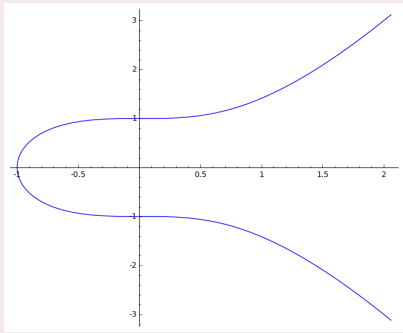
Indivar Gupta
SAG, DRDO, Delhi

Examples for Characteristic 0:

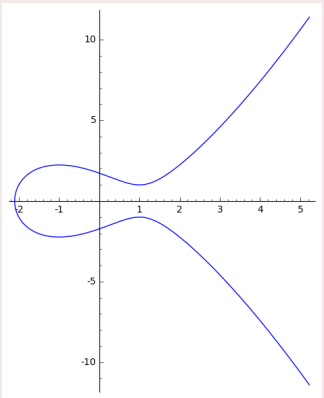
$y^2 = x^3 - 1$



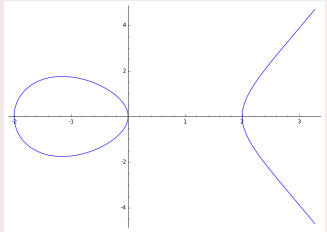
$y^2 = x^3 + 1$



$y^2 = x^3 - 3x + 3$



$y^2 = x^3 - 4x$



Elliptic Curve

Indivar Gupta
SAG, DRDO, Delhi

Generalized Form: An elliptic curve E over a field $\overline{K}$ is defined by a Weiestrass equation: $E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, $a_i \in \overline{K}$ together with a point $\mathcal{O}$ called point at *infinity*.

- E/K- if $a_1, a_2, a_3, a_4, a_6 \in K$
- $E(K)$ the set of K-*rational points* of E, with the point $\mathcal{O}$.

$$E(K) = \{(x,y) \in K^2 : y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6\} \cup \{\mathcal{O}\}. \quad (1)$$

- we define some constants:

$$\left.\begin{aligned} b_2 = a_1^2 + 4a_2,\ b_4 = a_1a_3 + 2a_4,\ b_6 = a_3^2 + 4a_6, \\ b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_2a_4 - a_2a_3^2 - a_4^2, \\ c_4 = b_2^2 - 24b_4, c_6 = -b_2^3 + 36b_2b_4 - 216b_6. \end{aligned}\right\}$$

  discriminant $\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6$.

- $(K) \neq 2, 3, \Delta = (c_4^3 - c_6^2)/1728$.
- The curve $f(x,y) = 0$ is non singular iff at least one of $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \neq 0$. Equivalent condition is Discriminant $\Delta \neq 0$.

- Let E be an elliptic curve given by a Weierstrass equation. Then $(E, +)$ is an abelian group with identity element $\mathcal{O}$ and $E(K)$ is a subgroup of E.

- $P(x_1, y_1)$ and $Q(x_2, y_2)$ be two points on a curve. the explicit formula for $P + Q = (x_3, y_3)$ can be computed easily. If the curve is defined in equation (1), then

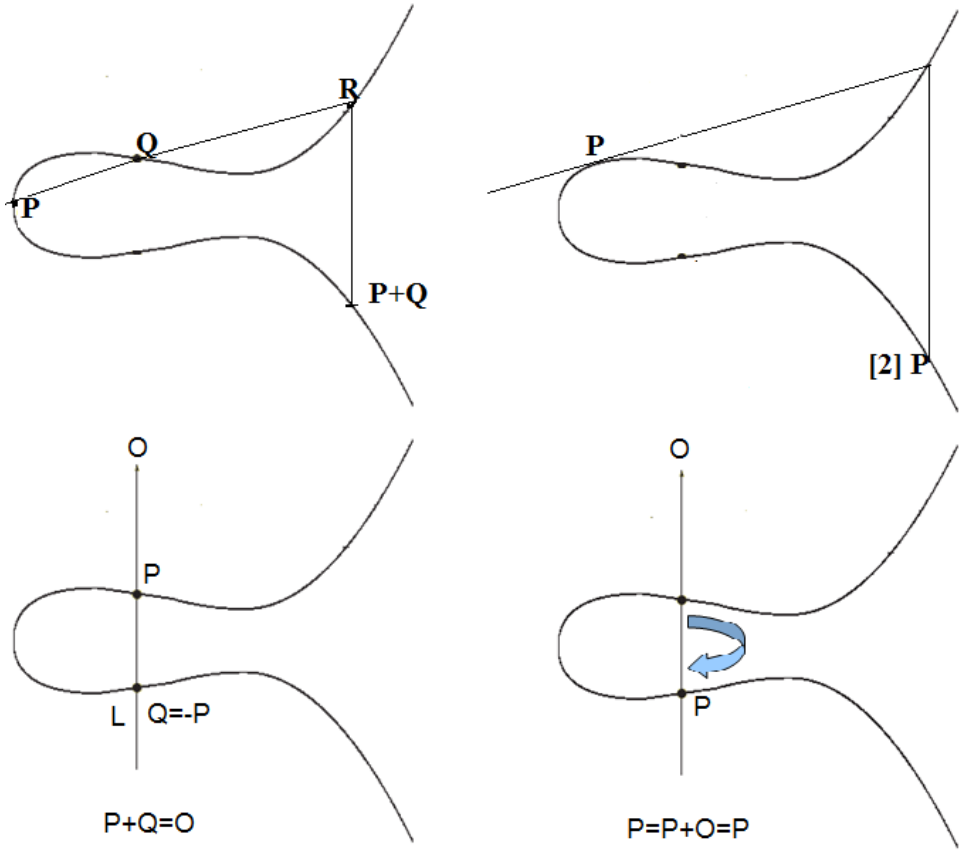
$$\begin{aligned} x_3 &= \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \\ y_3 &= -(\lambda + a_1)x_3 - \beta - a_3 \end{aligned} \tag{2}$$

where $\beta = y_1 - \lambda x_1$ and

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{if } P \neq Q, \\ \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}, & \text{if } P = Q. \end{cases}$$

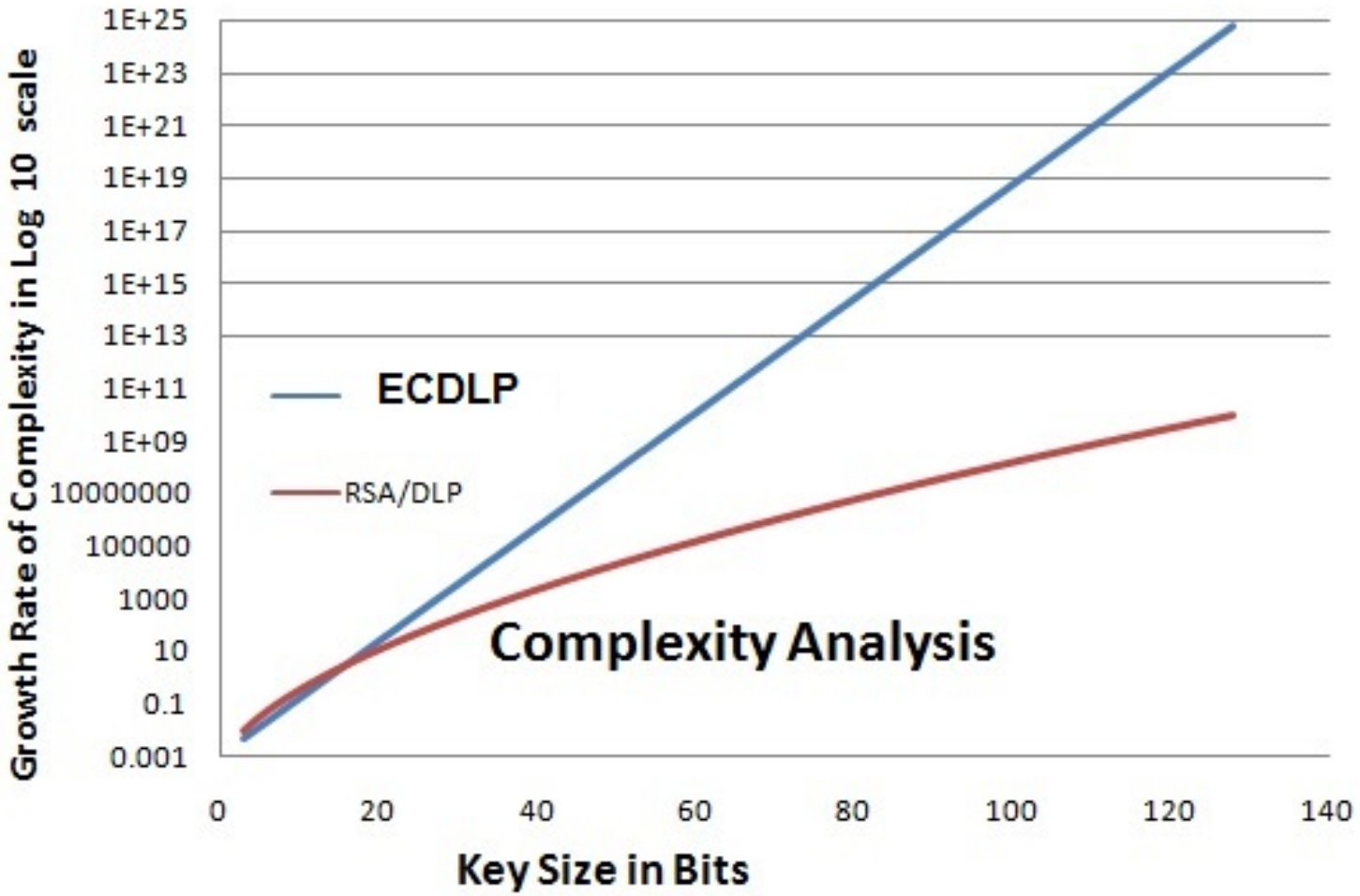
- $-P = (x_1, -y_1 - a_1x_1 - a_3)$

Indivar Gupta
SAG, DRDO, Delhi



Complexity Analysis

Indivar Gupta
SAG, DRDO, Delhi



Elliptic Curve based Cryptographic Schemes

Indivar Gupta
SAG, DRDO, Delhi

Koblitz (1987) and Miller (1985) first recommended the use of elliptic-curve groups (over finite fields) in cryptosystems. Some cryptographic schemes [6]:

1. Signature Scheme
   - ECDSA
   - EC-KCDSA
2. Public Key Encryption
   - ElGamal Elliptic Curve Encryption.
   - Elliptic Curve Integrated Encryption (ECIES)(Bellare and Rogaway, Variant of ElGamal).
   - Provably Secure Encryption Curve Scheme(PSEC)-Fujisaki and Okamoto.
3. Key Agreement
   - Diffie-Hellman.
   - Station-to-station (Diffie, Van Oorschot, Wiener).
   - ECMQV (Menezes, Qu, Vanstone).

ElGamal Elliptic Curve Cryptosystem

Indivar Gupta
SAG, DRDO, Delhi

ElGamal Cryptosystem Consists of:

- Setup: **Domain Parameters** $D = (F_p, E, P, n)$
- Key Generation Algorithm: Key Pair (d, Q), where $Q = d.P$.
- Encryption Algorithm: Ciphertext
- Decryption Algorithm: Plaintext

Encryption

Indivar Gupta
SAG, DRDO, Delhi

- **Input:** Elliptic curve domain parameters (F_p, E, P, n), public key $Q = d.P$, plaintext m.
- **Output:** Ciphertext (C_1, C_2)
  1. Represent the message m as a point M in $E(F_p)$.
  2. Select random $k \in [2, n-1]$.
  3. Compute $C_1 = k.P$(scalar multiplication).
  4. Compute $C_2 = M + k.Q$.
  5. Return (C_1, C_2).

Decryption

Indivar Gupta
SAG, DRDO, Delhi

- **Input:** Elliptic curve domain parameters (F_p, E, P, n), private key d, ciphertext (C_1, C_2).
- **Output:** Plaintext m
  1. Compute $M = C_2 - d.C_1$.
  2. Return (m).

Key Size Comparison:

Indivar Gupta
SAG, DRDO, Delhi

Key sizes (in bits) recommended by the National Institute of Standards and Technology (NIST) to protect keys used in (DES) and (AES) together with the key sizes for RSA, Diffie-Hellman and elliptic curves that are needed to provide equivalent security. [8]

Symmetric Key Size	**RSA and Diffie-Hellman Key Size**	**Elliptic Curve Key Size**
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

[8] https://www.nsa.gov/business/programs/elliptic_curve.shtm

ECC:Advantage

- Shorter Key Length translates to
  1. Faster implementation
  2. Less power consumption
  3. Less silicon area
- Can be implemented in 8/16/32-bits microprocessor with reasonable amount of security.
- Can be implemented in Smart Card, PDA etc..

Implementation of ECC

Indivar Gupta
SAG, DRDO, Delhi

Note: Basic necessity for implementation of ECC: a good finite filed library

- Choice of Finite Filed
- Choice of Basis
- Choice of coordinate Systems
- Choice of methods for scaler multiplication

Certicom ECC CChallanges[9]

Indivar Gupta
SAG, DRDO, Delhi

- The Challenge is to compute the ECC private keys from the given list of ECC public keys and associated system parameters
- Certicom Proposed two levels of challenges for elliptic curves over prime fields ($\mathbb{F}_p$) and elliptic over extension of finite fields ($\mathbb{F}_{2^n}$).

Level I	level II
109-bit challenge Solved EECp-109 Solved in 2002 EECC2-109 Solved in 2004	163-bit challenge
131-bit challenge	191-bit challenge
	239-bit challenge
	359-bit challenge

- World records: 113-bit Koblitz curve using FPGA-cluster

[9] Certicom Research

Identity Based Cryptography I

Indivar Gupta
SAG, DRDO, Delhi

- Conventional Public-key cryptography is dependent on a public-key infrastructure(PKI).
- PKI addresses authentication of public key problem
- A PKI often works with a party trusted by all users, called Certification Authority (CA)
- CA has to generate and sign certificates containing public key of users along with identity.
- PKI- based cryptography is both time-consuming and error-prone.
- In 1984, Shamir introduced the concept of identity-based cryptography.
- It uses user identity attributes, such as email addresses/ phone numbers company address instead of digital certificates

Identity Based Cryptography II

- More precisely, the public key of a user is derived directly from publicly available information.
- Practical realization of identity-based public key cryptography (ID-PKC) came prom pairings.
- In 2001, Boneh and Franklin proposed the first identity-based encryption scheme, using the bilinearity of pairings.
- Computationally Hard prob: Bilinear Diffie Hellman Problem
- Although it provides some advantages over PKI-based approaches, it is not without its drawbacks (Key escrow).

Outline

Indivar Gupta
SAG, DRDO, Delhi

Some Open Source Libraries[10]

- SAGE [System for Algebra and Geometry Experimentation]: http://www.sagemath.org/
- PARI/GP: http://pari.math.u-bordeaux.fr/
- crypto - OpenSSL cryptographic library: https://www.openssl.org/docs/crypto/crypto.html
- GAP http://www.gap-system.org
- NTL: A Library for doing Number Theory http://www.shoup.net/ntl/
- MIRACL http://indigo.ie/?mscott/
- GNU MP http://www.swox.com/gmp/

[10] Appendix B, [6]

References I

Neal Koblitz,

A Course in Number Theory and Cryptography (Graduate Texts in Mathematics, Springer, 2nd edition, 1994.

Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone

Handbook of Applied Cryptography, CRC Press, 1996.

John Talbot & Dominic Wels,

Complexity and Cryptography : An Introduction, Cambridge University Press, 2006.

Lawrence C. Washington,

Elliptic Curves: Number Theory and Cryptography, 2nd ed (Discrete Mathematics and Its Applications), CRC Press 2008.

Song Y. Yan & M.E. Hellmann,

Number Theory for Computing, 2nd ed, Springer, 2002.

References II

Darrel Hankerson, Alfred J. Menezes & Scott Vanstone,

Guide to Elliptic Curve Cryptography, Springer-Verlag, New York, 2004.

I F Blake, G. Seroussi & N P Smart,

Elliptic Curves in Cryptography, Cambridge University Press, 1999.

I F Blake, G. Seroussi & N P Smart,

Advances in Elliptic Curve Cryptography, Cambridge University Press 2nd Ed, 2005.

Abhijit Das

Computational Number Theory, , CRC Press, 2013.

Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen & Frederik Vercauteren

Handbook of Elliptic and Hyperelliptic Curve Cryptography, Chapman and Hall/CRC, 2005.

References III

Alasdair McAndrew,

Introduction to Cryptography with Open-Source Software, CRC Press, 2011.

Jintai Ding, Jason E. Gower & Dieter S Schmidt,

Multivariate Public Key Cryptography, Springer, 2006.

Daniel J Bernstein, Johannes Buchmann, Erik Dahmen,

Post-Quantum Cryptography, Springer, 2009

J L Mullen and D Panario,

Hand Book of Finite Fields, CRC Press, 2013.

Boneh, D. and Franklin, M.: Identity-Based Encryption from the Weil Pairing. Advances in Cryptology - Crypto 2001, LNCS 2139, Springer-Verlag (2001), pp. 213?229.

Thanks

indivargupta@sag.drdo.in