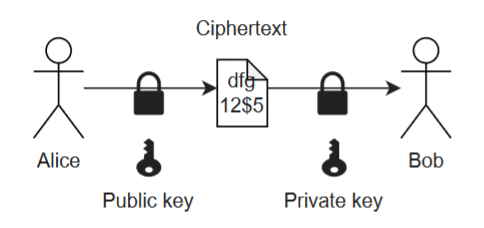
Asymmetric Cryptography

Functionality

Every user has two keys

* Public key (encryption key)
* Private key (decryption key)

(Dis)advantages of asymmetric cryptography

Advantages:

* Does not require sharing of private key
* Everybody that has the public key can send a message
  + Less keys needed
* Only the owner of the private key can decrypt
* Can sign messages
  + Non-repudiation

Disadvantages:

* Slow, not practical for long messages
* Distribution of public key
* Long keys needed

Modular arithmetic

Remainder after division:

* 12%7 = 5
* -2%7 = 5

If two numbers are the same given a modulus, we can write that as:

* + Pronounces as 5 is congruent to 12 mod 7

The numbers modulo form a set: ; For every we have

Mathematical group

* Set of elements (typically numbers) that are related to each other
* All operations in this group are done modulo
* We write this group as , where is the modulo
* The set must have the following properties, given an operation
  + Closure: For any two x and y in the group, is also in the group
  + Associativity: For any x, y, z in the group,
  + Identity existence: There is an element e such that
  + Inverse existence: For any x in the group, there is a y, such that
* Examples:
  + Group , Elements in group:
  + Group , Elements in group:
  + Difference between these groups
    - 5 is prime, 4 is not
    - Elements that share a factor with the modulo are not in the group

Co-prime

* where p is prime has n – 1 elements
  + This is because there are no elements that share a divisor (other than 1) with p
* In groups where n is not prime, the group consists of all elements that do not share a factor with n (other than 1)
* When two numbers share no factors other than 1, they are called co-prime or relative prime
* To test if two numbers are co-prime, we can use greatest common divisor (GCD)
* Number of elements in can be calculated by Euler’s totient function:
  + We will only be focusing on our problem of
  + Then
* Example: Calculate gcd(78, 129
  + Gcd(78, 12) = gcd(12, 78 mod 12)
  + = gcd(12, 6)
  + = gcd(6, 12 mod 6)
  + = gcd(6, 0)
  + = 6
  + So 12 and 78 are NOT coprime, because the gcd is not 1

Calculating multiplicative inverse

* Inverse existence: for any x in the group, there is a y such that
* Take
* e = 1
* Multiplicative inverse of 3: 2
* Can use the extended GCD to calculate
* returns a pair with

Example: find a, b such that

* First: calculate the
  + 1914 = 2 \* 899 + 116
  + 899 = 7 \* 116 + 87
  + 116 = 1 \* 87 + 29
  + 87 = 3 \* 29 + 0
  + 🡪
* Second: backfill starting at the last second to last equation
  + 116 = 1 \* 87 + 29 🡪 29 = 116 – 1 \* 87
  + 29 = 116 – 1 \* (899 – 7 \* 116)
  + 29 = 116 – 1 \* 899 + 7 \* 116
  + 29 = 8 \* 116 – 1 \* 899
  + 29 = 8 \* (1914 – 2 \* 899) – 1 \* 899
  + 29 = 8 \* 1914 – 16 \* 899 – 1 \* 899
  + 29 = 8 \* 1914 – 17 \* 899

RSA

Key generation – Steps

1. Choose two large prime numbers p and q
2. Calculate the modulo
   1. n is the key length
3. Calculate
4. Choose an integer e, such that and the , thus e is coprime to
   1. Common numbers are e = 3 and
5. Determine d as
   1. d is the multiplication inverse of e
6. Keys are:
   1. Public key: (e, n)
   2. Private key: (d, n)

Encryption / Decryption

* Message: m; Ciphertext: c
* Encryption:
* Decryption:
* Works because:

Example of RSA Key generation

1. Generate two primes: p = 5 and q = 11
2. Calculate
3. Calculate
4. Pick an e:
   1. E.g., e = 17
   2. Verify:
      1. 40 = 2 \* 17 + 6
      2. 17 = 2 \* 6 + 5
      3. 6 = 1 \* 5 + 1
      4. 5 = 5 \* 1 + 0 ✓
5. Compute d as the multiplicative inverse of e
   1. We have that
   2. We can use the extended gcd to find:
   3. Pick
      1. Then we have
      2. Backfill the gcd from step 4
         1. 1 = 6 – 1 \* 5
         2. 1 = 6 – 1 \* (17 – 2 \* 6)
         3. 1 = 6 – 1 \* 17 + 2 \* 6
         4. 1 = 3 \* 6 – 1 \* 17
         5. 1 = 3 \* (40 – 2 \* 17) – 1 \* 17
         6. 1 = 3 \* 40 – 7 \* 17
         7. Because d is minus, do:
6. Keys:
   1. Public key: (e, n) = (40, 55)
   2. Private key: (d, n) = (33, 55)
7. Encrypt message: 31
8. Decrypt cipher: 26

Weaknesses of RSA

Do not use schoolbook RSA:

* RSA is deterministic, will always encrypt the same message with the same ciphertext
  + Like AES with ECB
* Given two ciphertexts and , we can derive the ciphertext of
  + Can deduce information about the original message
  + Like with the OTP, when a key is reused
* Solution: use a proper padding scheme that hides relations
  + Optical Asymmetric Encryption Padding (OAEP)

Signature in RSA

* RSA can be used to sign a message
* RSA has the property that it works both ways
  + Encrypt with public key and decrypt with private key
  + Encrypt with private key and decrypt with public key
* When a message is encrypted with the private key (signed), we know
  + That only the owner of the private key can do this
    - Non-repudiation
  + Everyone can decrypt the message
    - Verify the contents of what is “signed”

Signature in RSA – Weakness

* Some trivial messages can be forged
  + and
* Imagine Bob does not want to sign the message m = 15, which would give
  + We take a random number r = 24 and its inverse = 39
  + Calculate the encryption:
  + Compute
  + Let Bob sign it:
  + We can now multiply the ciphertext with to get
* Fix for this: Hashing the message
  + Use a padding scheme: Probabilistic Padding Scheme (PPS)

Security protocols

General

* Communication in the form of “A 🡪 B : m” which means Alice sends message m to Bob
* A sequence of such message is intended to achieve a security goal
  + Confidentiality, Integrity, Authenticity, Non-repudiation
* After every step of the protocol the beliefs of the participants change
* When something goes wrong the protocol is aborted

Asymmetric protocols

* Everyone has to keys
  + for encryption, public key
  + for decryption, private key
* Encrypt:
* Decrypt:
* and for RSA:
* Assumptions:
  + Everyone has two keys
  + Encryption is one-way (need for private key)
  + Private key cannot be reconstructed

Protocol for confidentiality

* Symmetric version:
* Asymmetric version:
* What is an assumption: everybody knows everybody’s public key
* No integrity guaranteed

Protocol for integrity

* Symmetric version:
* Asymmetric version:
* Does this work?
  + No, everybody knows the public key of Bob
  + Integrity will be combined with non-repudiation

Non-repudiation

* We can use digital signatures for non-repudiation:
* What does Bob need to check?
  + That m and H(m) are the same
* What does this guarantee?
  + Integrity
  + Non-repudiation
  + Authenticity
* Also include confidentiality:

Asymmetric is slow

* Asymmetric algorithms are slow
* Asymmetric algorithms only work on small messages (less than the modulo in case of RSA)
* Symmetric encryption is fast and works on arbitrary sizes
* How can we fix slowness of asymmetric?
  + Use a session key: