Lecture 18: Computational Number Theory

Public Key Encryption Systems

Alice and Bob want to communicate private messages on a public communication channel.

1. Encryption: How can Bob encode a message into ciphertext that only Alice can decode?
2. Digital Signatures: How can Alice send a message so Bob knows it is from her?

If Alice and Bob announce the encryption scheme, couldn't any eavesdropper then decrypt the msg?

Public Key (\(P_A, P_B\))

\(P_A\) Alice \(P_B\) Bob

Secret Key (\(S_A, S_B\))

\(S_A\) \(S_B\)

\(P_A\) \(P_A (M)\) \(M\) \(P_A (C)\) \(S_A (C)\) are functional inverses

\(P_B\) \(S_B\) \(S_B (M)\) \(M\)

How to use:

Encryption: Bob: \(M \xrightarrow{P_B} C = P_A (M)\)

Alice: \(C \xrightarrow{S_A} M = S_A (C)\)

\(\text{transmit on insecure channel}\)

\(eavesdropper\)
Digital Signature

Alice signs $m'$:

$$m' \rightarrow \sigma = \sigma_a(m')$$

Bob verifies $m'$ comes from Alice

$$\text{transmit } (m', \sigma) \text{ on insecure channel}$$

Ciphertext (sign encrypted msg): Transmit $\text{transmit } P_B(m', \sigma)$$

Problem: Alice-Rabin key

Fundamental Difficulty: Create system of functions whereby many pairs constructions of $P_B$ and $\sigma_a$ can be easily formed such that disclosure of $P_B$ does not allow others to determine $\sigma_a$.

PKA: Public Key Cryptosystem $\rightarrow$ world standard

Ronald Rivest (MIT EECS; author of DES)

Adi Shamir

Leonard Adleman

2002 Turing Award

Based on relative ease of finding large prime numbers and extreme difficulty of factoring product of two primes.
RSA

1. Let $p 
eq q$ be 2 large prime integers (512 or 1024 bits)
2. Let $n = p \cdot q$ and $\phi(n) = (p-1)(q-1)$
3. Let $e$ be small, odd integer such that $\text{gcd}(e, \phi(n)) = 1 \Rightarrow e$ and $\phi(n)$ are "relatively prime"
4. Compute $d$ such that $d \cdot e \equiv 1 \pmod{\phi(n)}$ [multiplicative inverse]

Publish $P = (e, n)$ as Public Key
Keep $S = (d, n)$ as Private Key

Encryption: $C = P(M) = M^e \pmod{n}$

Decryption: $S(C) = C^d \pmod{n}$

$$= M^{ed} \pmod{n}$$

$$= M^{1 + k(p-1)(q-1)} \pmod{n}$$

$$= M \pmod{n}$$

One method of cracking code is to find $d$. This requires factoring $n$ into $p$ and $q$ to construct $\phi(n) = (p-1)(q-1)$ to compute $d$ in step 4. Factoring product of two large primes turns out to be very difficult.
The density of prime numbers is rather large.

Prime distribution function \( \pi(n) \) is the number of primes less than or equal to \( n \).

Prime number theorem: \( \lim_{n \to \infty} \frac{\pi(n)}{\frac{n}{\ln n}} = 1 \).

Probability that \( n \) is prime is \( \frac{1}{\ln n} \).

They need to examine \( \sigma \) len \( n \) integers near \( n \) to find a prime.

For 512-bit prime number, examine \( \sigma \) len \( 2^{512} \times 335 \) random numbers (half that value if restrict to odds).

Primality Testing:

1. Naive: Trial Division
   - Divide by each integer \( 2, 3, \ldots \) up to \( \sqrt{n} \).
   - Can skip evens beyond \( 2 \).
   - Exponential in length of \( n \) because \( \beta \) bits \( \Rightarrow \beta = \log (n+1) \Rightarrow 2^{\beta - 1} \approx \sqrt{n} = \Theta(2^{\beta/2}) \).
(3) Pseudo-primality Testing (almost works)

Definition: \( n \) is a base-\( a \) pseudoprime if \( n \) is composite and \( a^{n-1} \equiv 1 \pmod{n} \)

Note: By Fermat's theorem if \( p \) is prime then \( a^{p-1} \equiv 1 \pmod{p} \)

For all \( \text{coprime} \) with \( n \), where \( a \neq 0 \bmod{n} \)

\( \begin{array}{c}
7: & a \equiv 2 \pmod{7} \quad a^7 \equiv 2^7 \equiv 64 \\ 6: & a \equiv 2 \pmod{6} \quad a^6 \equiv 2^6 \equiv 64 \\ 5: & a \equiv 2 \pmod{5} \quad a^5 \equiv 2^5 \equiv 32 \\ 4: & a \equiv 2 \pmod{4} \quad a^4 \equiv 2^4 \equiv 16 \\ 3: & a \equiv 2 \pmod{3} \quad a^3 \equiv 2^3 \equiv 8 \\
\end{array} \)

If \( n \) does not satisfy above for same \( a \), then \( n \) is composite.

While not rigorously true, it turns out that if \( n \) does satisfy above for same \( a \), \( n \) is usually prime.

Errors are rare. Let \( a = 2 \)

For only 22 values of \( n < 10,000 \) does this fail:

512 - but \( n \) will be prime in all but \( 1 \) out of \( 10^2 \) trials.
1024 - but \( n \) is \( \ldots \) in all but \( 10 \) out of \( 10^4 \) trials.

You might try to improve by checking more values of \( a \). However, there is a small class of \( a \) (cyclotomic numbers) for which this strategy will fail for all values of \( a \).

\[ \checkmark \text{better soln.} \]
Two fixes:
- multiple, random values of \( a \)
- detects possibility of Carmichael numbers

Let \( \text{WITNESS}(a, n) \) be a routine that reports whether the integer \( a \) can demonstrate \( n \) is composite.

Let \( n - 1 = 2^t \cdot u \), where \( t \geq 1 \) and \( u \) is odd

\( x_0 \leftarrow a^u \mod n \)

\( x_i \leftarrow x_{i-1}^2 \mod n \)

for \( i \leftarrow 1 \) to \( t \)

if \( x_i = 1 \) and \( x_{2i} \neq 1 \) and \( x_{u+i} \neq n-1 \) (Carmichael)

then return \( \text{TRUE} \)

if \( x_t \neq 1 \)

then return \( \text{TRUE} \) → detects base-a pseudoprime failure

return \( \text{FALSE} \) → \( n \) is not provably composite & may be prime

\[ \text{MILLER-RABIN}(n, s) \]

for \( j \leftarrow 1 \) to \( s \)

do \( a \leftarrow \text{RANDOM}(1, n-1) \)

if \( \text{WITNESS}(a, n) \)

then return composite → guaranteed

return \( \text{PRIME} \) → highly probable
Running Time: Let \( n \) be represented as \( \beta \) bits
\[\text{H-R uses } O(\beta^2) \text{ arithmetic operations} \]
\[O(\beta^3) \text{ bit operations} \]
- much less than trial division

**Theorem:** If \( n \) is an odd composite number, then the
# of witnesses to the compositeness of \( n \) is at least \( \frac{n-1}{2} \)

**Theorem:** For any odd integer \( n \geq 2 \) and positive integer \( s \),
the probability that \( \text{H-R}(n,s) \) errs is at most \( 2^{-s} \).

Proof:

Effectively, some large \( s \) should suffice.

In fact, when applied to randomly chosen large integers, a
much smaller value \( (s+3) \) should suffice.