Modulo Multiplication as a Public Key Cipher

- a and b are inverses mod n iff:
  - \( a \times b \mod n = 1 \)
- Only numbers that are relatively prime to \( n \) have multiplicative inverses mod n
- Inverses mod n can be found using Euclid's algorithm.
- Can we use multiplicative inverses modulo n as public and private keys?

Modulo Multiplication as a Public Key Cipher

- \( e, d \) are multiplicative inverses mod n iff:
  - \( ed \mod n = 1 \)
- If \( e \) and \( d \) are inverses then:
  - \( m \times e \times d \mod n \equiv m \mod n \)
  - For example:
    - Select \( e = 5, d = 6, n = 29 \), so \( ed = 1 \mod 29 \)
    - Encrypt 17: \( 17 \times 5 \mod 29 = 85 \mod 29 = 27 \)
    - Decrypt 27: \( 27 \times 6 \mod 29 = 162 \mod 29 = 17 \)

Multiplication mod n for Public/Private Keys Scramble

- If we transmit in 5 bit blocks with \( n = 32 \), then use
  - 3 as the public key, 11 as the private key
  - Arbitrarily select plaintext as 10110₂ = 22
- Encryption
  - Given \( X = 22, k = 3, n = 32 \)
  - \( E(X, k) = X \times k \mod n \)
  - \( 3 \times 22 \mod 32 = 2 \mod 32 = 2 = 00010₂ \)
- Decryption
  - \( k^{-1} = \text{inverse (3 mod 32)} = 11 \)
  - \( D(X', k^{-1}) = X'^{k^{-1}} \mod n, \text{where } X' = 2 \& k^{-1} = 11 \)
  - \( 11 \times 2 \mod 32 = 22 \mod 32 \)

Multiplication mod n for Public/Private Keys Scramble

- Unfortunately, multiplicative inverses are not secure.
- So, can we securely use multiplicative inverses modulo n as secure encryption?
- Kind of. We actually use the properties of multiplicative inverses mod n to use exponentative inverses mod n
"ed" As An Exponentive Inverse

1. By Euler's Theorem:
   - \( a^{\phi(n)} \equiv 1 \mod n \), then
   - \( a^{k\phi(n)+1} \equiv a \mod n \)

2. Select \( ed \equiv 1 \mod \Phi(n) \), then

3. \( ed = k\Phi(n) + 1 \) for some \( k \), so

4. \( m^{ed} = m^{k\Phi(n)+1} \)

5. By Euler's theorem,
   - \( m^{k\Phi(n)+1} \mod n \equiv m \mod n \)

6. Thus, for any \( m \),
   - \( m^{ed} \mod n = m \mod n \)

Exponentiation mod \( n \) As Public/Private Key Scramble

- Find \( e \) and \( d \) such that \( med = m \mod n \)
  - That is, find \( e \)'s exponentive inverse
- \( D[E[m,e],d] \equiv (m^e \mod n)^d \mod n \equiv m \)
- Encryption: \( E(m,e) \equiv m^e \mod n \)
- Decryption: \( D(c,d) \equiv c^d \mod n \)
  - \( (m^e \mod n)^d \mod n \)
  - \( m^{ed} \mod n \)
  - \( m \mod n \)

The Foundation of RSA

- \( x^y \mod n = x \mod \phi(n) \mod n \)
- If \( y \mod \Phi(n) = 1, \) then for any \( x \), \( x^y \mod n = x \mod n \)
- If we can choose \( e \) and \( d \) such that
  - \( ed = 1 \mod \phi(n) \)
  - then we can encrypt by raising \( x \) to the \( e \)th power and decrypt by raising to the \( d \)th power.

The RSA Algorithm

1. Select two large primes, \( p,q \). Multiply them to get \( n \).
2. As your public key, select \( e \) relatively prime to \( \Phi(n) \)
3. As your private key, *find \( d \) that is the multiplicative inverse of \( e \mod \Phi(n) \).
4. Encrypt \( m < n \) as, \( c = m^e \mod n \).
5. Decrypt \( c \) as, \( m = c^d \mod n \).

Why does finding \( d \) as the multiplicative inverse of \( e \mod \Phi(n) \), make \( d \) the exponentive inverse of \( e \mod n \)?

* Multiplicative inverse can be found using Euclid's algorithm.

RSA

- Rivest, Shamir, Adleman, 1978, MIT
- Variable key size, common to use 1024
- Generating RSA keys is based on finding multiplicative inverses of large numbers (modulo), which is not hard
- Generating RSA ciphertext is based on modulo exponentiation, which is not hard
- RSA's strength is based on difficulty of factoring large numbers and computing discrete logarithms, WHICH ARE HARD
- There may be other trap doors in RSA, but none have been found yet.

RSA Example

1. Select two large primes, \( 2357, 2551 \). Multiply them to get \( n = 6012707 \)
2. As your public key, select \( e = 3674911 \), relatively prime to \( \Phi(n) = 600780 \)
3. As your private key, *find \( d = 422191 \) that is the multiplicative inverse of \( e \mod \Phi(n) \).
4. Encrypt \( m = 5234673 < n \) as, \( c = m^e \mod n = 5234673^{3674911} \mod 6012707 = 3650502 \)
5. Decrypt \( c \) as, \( m = c^d \mod n = 3650502^{422191} \mod 6012707 = 5234673 \)
6. Decrypt \( c \) as, \( m = c^d \mod n = 3650502^{422191} \mod 6012707 = 5234673 \)
### An Attack on RSA

1. An intruder, X intercepts a message (m) intended for Alice encrypted under Alice's public key (e): 
   \[(m)^e \mod n\]
2. The intruder generates a value x, computes: 
   \[x^e \mod n = (x^m \mod n)\] and sends it to Alice.
3. Alice decrypts the message to attain the value: 
   \[x \mod n\], which appears to be garbage.
4. If Alice disposes of the "garbage" carelessly, the intruder can recover it and compute: 
   \[(x^m x^{-1} \mod n) = m\]

---

### RSA Summary

1. Keys are asymmetric
2. THE PREDOMINANT PUBLIC KEY CIPHER
3. It's strength is based on the difficulty of factoring large numbers and difficulty of computing discrete logarithms.
4. It is much more computationally intensive than DES, IDEA, AES, etc.
5. It has avoidable weaknesses.

---

### El Gamel

- Based on computation of discrete logs, i.e. finding an integer solution to: \[b^x = c \mod m\] is a discrete logarithm base b of c mod m.
- For random m, b, & c, there may be no such x.
- For prime modulo, and some b, there exists a discrete log for any c not divisible by p.
- Too hard for me.
- Again, involves message exponentiation.

---

### Elliptic Curve Cryptography

- Strength also comes from the difficulty of computing discrete logarithms.
- Elliptic curves over finite fields determine groups.
- \[Z_n^*\] is a group (multiplication is: 
  - Closed, associative, has an identity element, each element has an inverse).
- ECC's have a good resistance to attack versus key size.

---

### Zero Knowledge

- A proof that divulges nothing other than the confirmation of the original assertion to the verifier.
- A proof such that the verifier can prove no more after the proof than they could if they were given the new theorem by an oracle.

---

### An Example

- Meteorologist: I can make it rain!
- Skeptic: How?
- Meteorologist: Not telling.
- Skeptic: OK, make it rain:
  - Here
  - At 8:00 am and 8:00 pm
  - For the next five days.
Another Example
- Man has a secret passage between two caves. Prove it without giving up any information
  1. Verifier turns their back
  2. Prover enters the cave, (verifier does not see which side is entered)
  3. Verifier identifies which side the prover should come out
  4. Prover comes out the proper side

Two Questions
1. Is the proof actually a proof?
2. Is the proof zero knowledge?

Both are VERY hard questions!

Uses for Zero Knowledge
- Authentication

ZKPs Are Usually:
- Interactive
- Probabilistic

Interactive Proofs
- Q&A between prover/verifier
- Verifier asks questions until accept or reject
- Bounded number of questions

Probabilistic Proofs
- Key Parameters
  - Size of the sample space
  - Time the judge is given to make the decision
Review

- Modular Multiplication as Cryptography
- Modular Exponentiation as Cryptography
- RSA
- Others
- Zero Knowledge