Public key encryption

Computer Security DD2395

Roberto Guanciale
robertog@kth.se

2015-11-29

Hash functions

Question 4: weak collision resistant means that there is no collision. For every x there is no y such that $f(x) = f(y)$

A Yes

B No

Hash functions

Question 5: weak collision resistant means that there are few collisions. For every x there are few y such that $f(x) = f(y)$

A Yes

B No

Hash functions

For every x it is computationally difficult to find y such that $f(x) = f(y)$

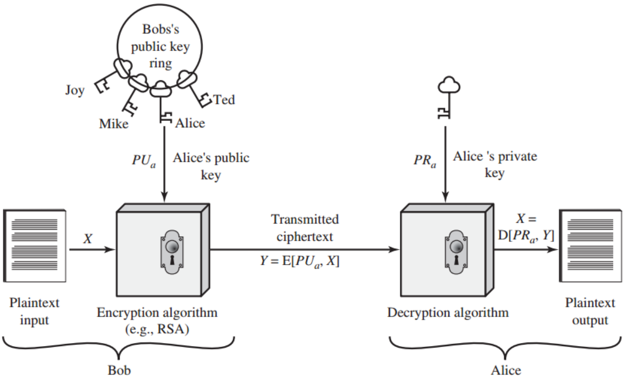
Let F be the function that truncates a message and keeps the first megabyte. Assuming that usually we have messages of 100 MB.

1. How many messages have the same hash if we use F?
2. How many messages have the same hash if we use SHA_{256}?
3. There are more collisions in F or SHA_{256}?
4. Which function should I use as hash function? why?

Public key encryption

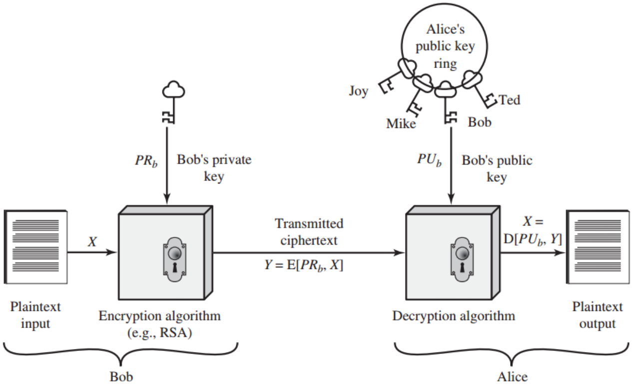
- Sharing symmetric keys is difficult
- No public key encryption mechanism existed before 1970s

Public key encryption



Confidentiality

Public key encryption



Integrity/authentication

Requirements

- It is easy for B to generate the key pair (PU_b, PR_b)

Requirements

- It is easy for B to generate the key pair (PU_b, PR_b)
- It is easy for A, knowing PU_b and M, to compute $C = E(PU_b, M)$
- It is easy for B, knowing PR_b and C, to compute $M = D(PR_b, C)$

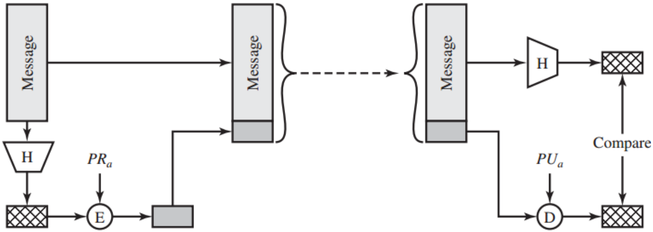
Requirements

- It is easy for B to generate the key pair (PU_b, PR_b)
- It is easy for A, knowing PU_b and M, to compute $C = E(PU_b, M)$
- It is easy for B, knowing PR_b and C, to compute $M = D(PR_b, C)$
- It is not feasible to compute PR_b knowing PU_b
- It is not feasible to infer M knowing PU_b and C

Requirements

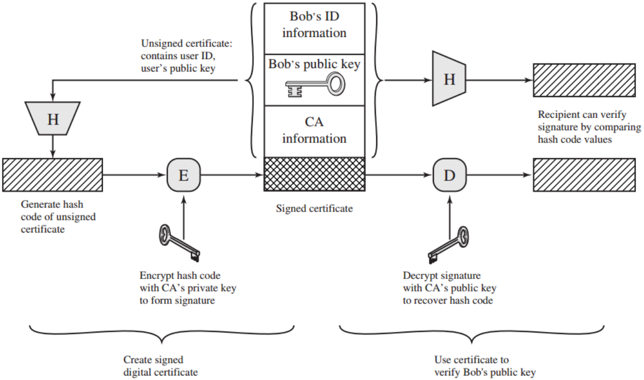
- It is easy for B to generate the key pair (PU_b, PR_b)
- It is easy for A, knowing PU_b and M, to compute $C = E(PU_b, M)$
- It is easy for B, knowing PR_b and C, to compute $M = D(PR_b, C)$
- It is not feasible to compute PR_b knowing PU_b
- It is not feasible to infer M knowing PU_b and C
- (optional) $M = D(PU_b, E(PR_b, M)) = D(PR_b, E(PU_b, M))$

Applications



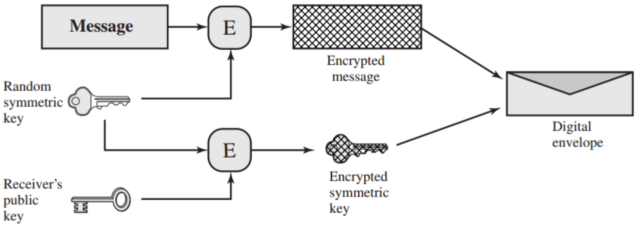
Digital Signature (i.e. authentication of sender/integrity of payload)

Applications



Digital Certificate (i.e. authentication/integrity of public keys)

Applications



Exchange of symmetric keys/Digital envelopes

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p-1)(q-1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p-1)(q-1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$
- It is hard to find d such that $M = (M^e)^d \ mod \ n$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p - 1)(q - 1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$
- It is hard to find d such that $M = (M^e)^d \ mod \ n$
- Discrete logarithmic problem

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p - 1)(q - 1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$
- It is hard to find d such that $M = (M^e)^d \ mod \ n$
- Discrete logarithmic problem
- n is usually 2048 bits

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p - 1)(q - 1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p-1)(q-1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

- $33 = 3 * 11$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p-1)(q-1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

- $33 = 3 * 11$
- $20 = 2 * 10$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p - 1)(q - 1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

- $33 = 3 * 11$
- $20 = 2 * 10$
- 7

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p - 1)(q - 1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

- $33 = 3 * 11$
- $20 = 2 * 10$
- 7
- 3

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p-1)(q-1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

- $33 = 3 * 11$
- $20 = 2 * 10$
- 7
- 3
- $(33, 7)$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p-1)(q-1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

- $33 = 3 * 11$
- $20 = 2 * 10$
- 7
- 3
- $(33, 7)$
- $(33, 3)$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p-1)(q-1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

- $33 = 3 * 11$
- $20 = 2 * 10$
- 7
- 3
- $(33, 7)$
- $(33, 3)$
- $29 = 2^7 \ \% \ 33$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- $n = p * q$, where p, q primes
- $\phi(n) = (p-1)(q-1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

- $33 = 3 * 11$
- $20 = 2 * 10$
- 7
- 3
- $(33, 7)$
- $(33, 3)$
- $29 = 2^7 \ \% \ 33$
- $2 = 29^3 \ \% \ 33$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- Encryption: $C = M^e \ mod \ n$
- Decryption: $M = C^d \ mod \ n$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- Encryption: $C = M^e \ mod \ n$
- Decryption: $M = C^d \ mod \ n$
- $C^d \ mod \ n = (M^e \ mod \ n)^d \ mod \ n = (M^e)^d \ mod \ n = M^{ed} \ mod \ n$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- Encryption: $C = M^e \; mod \; n$
- Decryption: $M = C^d \; mod \; n$
- $C^d \; mod \; n = (M^e \; mod \; n)^d \; mod \; n = (M^e)^d \; mod \; n = M^{ed} \; mod \; n$

1. e, d, n such that $M = M^{ed} \; mod \; n$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- Encryption: $C = M^e \ mod \ n$
- Decryption: $M = C^d \ mod \ n$
- $C^d \ mod \ n = (M^e \ mod \ n)^d \ mod \ n = (M^e)^d \ mod \ n = M^{ed} \ mod \ n$

1. e, d, n such that $M = M^{ed} \ mod \ n$
2. It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

Ron Rivest, Adi Shamir, and Len Adleman: RSA

- Encryption: $C = M^e \ mod \ n$
- Decryption: $M = C^d \ mod \ n$
- $C^d \ mod \ n = (M^e \ mod \ n)^d \ mod \ n = (M^e)^d \ mod \ n = M^{ed} \ mod \ n$

1. e, d, n such that $M = M^{ed} \ mod \ n$
2. It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$
3. It is not feasible to discover d knowing e and n

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  $e = 0$: $c_0 = 1 \ mod \ n = 1$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  $e = 0$: $c_0 = 1 \ mod \ n = 1$
  $e = 1$: $c_1 = M \ mod \ n$

Requirement 2: It is easy to compute M^e mod n and C^d mod n

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  $e = 0$: $c_0 = 1 \ mod \ n = 1$
  $e = 1$: $c_1 = M \ mod \ n$
  $e = 2$: $c_2 = c_1 * M \ mod \ n$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  $e = 0$: $c_0 = 1 \ mod \ n = 1$
  $e = 1$: $c_1 = M \ mod \ n$
  $e = 2$: $c_2 = c_1 * M \ mod \ n$
  $\ldots O(e)$ multiplications

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  $e = 0$: $c_0 = 1 \ mod \ n = 1$
  $e = 1$: $c_1 = M \ mod \ n$
  $e = 2$: $c_2 = c_1 * M \ mod \ n$
  $\ldots O(e)$ multiplications

  $7^{257} \ mod \ 13$

Requirement 2: It is easy to compute $M^e \; mod \; n$ and $C^d \; mod \; n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \; mod \; n = (a \; mod \; n) * (b \; mod \; n) \; mod \; n$
  $e = 0$: $c_0 = 1 \; mod \; n = 1$
  $e = 1$: $c_1 = M \; mod \; n$
  $e = 2$: $c_2 = c_1 * M \; mod \; n$
  $\ldots O(e)$ multiplications

  $7^{257} \; mod \; 13$
  $7^1 \; mod \; 13 = 7 \; mod \; 13 = 7$

Requirement 2: It is easy to compute $M^e \; mod \; n$ and $C^d \; mod \; n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \; mod \; n = (a \; mod \; n) * (b \; mod \; n) \; mod \; n$
  $e = 0$: $c_0 = 1 \; mod \; n = 1$
  $e = 1$: $c_1 = M \; mod \; n$
  $e = 2$: $c_2 = c_1 * M \; mod \; n$
  $\ldots O(e)$ multiplications

  $7^{257} \; mod \; 13$
  $7^1 \; mod \; 13 = 7 \; mod \; 13 = 7$
  $7^2 \; mod \; 13 = 7 * 7 \; mod \; 13 = 49 \; mod \; 13 = 10$

Requirement 2: It is easy to compute M^e *mod* n and C^d *mod* n

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  $e = 0$: $c_0 = 1 \ mod \ n = 1$
  $e = 1$: $c_1 = M \ mod \ n$
  $e = 2$: $c_2 = c_1 * M \ mod \ n$
  $\ldots O(e)$ multiplications

  $7^{257} \ mod \ 13$
  $7^1 \ mod \ 13 = 7 \ mod \ 13 = 7$
  $7^2 \ mod \ 13 = 7 * 7 \ mod \ 13 = 49 \ mod \ 13 = 10$
  $7^3 \ mod \ 13 = 10 * 7 \ mod \ 13 = 70 \ mod \ 13 = 5$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  $e = 0$: $c_0 = 1 \ mod \ n = 1$
  $e = 1$: $c_1 = M \ mod \ n$
  $e = 2$: $c_2 = c_1 * M \ mod \ n$
  $\ldots O(e)$ multiplications

  $7^{257} \ mod \ 13$
  $7^1 \ mod \ 13 = 7 \ mod \ 13 = 7$
  $7^2 \ mod \ 13 = 7 * 7 \ mod \ 13 = 49 \ mod \ 13 = 10$
  $7^3 \ mod \ 13 = 10 * 7 \ mod \ 13 = 70 \ mod \ 13 = 5$
  $7^4 \ mod \ 13 = 5 * 7 \ mod \ 13 = 35 \ mod \ 13 = 9$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- Naive approach: compute M^e
  $O(e)$ multiplications, memory intensive
- Memory efficient:
  $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  $e = 0$: $c_0 = 1 \ mod \ n = 1$
  $e = 1$: $c_1 = M \ mod \ n$
  $e = 2$: $c_2 = c_1 * M \ mod \ n$
  $\ldots O(e)$ multiplications

  $7^{257} \ mod \ 13$
  $7^1 \ mod \ 13 = 7 \ mod \ 13 = 7$
  $7^2 \ mod \ 13 = 7 * 7 \ mod \ 13 = 49 \ mod \ 13 = 10$
  $7^3 \ mod \ 13 = 10 * 7 \ mod \ 13 = 70 \ mod \ 13 = 5$
  $7^4 \ mod \ 13 = 5 * 7 \ mod \ 13 = 35 \ mod \ 13 = 9$
  $\ldots$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

Requirement 2: It is easy to compute M^e mod n and C^d mod n

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\ldots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \ldots (7^{2^0})^1 = (7^{2^8}) * 1 * \ldots (7^{2^0})$

  $C := 1$
  for $i \in [0 \ldots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $C = 1, \ M = 7$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\ldots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \ldots (7^{2^0})^1 = (7^{2^8}) * 1 * \ldots (7^{2^0})$

  $C := 1$
  for $i \in [0 \ldots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $a_0 = 1$ then $C := 1 * 7 \ mod \ 13 = 7$ and
  $M := 7 * 7 \ mod \ 13 = 10$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $C = 7, \ M = 10$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
  if $(a_i = 1)$ then $C := C * M \ mod \ n$
  $M := M * M \ mod \ n$

  $a_1 = 0$ then $C = 7$ and $M := 10 * 10 \ mod \ 13 = 9$

Requirement 2: It is easy to compute $M^e \; mod \; n$ and $C^d \; mod \; n$

- $a * b \; mod \; n = (a \; mod \; n) * (b \; mod \; n) \; mod \; n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \; mod \; n$
      $M := M * M \; mod \; n$

  $C = 7, \; M = 9$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $a_2 = 0$ then $C = 7$ and $M := 9 * 9 \ mod \ 13 = 3$

Requirement 2: It is easy to compute M^e mod n and C^d mod n

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $C = 7, \ M = 3$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
  if $(a_i = 1)$ then $C := C * M \ mod \ n$
  $M := M * M \ mod \ n$

  $a_3 = 0$ then $C = 7$ and $M := 3 * 3 \ mod \ 13 = 9$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $C = 7, \ M = 9$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $a_4 = 0$ then $C = 7$ and $M := 9 * 9 \ mod \ 13 = 3$

Requirement 2: It is easy to compute M^e mod n and C^d mod n

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $C = 7, \ M = 3$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $a_5 = 0$ then $C = 7$ and $M := 3 * 3 \ mod \ 13 = 9$

Requirement 2: It is easy to compute $M^e \; mod \; n$ and $C^d \; mod \; n$

- $a * b \; mod \; n = (a \; mod \; n) * (b \; mod \; n) \; mod \; n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
  if $(a_i = 1)$ then $C := C * M \; mod \; n$
  $M := M * M \; mod \; n$

  $C = 7, \; M = 9$

Requirement 2: It is easy to compute $M^e \; mod \; n$ and $C^d \; mod \; n$

- $a * b \; mod \; n = (a \; mod \; n) * (b \; mod \; n) \; mod \; n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\ldots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \ldots (7^{2^0})^1 = (7^{2^8}) * 1 * \ldots (7^{2^0})$

  $C := 1$
  for $i \in [0 \ldots bits]$
  $\quad$ if $(a_i = 1)$ then $C := C * M \; mod \; n$
  $\quad$ $M := M * M \; mod \; n$

  $a_6 = 0$ then $C = 7$ and $M := 9 * 9 \; mod \; 13 = 3$

Requirement 2: It is easy to compute M^e *mod* n and C^d *mod* n

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$

  Let $e = \sum_i a_i * 2^i$

  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$

  thus $M^e = \prod_i (M^{2^i})^{a_i}$

  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $C = 7, \ M = 3$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\ldots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \ldots (7^{2^0})^1 = (7^{2^8}) * 1 * \ldots (7^{2^0})$

  $C := 1$
  for $i \in [0 \ldots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $a_7 = 0$ then $C = 7$ and $M := 3 * 3 \ mod \ 13 = 9$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\ldots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \ldots (7^{2^0})^1 = (7^{2^8}) * 1 * \ldots (7^{2^0})$

  $C := 1$
  for $i \in [0 \ldots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $C = 7, \ M = 9$

Requirement 2: It is easy to compute $M^e \ mod \ n$ and $C^d \ mod \ n$

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$
  Let $e = \sum_i a_i * 2^i$
  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$
  thus $M^e = \prod_i (M^{2^i})^{a_i}$
  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$
  for $i \in [0 \dots bits]$
      if $(a_i = 1)$ then $C := C * M \ mod \ n$
      $M := M * M \ mod \ n$

  $a_8 = 1$ then $C = 7 * 9 \ mod \ 13 = 11$ and
  $M := 9 * 9 \ mod \ 13 = 3$

Requirement 2: It is easy to compute M^e mod n and C^d mod n

- $a * b \ mod \ n = (a \ mod \ n) * (b \ mod \ n) \ mod \ n$

  Let $e = \sum_i a_i * 2^i$

  e.g. $257 = (1 * 2^8) + (0 * 2^7) + (0 * 2^6) + (\dots) + (1 * 2^0)$

  thus $M^e = \prod_i (M^{2^i})^{a_i}$

  e.g. $7^{257} = (7^{2^8})^1 * (7^{2^7})^0 * \dots (7^{2^0})^1 = (7^{2^8}) * 1 * \dots (7^{2^0})$

  $C := 1$

  for $i \in [0 \dots bits]$

      if $(a_i = 1)$ then $C := C * M \ mod \ n$

      $M := M * M \ mod \ n$

  $C = 11, \ M = 3$

Exercise

Work in pairs (10 mins), generate the keys, encrypt and decrypt a message.

- $n = p * q$, where p, q primes
- $\phi(n) = (p-1)(q-1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- Public key $PU = [n, e]$
- Private Key $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $M = C^d \ mod \ n$

- $33 = 3 * 11$
- $20 = 2 * 10$
- 7
- 3
- $(30, 7)$
- $(30, 3)$
- $29 = 2^7 \ \% \ 30$
- $2 = 29^3 \ \% \ 30$

Requirement 1: $M = M^{ed} \ mod \ n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)

Requirement 1: $M = M^{ed}\ mod\ n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)
  - $e * d\ mod\ \phi(n) = 1$

Requirement 1: $M = M^{ed} \ mod \ n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)
  - $e * d \ mod \ \phi(n) = 1$
  - $e * d = 1 + h * \phi(n)$ for some h

Requirement 1: $M = M^{ed} \ mod \ n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)
    - $e * d \ mod \ \phi(n) = 1$
    - $e * d = 1 + h * \phi(n)$ for some h
- assuming that M and n are coprime:

Requirement 1: $M = M^{ed} \ mod \ n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)
  - $e * d \ mod \ \phi(n) = 1$
  - $e * d = 1 + h * \phi(n)$ for some h
- assuming that M and n are coprime:
  - Euler theorem: $M^{\phi(n)} \equiv_{mod \ n} 1$

Requirement 1: $M = M^{ed} \ mod \ n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)
  - $e * d \ mod \ \phi(n) = 1$
  - $e * d = 1 + h * \phi(n)$ for some h
- assuming that M and n are coprime:
  - Euler theorem: $M^{\phi(n)} \equiv_{mod \ n} 1$
  - M^{ed}

Requirement 1: $M = M^{ed} \ mod \ n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)
  - $e * d \ mod \ \phi(n) = 1$
  - $e * d = 1 + h * \phi(n)$ for some h
- assuming that M and n are coprime:
  - Euler theorem: $M^{\phi(n)} \equiv_{mod \ n} 1$
  - $M^{ed} = M^{1+h*\phi(n)}$

Requirement 1: $M = M^{ed} \; mod \; n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)
  - $e * d \; mod \; \phi(n) = 1$
  - $e * d = 1 + h * \phi(n)$ for some h
- assuming that M and n are coprime:
  - Euler theorem: $M^{\phi(n)} \equiv_{mod \; n} 1$
  - $M^{ed} = M^{1+h*\phi(n)} = M * (M^{\phi(n)})^h$

Requirement 1: $M = M^{ed} \ mod \ n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)
  - $e * d \ mod \ \phi(n) = 1$
  - $e * d = 1 + h * \phi(n)$ for some h
- assuming that M and n are coprime:
  - Euler theorem: $M^{\phi(n)} \equiv_{mod \ n} 1$
  - $M^{ed} = M^{1+h*\phi(n)} = M * (M^{\phi(n)})^h \equiv_{mod \ n} M * 1^h$

Requirement 1: $M = M^{ed} \ mod \ n$

- Let e and d be multiplicative inverse modulo $\phi(n)$ (Euler totient function: numbers coprime with n)
  - $e * d \ mod \ \phi(n) = 1$
  - $e * d = 1 + h * \phi(n)$ for some h
- assuming that M and n are coprime:
  - Euler theorem: $M^{\phi(n)} \equiv_{mod \ n} 1$
  - $M^{ed} = M^{1+h*\phi(n)} = M * (M^{\phi(n)})^h \equiv_{mod \ n} M * 1^h \equiv_{mod \ n} M$

Requirement 1: $M = M^{ed} \ mod \ n$

- $e * d \ mod \ \phi(n) = 1$
- $n = p * q$, where p, q primes

Requirement 1: $M = M^{ed} \ mod \ n$

- $e * d \ mod \ \phi(n) = 1$
- $n = p * q$, where p, q primes
- Euler total function is multiplicative: a, b coprime then $\phi(a * b) = \phi(a) * \phi(b)$

Requirement 1: $M = M^{ed} \ mod \ n$

- $e * d \ mod \ \phi(n) = 1$
- $n = p * q$, where p, q primes
- Euler total function is multiplicative: a, b coprime then $\phi(a * b) = \phi(a) * \phi(b)$
- If a prime then $\phi(a) = (a - 1)$

Requirement 1: $M = M^{ed} \ mod \ n$

- $e * d \ mod \ \phi(n) = 1$
- $n = p * q$, where p, q primes
- Euler total function is multiplicative: a, b coprime then $\phi(a * b) = \phi(a) * \phi(b)$
- If a prime then $\phi(a) = (a - 1)$
- $\phi(n) = (p - 1)(q - 1)$

Requirement 1: $M = M^{ed} \ mod \ n$

- $n = p * q$, where p, q primes
- $e * d \ mod \ \phi(n) = 1$
- $\phi(n) = (p - 1)(q - 1)$

Requirement 1: $M = M^{ed} \ mod \ n$

- $n = p * q$, where p, q primes
- $e * d \ mod \ \phi(n) = 1$
- $\phi(n) = (p - 1)(q - 1)$
- The multiplicative inverse of e modulo $\phi(n)$ exists if and only if e and $\phi(n)$ are coprime

Requirement 1: $M = M^{ed}\ mod\ n$

- $n = p * q$, where p, q primes
- $e * d\ mod\ \phi(n) = 1$
- $\phi(n) = (p-1)(q-1)$
- The multiplicative inverse of e modulo $\phi(n)$ exists if and only if e and $\phi(n)$ are coprime
- Select e as random coprime of $\phi(n)$

Requirement 1: $M = M^{ed} \ mod \ n$

- $n = p * q$, where p, q primes
- $e * d \ mod \ \phi(n) = 1$
- $\phi(n) = (p - 1)(q - 1)$
- The multiplicative inverse of e modulo $\phi(n)$ exists if and only if e and $\phi(n)$ are coprime
- Select e as random coprime of $\phi(n)$
- Compute d

RSA

- $n = p * q$, where p, q primes
- $e * d \ mod \ \phi(n) = 1$
- $\phi(n) = (p - 1)(q - 1)$
- Select e as random coprime of $\phi(n)$
- Compute d such that $e * d \ mod \ \phi(n) = 1$
- $PU = [n, e]$
- $PR = [n, d]$
- Encryption $C = M^e \ mod \ n$
- Decryption $D = C^d \ mod \ n$

RSA: security

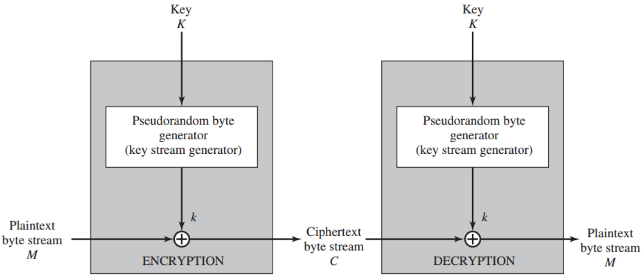
- Brute force
- Side channels: e.g. time to decrypt
- Mathematical attacks: discrete logarithmic
- Mathematical attacks: factorize the product of two prime numbers
- Quantum computing

Number of Decimal Digits	Date Achieved
100	April 1991
110	April 1992
120	June 1993
130	April 1996
140	February 1999
155	August 1999
190	November 2005
232	December 2009
270	75 000 USD

Other public-key encryption mechanism

- Dffie-Hellman: generation of a shared symmetric key
- Elliptic curve: like RSA (cheaper, different mathematical assumption)

Stream cyphers



- encryption sequence should have a large period
- key stream should resemble a random stream

Thank you

Questions?