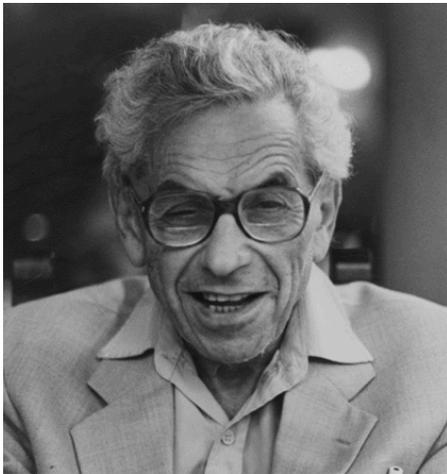


SM280: Cryptography
Assoc. Prof. Will Traves

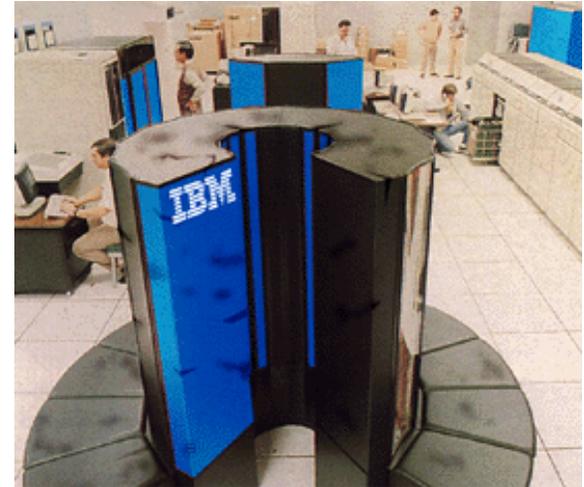
19 SEP 2006

Primality Testing

- CIA paid \$\$ for large primes
- Prime Number Theorem (Erdős):
 $\#\text{primes} \leq x \sim \ln(x)$
- Factoring to test for primality
 - Serious drawbacks: 10 billions years to factor a 60 digit number using basic division



Paul
Erdős



ASCII
White

Fermat's Little Theorem

- FLT: if p is prime then $a^{p-1} = 1 \pmod p$ for all $a \neq 0$.

EX: ($p = 7, a = 2$) $2^6 = 64 = 1 \pmod 7$.

EX: ($p = 12, a = 2$) $2^{11} = 2048 = 8 \pmod{12}$,

so 12 is not prime.

- ~~Combinatorial numbers~~



Pierre de Fermat postage, France, 2001



Kayal, Saxena and Agrawal

Extended FLT

- Thm: if $n = pq$ and $\Phi(n) = (p-1)(q-1)$, then $a^{1+k\Phi(n)} = a \pmod{n}$.



Euler on a 10 mark note
(still legal currency in Germany)

Public Key Encryption

- Alice wants to send a message to Bob without the eavesdropper Eve being able to decipher it.



RSA Encryption

- Rivest, Shamir, Adelman invented one of the first public key encryption algorithms
- Bob: $n = pq$, $\Phi(n) = (p-1)(q-1)$,
e relatively prime to $\Phi(n)$
- Bob: publishes n , e .
- Alice: send $m^e \bmod n$ rather than m
- Bob: $d = 1/e \bmod \Phi(n)$ so $ed = 1 + k\Phi(n)$.
 $(m^e)^d = m^{ed} = m^{1+k\Phi(n)} = m \bmod n$

Eve's difficulty

- To decrypt, Eve needs d and we can only get this by finding $\Phi(n) = (p-1)(q-1)$.
- Finding $\Phi(n)$ is equivalent to factoring n
- Factoring $n=pq$ is expected to be hard
- RSA.com factoring challenges:
 - RSA-129 (1977; 1994) Lenstra
 - RSA-576 * 174 digits * \$10,000
 - RSA-2048 * 617 digits * \$200,000



A. Lenstra

Digital Signatures

- Bob wants Alice to sign a document but Alice wants security from counterfeiting.
- Bob: sends m to Alice
- Alice: $n=pq$, e , $d=1/e \pmod n$ as before publishes n , e
- Alice sends (m, m^d) to Bob
- Bob: checks it's Alice's signature
 $(m^d)^e = m \pmod n.$

Counterfeiting

- Alice sends (m, m^d) to Bob
- Bob: checks it's Alice's signature
 $(m^d)^e = m \pmod n$.
- If Bob claims Alice signed x then she can deny because $(m^d)^e \neq x \pmod n$.
- To find the proper signature for x Bob needs to find y such that $y^e = x \pmod n$
(equivalent to decrypting RSA message x)

Efficiency and Hash Functions

- The signature is as long as the document
- To shorten the signature, Alice just signs a hash of the document
- Hash function: message \rightarrow message digest
 - Fast computation
 - Given m.d. y , hard to find m such that $h(y)=m$
 - Hard to find m, m' with $h(m) = h(m')$
- NIST – Secure Hash Algorithm
- Bob – needs to find x st $h(x)^e = h(m)^e$ but x is most likely meaningless and hard to find

Birthday Paradox

- Birthday paradox: if 23 people in a room then 50% chance that two share a BD
$$P = 1 - (1 - 1/365)(1 - 2/365) \dots (1 - 22/365)$$
- if n people in two rooms and r objects then chance of match between rooms is about $1 - e^{-\lambda}$ where $r = (\lambda n)^{1/2}$.
- This leads to an attack on signature schemes

Birthday Attack

- Alice signs 50 bit hash of message digest
- Bob unlikely to find bad doc with same hash ($1/2^{50}$)
- Bob finds 30 places to make a change in m and makes 2^{30} good and bad docs
- Since $n=2^{30}$ and $r=2^{50}$, $\lambda=2^{10}$ and $P \sim 1$ for good and bad messages with same hash
- Bob gets Alice to sign good and replaces with bad.

Foiled!

- Alice can foil Bob's scheme by making a small change in the document before she signs
- Then Bob is back to finding a bad document with same hash as the signed good document ($1/2^{50}$ chance).

