Cryptography: Then (up until 1970), Now, and In Future

Duncan Buell

Ancient Days

Until the 1970s, all cryptography was *symmetric*

That is, encrypting and decrypting use (essentially) the same key

This makes key management to maintain secrecy of the keys a very big deal

Caesar Cipher

Index the letters in alpha order

Shift each letter down by *n* spaces (e.g. 3)

A → D, B → E, C → F, and so forth

So decrypting is a shift of *-n*

Asymmetric/Public Key

Early 1970s, Cocks/Ellis/Williamson in the UK

1978, CACM, Rivest-Shamir-Adleman

Primes and Primitive Roots

Consider prime *p=11* and *primitive root 7*

We power up 7 modulo 11:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10 (exponents)

7, 5, 2, 3, 10, 4, 6, 9, 8, 1 (reduction mod 11)

For primes and powers of prim roots, we get essentially a random sequence of the *p-1* linear residues modulo *p*

The exponents are the *discrete logarithms*

Working Modulo $N = pq$

Consider $N = pq$ for primes p and q

Not quite a single cycle (as for primes) but almost, and of order $(p-1)(q-1)$

We will want to choose p and q carefully

(Get different big primes dividing $p-1$ and q-1, for example)

Montgomery Multiplication

Arithmetic on large (4096 bit) operands would be very slow, especially for modular reduction

But arithmetic on special operands can be quite fast

This is why the largest known primes are usually Mersenne numbers

Montgomery Multiplication (2)

Consider $M = 2^n - 1$, a Mersenne number

The product P of two integers $< M$ is $2n$ bits long and is $P=A*(2^n) + B$, with A and B each n bits

$$A*(2^n) + B = A*(2^{n-1}) + A + B$$

Reduce P mod M by adding left and right halves

Montgomery Multiplication (3)

Montgomery multiplication (Peter Montgomery) essentially involves converting all the arithmetic modulo $N=pq$ into arithmetic for which reduction modulo N is simple and fast

Essentially pre-multiply everything by an appropriate integer to make reduction modulo N look like reduction modulo Mersenne numbers

Rivest-Shamir-Adleman

Choose *p* and *q* of 2048 bits each, so *N=pq*

Knowing *p* and *q*, choose public exponent *E*, and compute the private exponent *D* for which

ED = 1 modulo (p-1)(q-1)

Publish *N* and *E*; hold *D* as private

To send me a message *M*, anyone can compute *C= M^E* modulo *N* and send me *C*

Without factoring *N*, computing *D* is hard, so only I can compute *C^D = M^(ED) = M modulo N*

Decrypting requires *D*, which requires factoring *N*

Diffie-Hellman Key Exchange

RSA isn’t used much because it is slow by comparison with elliptic curves

With two primes of 2048 bits, one has to do 4096 bit arithmetic, and that is “digit-sequential”

The NIST AES standard was developed to be feasible for credit cards, with minimal processing, and AES heavily uses byte-oriented computations

Cocks-Ellis-Williamson (or) Diffie-Hellman Key Exchange

Armadillo has public prime *P* and prim root *g* and a secret exponent *E*

Armadillo computes *A = g^E* and sends to Bobcat

Bobcat knows P, *g*, computes her own *E'*, computes *B = g^E'* and sends that to Armadillo, and computes *k = A^E' = g^(EE')*

Armadillo computes *B^E = g^(EE') = k*

And they both now have a common key to be used in a symmetric system (like AES)

The Discrete Log Problem

Given *e*, *N*, and *a* = *e^k modulo N*, find *k*

This is a computationally hard problem, although there is an "index calculus" method that can be used for some values of *N*

So we don't do things mod primes or products of primes, but using elliptic curves

Elliptic Curves

Consider the rational solutions to

$$y^2 = x^3 + Ax + B$$

where A and B can be chosen rational

The rational solutions form a mathematical group

A straight line cuts a cubic in three places, and the three (rational) points on the curve sum to zero

Bear with me … curve groups are written additively, not multiplicatively

Elliptic Curves (2)

Elliptic Curves (3)

Intersect $y = Mx + B$ with the curve, with two rational points $(x1,y1)$ and $(x2,y2)$ on the line

Then $M = (y2-y1)/(x2-x1)$ is rational

And $x^3 - M^2 x^2 + Sx + T = 0$, some S and T

Newton's equations: $x1 + x2 + x3 = M^2$

So the third $x3$ is rational, and thus y3 is rational

Elliptic Curves (4)

Elliptic curve groups have a finite number of generators of infinite order

The discrete log problem for curves does not have a solution – the multiples of a generator are also a reasonably random walk through the solutions

BUT (VERY BIG DEAL) one can work with much smaller arithmetic than with RSA or CEW/DH.

These days, routine is curves for key exchange and then AES for actual message passing

But Quantum Computing!!??

When we have quantum computers of reasonable size, all the stuff just discussed becomes trivial to break

We need other approaches for public key

All the recent NIST post-quantum algorithms are based on lattices

But Quantum Computing!!??

Shor's algorithm is a quantum algorithm for finding the prime factors of an integer. It was developed in 1994 by the American mathematician Peter Shor.[1][2] It is one of the few known quantum algorithms with compelling potential applications and strong evidence of superpolynomial speedup compared to best known classical (non-quantum) algorithms.[3] However, beating classical computers will require millions of qubits due to the overhead caused by quantum error correction.[4]

Shor proposed multiple similar algorithms for solving the factoring problem, the discrete logarithm problem, and the period-finding problem. "Shor's algorithm" usually refers to the factoring algorithm, but may refer to any of the three algorithms. The discrete logarithm algorithm and the factoring algorithm are instances of the period-finding algorithm, and all three are instances of the hidden subgroup problem.

Lattices

Given (0, 1) and (1, 0) in two dimensions, we have a *lattice* of all the points in the plane with integer coordinates

What if we go to several hundred dimensions, and non-integral basis elements?

We are looking at all the integer linear combinations of the basis elements

Hard Lattice Problems

Shortest Vector:

Given a lattice *L*, what's the shortest vector?

Closest vector:

Given a point in *n*-space, what's the closest lattice point?

These are computationally hard problems

http://duncanbuell.org

buell@acm.org

Chair Emeritus – NCR Chair in Computer Science and Engineering
Department of Computer Science and Engineering
University of South Carolina
Columbia, South Carolina 29208