Mathematics of Public Key Cryptography

Eric Baxter

April 11, 2014

Overview

- Brief review of public-key cryptography
- Mathematics behind public-key cryptography algorithms

What is Public-Key Cryptography?

Cryptographic algorithm that uses *public* and *private* keys.

Public key:

- Public (everyone can see it)
- Used to encrypt plaintext or verify a digital signature

Private key:

- Private (only you can see it)
- Used to decrypt ciphertext or create digital signature

What makes the system useful/secure?

- Easy/quick to generate public/private key pair
- Hard/slow to extract private key from public key

Why Public-Key Cryptography?

Primary advantage of public-key cryptography:

- Doesn't require secure initial key exchange

Applications of public-key cryptography:

- Pretty Good Privacy (PGP) [commonly used computer program for encrypting and signing messages]
- GNU Privacy Guard [GPL licensed alternative to PGP]
- Transport Layer Security (TLS) [provides secure sessions when communicating over internet]

Why not Public-Key Cryptography?

Public-key cryptography also has several associated problems:

- Can be computationally expensive ($\rightarrow$ use hybrid cryptosystem)
- How can you be sure that owner of public key is who you think it is? ($\rightarrow$ use something like web of trust)
- Security based on math problems. Could new breakthrough break cryptosystem?

How Does Public-Key Cryptography Work?

All current public-key algorithms are based on mathematical problems that have no known efficient solution.

There are three problems that are widely used in public-key cryptography:

1. Integer factorization
2. Discrete logarithm
3. Elliptic curves

P-K Cryptography Based on Integer Factorization

Idea: it is easy to calculate products of integers, but hard to factor integers.

Toy version:

- For primes p, q, the public key is $n = pq$ and the private key is p

Definition

An integer p is prime if $p \geq 2$ and the only divisors of p are 1 and p.

Theorem

The Fundamental Theorem of Arithmetic says that every integer $n > 1$ is either prime or can be uniquely expressed as the product of primes.

P-K Cryptography Based on Integer Factorization

Actual implementation: RSA.

Public key is $n = pq$ and exponent, e. Private key is n and d such that $ed \equiv 1 (\mathrm{mod}\ (p-1)(q-1))$.
Encrypted message is $c(m) = m^e (\mod n)$ and to decrypt we use $m(c) = c^d (\mod n)$.

Definition

Modular arithmetic:
We say $a \equiv b (\mathrm{mod}\ c)$ if upon dividing a and b by c, the remainders are equivalent. Or, $a \equiv b (\mathrm{mod}\ c)$ if $\exists n \in \mathbb{Z}$ s.t. $(a - b) = nc$.

P-K Cryptography Based on Integer Factorization

RSA:
Public key is $n = pq$ and exponent, e. Private key is n and d such that $ed \equiv 1 (\bmod (p-1)(q-1))$. Encrypted message is $c(m) = m^e (\bmod n)$.

To break RSA, need to take e^{th} roots modulo composite n.

- Easiest known way to do this is to factor n
- Has been proven that getting d from n and e (i.e. getting private key from public key) is as hard as factoring n (assuming Extended Riemann Hypothesis).
- However, not known if breaking RSA is as hard as factoring. Maybe there's a way to take e^{th} root modulo n without factoring?

P-K Cryptography Based on Integer Factorization

Is factoring hard?

No known polynomial time algorithm exists to factor integers

Definition

Time complexity:
Quantifies amount of time taken by algorithm as function of length of input.
$O(1)$ means algorithm takes constant time regardless of input lenght.
$O(n)$ means algorithm time scales linearly with input.
etc.

Fastest known factoring algorithm is the general number field sieve. Runs in $\sim O(e^{1.9(\log N)^{1/3}(\log \log N)^{2/3}})$.

P-K Cryptography Based on Integer Factorization

Name	Complexity class	Running time ($T(n)$)	Examples of running times	Example algorithms
constant time		$O(1)$	10	Determining if an integer (represented in binary) is even or odd
inverse Ackermann time		$O(\alpha(n))$		Amortized time per operation using a disjoint set
iterated logarithmic time		$O(\log^* n)$		Distributed coloring of cycles
log-logarithmic		$O(\log \log n)$		Amortized time per operation using a bounded priority queue[2]
logarithmic time	DLOGTIME	$O(\log n)$	$\log n$, $\log(n^2)$	Binary search
polylogarithmic time		$\text{poly}(\log n)$	$(\log n)^2$	
fractional power		$O(n^c)$ where $0 < c < 1$	$n^{1/2}$, $n^{2/3}$	Searching in a kd-tree
linear time		$O(n)$	n	Finding the smallest item in an unsorted array
"n log star n" time		$O(n \log^* n)$		Seidel's polygon triangulation algorithm.
linearithmic time		$O(n \log n)$	$n \log n$, $\log n!$	Fastest possible comparison sort
quadratic time		$O(n^2)$	n^2	Bubble sort; Insertion sort; Direct convolution
cubic time		$O(n^3)$	n^3	Naive multiplication of two $n \times n$ matrices. Calculating partial correlation.
polynomial time	P	$2^{O(\log n)} = \text{poly}(n)$	n, $n \log n$, n^{10}	Karmarkar's algorithm for linear programming; AKS primality test
quasi-polynomial time	QP	$2^{\text{poly}(\log n)}$	$n^{\log \log n}$, $n^{\log n}$	Best-known $O(\log^2 n)$-approximation algorithm for the directed Steiner tree problem.
sub-exponential time (first definition)	SUBEXP	$O(2^{n^\varepsilon})$ for all $\varepsilon > 0$	$O(2^{\log n^{\log \log n}})$	Assuming complexity theoretic conjectures, BPP is contained in SUBEXP.[3]
sub-exponential time (second definition)		$2^{o(n)}$	$2^{n^{1/3}}$	Best-known algorithm for integer factorization and graph isomorphism
exponential time	E	$2^{O(n)}$	1.1^n, 10^n	Solving the traveling salesman problem using dynamic programming
factorial time		$O(n!)$	$n!$	Solving the traveling salesman problem via brute-force search
exponential time	EXPTIME	$2^{\text{poly}(n)}$	2^n, 2^{n^2}	
double exponential time	2-EXPTIME	$2^{2^{\text{poly}(n)}}$	2^{2^n}	Deciding the truth of a given statement in Presburger arithmetic

P-K Cryptography Based on Integer Factorization

Is factoring hard?

RSA challenge

- RSA labs published list of semiprimes (exactly two prime factors) with cash prizes for successful factorization
- Two weeks later, smallest number is factored (100 digits)
- In 2009, researchers factored 232-digit number (RSA-768) using hundreds of machines over a period of 2 years.

P-K Cryptography Based on Integer Factorization

RSA Number	Decimal digits	Binary digits	Cash prize offered	Factored on	Factored by
RSA-100	100	330	US$1,000[4]	April 1, 1991[5]	Arjen K. Lenstra
RSA-110	110	364	US$4,429[4]	April 14, 1992[5]	Arjen K. Lenstra and M.S. Manasse
RSA-120	120	397	$5,898[4]	July 9, 1993[6]	T. Denny et al.
RSA-129 [**]	129	426	$100 USD	April 26, 1994[5]	Arjen K. Lenstra et al.
RSA-130	130	430	US$14,527[4]	April 10, 1996	Arjen K. Lenstra et al.
RSA-140	140	463	US$17,226	February 2, 1999	Herman te Riele et al.
RSA-150 [*] ?	150	496		April 16, 2004	Kazumaro Aoki et al.
RSA-155	155	512	$9,383[4]	August 22, 1999	Herman te Riele et al.
RSA-160	160	530		April 1, 2003	Jens Franke et al., University of Bonn
RSA-170 [*]	170	563		December 29, 2009	D. Bonenberger and M. Krone [***]
RSA-576	174	576	$10,000 USD	December 3, 2003	Jens Franke et al., University of Bonn
RSA-180 [*]	180	596		May 8, 2010	S. A. Danilov and I. A. Popovyan, Moscow State University[7]
RSA-190 [*]	190	629		November 8, 2010	A. Timofeev and I. A. Popovyan
RSA-640	193	640	$20,000 USD	November 2, 2005	Jens Franke et al., University of Bonn
RSA-200 [*] ?	200	663		May 9, 2005	Jens Franke et al., University of Bonn
RSA-210 [*]	210	696		September 26, 2013[8]	Ryan Propper
RSA-704 [*]	212	704	$30,000 USD	July 2, 2012	Shi Bai, Emmanuel Thomé and Paul Zimmermann
RSA-220	220	729			
RSA-230	230	762			
RSA-232	232	768			
RSA-768 [*]	232	768	$50,000 USD	December 12, 2009	Thorsten Kleinjung et al.
RSA-240	240	795			
RSA-250	250	829			
RSA-260	260	862			

P-K Cryptography Based on Integer Factorization

Is factoring hard?

Shor's algorithm:
Quantum computer algorithm for factoring integers which runs in $O((\log N)^3)$.

Shor's algorithm has been demonstrated using early quantum computers!

Largest number factored using Shor's algorithm:

P-K Cryptography Based on Integer Factorization

Is factoring hard?

Shor's algorithm:
Quantum computer algorithm for factoring integers which runs in $O((\log N)^3)$.

Shor's algorithm has been demonstrated using early quantum computers!

Largest number factored using Shor's algorithm: 21

P-K Cryptography Based on Integer Factorization

How are primes used in RSA found?

Factoring is hardest when n is semi-prime. We want $n = pq$ where p and q are prime and:

- p and q should be of similar bit length but should not be very close
- p and q should be very large
- p and q should be chosen at random

Can find suitable p and q quickly using probabilistic primality tests. These algorithms run quickly and can determine whether a number is prime with high probability.

P-K Cryptography Based on Discrete Log Problem

Similar in many ways to integer factorization:

- No known polynomial time algorithms on non-quantum computers
- There are efficient algorithms on quantum computers
- Many algorithms can be adapted to both problems

The problem statement:
Find k such that $b^k = a$ for $a, b \in G$ where G is a group.

STUFF ABOUT GROUP THEORY

P-K Cryptography Based on Discrete Log Problem

How can we calculate discrete logs?

Simple algorithm is to raise group element, b, to higher and higher powers until we we find solution to $b^k = a$. The running time of this algorithm scales linearly with the group size, and thus exponentially in the number of digits in the size of the group.

P-K Cryptography Based on Discrete Log Problem

Which groups do we use?

P-K Cryptography Based on Elliptic Curves

Gives comparable security to RSA with significantly smaller key sizes and is less computationally demanding.

Very approximately:

An elliptic curve is a curve of the form $y^2 = x^3 ax + b$. Except here we're not interested in real a, b, x, y.

We define a multiplication operation for points on the curve.

With the above operation, points on the elliptic curve form a group.

Problem is related to computing discrete log on the elliptic curve group.

P-K Cryptography Based on Elliptic Curves

Applications:

- Tor
- Bitcoin
- iMessage

THE END