RSA and Fermat.

RSA:

Key Generation: (Alice)
Primes: p, q. N = pq.
Encryption Key: e, where \( \text{gcd}(e, (p−1)(q−1)) = 1 \)
Decryption Key: \( d = e^{-1} \mod (p−1)(q−1) \)

Message: \( m \)
Encryption (Bob): \( y = E(m) = m^e \mod N \).
Decryption (Alice): \( D(y) = y^d \mod N \).
Result: \( m^d \mod N \)
Want \( D(E(x)) = x \)
Thm: \( x^d \mod N \)

RSA and Fermat: mathematical connection

Thm: \( m^d = m \mod (pq) \) if \( ed = 1 \mod (p−1)(q−1) \)

Seems like magic!

Fermat’s Little Theorem: For prime p, and \( a \neq 0 \mod p \),
\( a^{p-1} = 1 \mod p \).
\( 3^6 \mod 7 \)? 1.
\( 3^7 \mod 7 \)? 3.
\( 3^{3^6-1} \mod 7 \)? \( (3^{3^6} + 3) \mod 7 \)? 3.

Corollary: \( a^{k(p−1)+1} = a \mod p \)
Get a back when exponent is 1 \( mod (p−1) \).
A little like RSA:
\( a^d \mod (p−1)(q−1) = a \)
when exponent is 1 \( mod (p−1)(q−1) \).

Proof of Corollary. If \( a = 0 \), \( a^{k(p−1)+1} = 0 \mod m \).
Otherwise \( a^{k(p−1)+1} = a^{(p−1)+1} = a \mod p \).

Idea: Fermat removes the \( k(p−1) \) from the exponent!

Correctness of RSA...

Lemma 1: For any prime \( p \) and any \( a, b \),
\( a^{x+k(p−1)(q−1)} = a \mod p \)

Lemma 2: For any two different primes \( p, q \) and any \( x, k \),
\( x^{1+k(p−1)(q−1)} = x \mod pq \)

RSA and Fermat: mathematical connection

Lemma 2: For any two different primes \( p, q \) and any \( x, k \),
\( x^{1+k(p−1)(q−1)} = x \mod pq \)

Theorem: RSA correctly decodes!

Recall
\( D(E(x)) = (x^e)^d = x^{ed} = x \mod pq \),
where \( ed = 1 \mod (p−1)(q−1) \)
\( ed = 1+k(p−1)(q−1) \)
\( x^{ed} = x^{k(p−1)(q−1)+1} = x \mod pq \).

Key Generation...

1. Find large (100 digit) primes \( p \) and \( q \)?
Prime Number Theorem: \( \pi(N) \) denotes the number of primes less than or equal to \( N \). For all \( N \geq 17 \)
\( \pi(N) \approx N/\ln N \).

Choosing randomly gives approximately 1/(ln N) chance of number being a prime. (How do you tell if it is prime? ...)

2. Choose \( e \) with gcd(\( e, (p−1)(q−1) \)) = 1.
Use gcd algorithm to test.

3. Find inverse \( d \) of \( e \) modulo \( (p−1)(q−1) \).
Use extended gcd algorithm.

All steps are polynomial in \( O(\log N) \), the number of bits.

Security of RSA.

Security?
1. Alice knows \( p \) and \( q \) (and \( d \), and other numbers).
2. Bob only knows, \( N \mod q \), and \( e \).
   Does not know, for example, \( d \) or factorization of \( N \).
3. Breaking this scheme \( \implies \) factoring \( N \).
   Don’t know how to factor \( N \) efficiently on regular computers.
Much more to it in practice!

If Bobs sends a message (Credit Card Number) to Alice, Eve sees it. (The encrypted CC number.) Eve can send same credit card number again!! “Replay attack”

The protocols are built on RSA but more complicated; For example, several rounds of challenge/response.

One trick:
Bob encodes credit card number, \( c \), concatenated with random \( k \)-bit number \( r \).

Never sends just \( c \).

Again, more work to do to get entire system.
CS161...