Quick reviews / corrections

Question Review 4

- Important lesson: Authentication does not imply encryption

- Second lesson: pay attention to the definition

Some people would have said that the left is an authentication scheme and the right one isn't

$\mathsf{Gen}' := \mathsf{Gen}$

$\mathsf{Mac}'(k, m) :=$

$t' \leftarrow \mathsf{Mac}(k, m)$

$t \leftarrow (t', m)$

return t

$\mathsf{Verify}'(k, m, t) :=$

$(t', m') \leftarrow t$

$if\ (m \neq m')$ return reject

return $\mathsf{Verify}(k, m, t')$.

$\mathsf{Gen}' := \mathsf{Gen}$

$\mathsf{Mac}'(k, m) :=$

$t' \leftarrow \mathsf{Mac}(k, m)$

$t \leftarrow (t', m)$

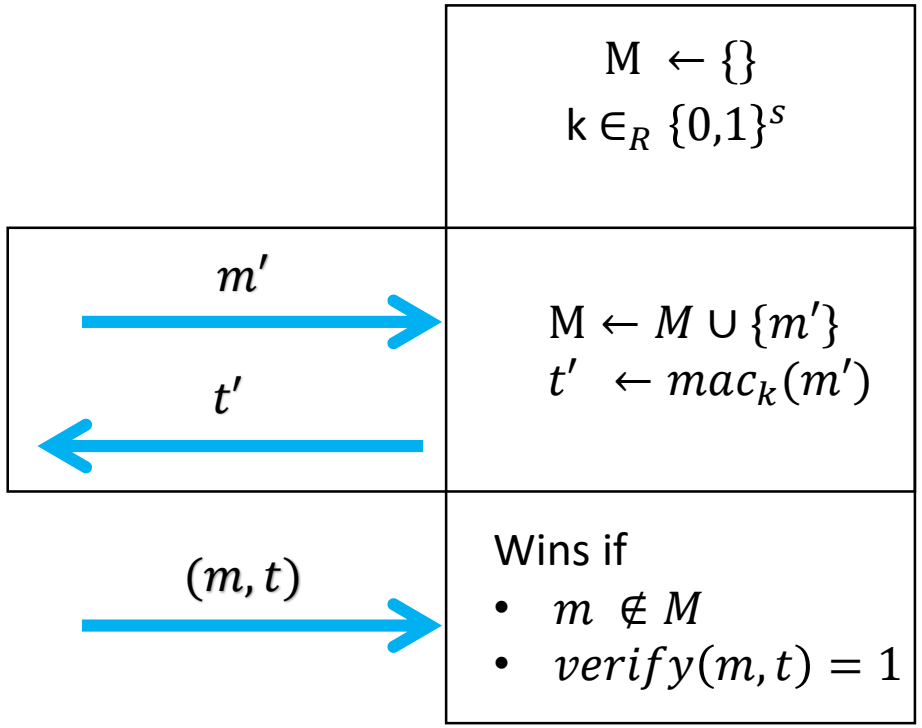
return t

$\mathsf{Verify}'(k, m, t) :=$

$(t', m') \leftarrow t$

return $\mathsf{Verify}(k, m, t')$.

Mac forgery game



Repeat as many times
as the adversary wants

Some people would have said that the left is an authentication scheme and the right one isn't

$\mathsf{Gen}' := \mathsf{Gen}$

$\mathsf{Mac}'(k, m) :=$

$t' \leftarrow \mathsf{Mac}(k, m)$

$t \leftarrow (t', m)$

return t

$\mathsf{Verify}'(k, m, t) :=$

$(t', m') \leftarrow t$

$if\ (m \neq m')$ return reject

return $\mathsf{Verify}(k, m, t')$.

$\mathsf{Gen}' := \mathsf{Gen}$

$\mathsf{Mac}'(k, m) :=$

$t' \leftarrow \mathsf{Mac}(k, m)$

$t \leftarrow (t', m)$

return t

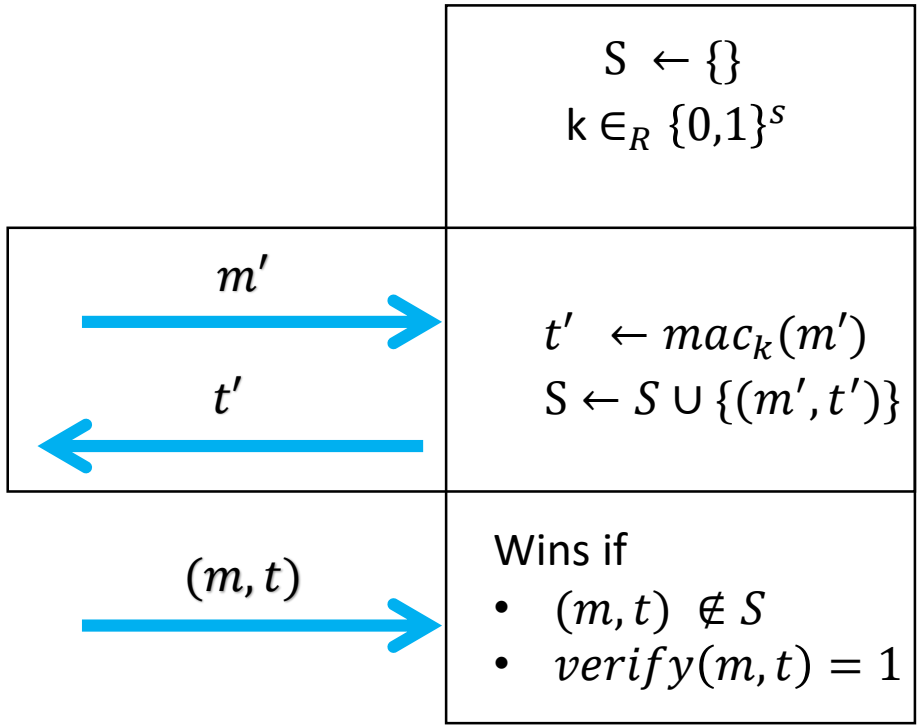
$\mathsf{Verify}'(k, m, t) :=$

$(t', m') \leftarrow t$

return $\mathsf{Verify}(k, m, t')$.

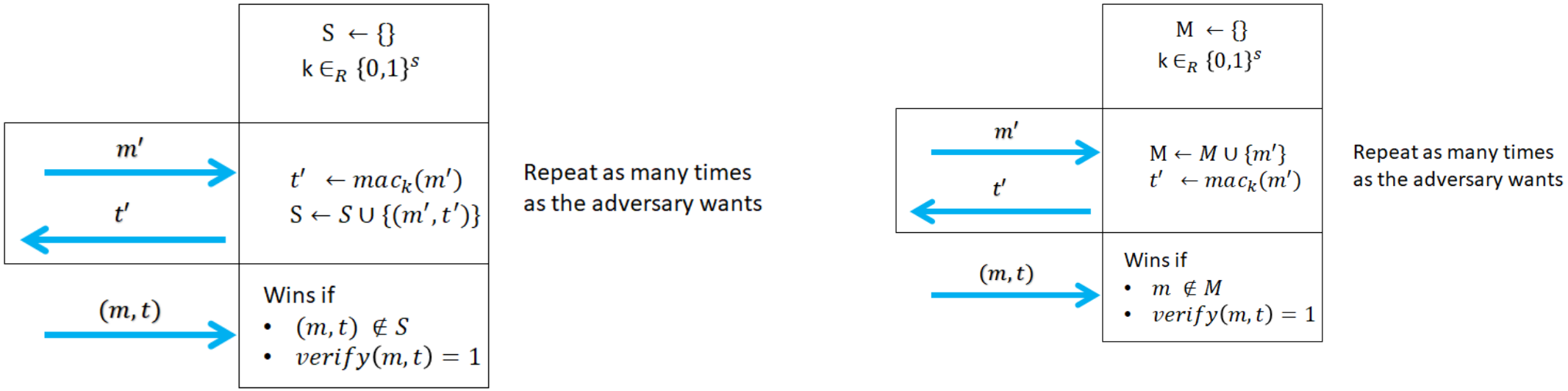
We can take a mac that is accepted on the right side and easily turn it into a mac on the left

Strong version of the mac forgery game



Repeat as many times
as the adversary wants

Mac game comparison



If a scheme (gen,mac,verify) is a strong mac scheme then it is also a mac scheme, the other way does not hold

According to the definition of strong authentication

$\mathsf{Gen}' := \mathsf{Gen}$

$\mathsf{Mac}'(k, m) :=$

$t' \leftarrow \mathsf{Mac}(k, m)$

$t \leftarrow (t', m)$

return t

$\mathsf{Verify}'(k, m, t) :=$

$(t', m') \leftarrow t$

$if\ (m \neq m')$ return reject

return $\mathsf{Verify}(k, m, t')$.

$\mathsf{Gen}' := \mathsf{Gen}$

$\mathsf{Mac}'(k, m) :=$

$t' \leftarrow \mathsf{Mac}(k, m)$

$t \leftarrow (t', m)$

return t

$\mathsf{Verify}'(k, m, t) :=$

$(t', m') \leftarrow t$

return $\mathsf{Verify}(k, m, t')$.

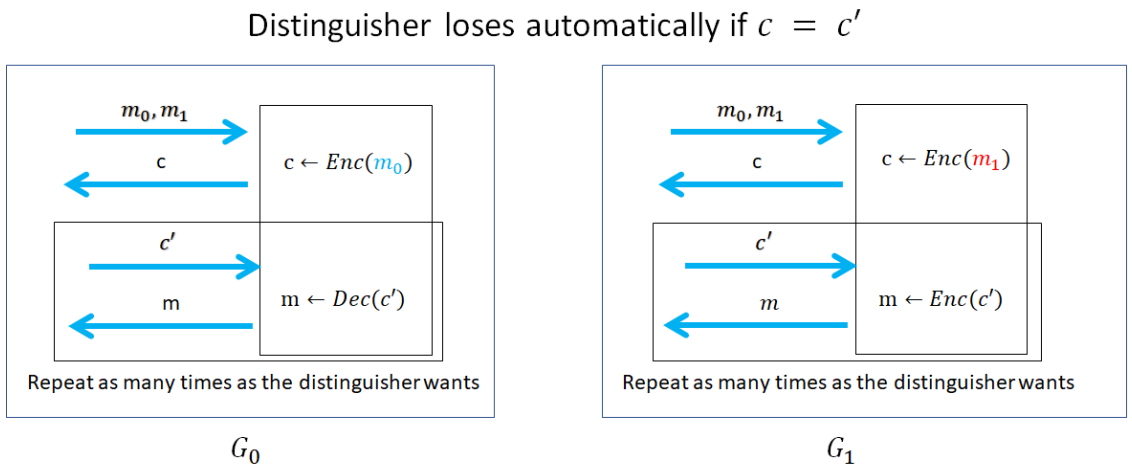
If (Gen,Mac,Verify) is a strong mac
then the left scheme is a strong mac scheme but not the right scheme

Interesting lesson

- Some definitions of security for a task are stronger.
- Depending on your variant of a definitions
  - a schemes can be considered secure depending on which definition you use
- We say that definition A implies definition B (A>B) if every scheme which satisfies definition A also satisfies definition B
  - CCA => CPA
  - CPA => distinguishability
  - Strong authentication => weak authentication
- You need precisely to look at a definition to see if a scheme fulfills or not

CCA game (chosen ciphertext)

- In the CCA game the adversary also gets to encrypt messages of his choice



Groups, fields, primes

Divisibility

- Definition: a divides n written as $(a \mid n)$ if there exists a number c such that $ca = n$

- $GCD(a, b) \coloneqq \max_{x \in \mathbb{N}} (x \mid a)\ and\ (x \mid b)$

Finding the GCD

- If $(b \mid a)$ then $GCD(a, b) = b$
- $GCD(a, b) = GCD(\max(a, b), \min(a, b))$
- (a>b) $\Rightarrow$ $GCD(a, b) = GCD(a - b, b)$

Simple algorithm for GCD

GCD(a,b)

if $(a < b)$

$return\ GCD(b, a)$

if $(a\ \%\ b\ ==\ 0)$

return b;

return $GCD(a - b, b)$

Proposition 8.3

- Let a, b then there exists positive integers X,Y such that

  Xa + Yb = gcd(a,b)

- Given u < n, we want to find v such that $u * v \ mod \ n = 1$
  - Equivalent to finding v : $uv \ + Yn = 1$

Simple algorithm for GCD

GCD(a,b)

if $(a \,\%\, b \;==\; 0)$

return b

if $(a < b)$

return $GCD(a, b)$

return $GCD(b,\; a \,\%\, b)$

Extended GCD algorithm

Same as GCD except that for a,b we also want to find

We also want to find X, Y such that $aX + bY = GCD(a, b)$

- Help us find inverse
  - Given u < n, we want to find v such that u $* v \; mod \; n = 1$
  - Equivalent to finding v : $uv + Yn = 1$
  - Can use EGC to find v

Extended GCD algorithm

EGCD(a,b)

if $(a < b)$

return $EGCD(b, a)$

if(b divides a)

return $(b, 0,1)$

// Compute q, r such that $a = qb + r$

$q = a \,/\, b$

$r = a - qb$

$(d, x, y) \leftarrow eGCD(b, r)$

return $EGCD(d, y, x - yq)$

Multiplicative inverse

- The multiplicative inverse of x in Z_n

$$y \in Z_n \text{ such that } x \times y = 1 \ (mod\ n)$$

Relatively prime

- We say that two numbers a, b are relatively prime (co-prime)

$$GCD(a, b) = 1$$

Examples

15,32

24, 35

Non-example

2,14

3, 183

Theorem on relatively prime

- If $a > b$ then there exists unique $p,\ r < a$

$$\mathrm{a} = \mathrm{b}q + r$$

Group

- A group
  - Consists
    - Set S
    - Operation $\odot : S \times S \rightarrow S$
    - Identity-element
  - Properties
    - Closure $x, y \in S \Rightarrow x \odot y \in S$
    - Identity $\exists\, e \in S : \; x \in S \Rightarrow e \odot x = x$ (we use e to denote the identity element)
    - Associativity $x, y, z \in S \Rightarrow (\mathrm{x} \odot y) \odot z \Rightarrow x \odot (y \odot z)$
    - Inverse: $x \in S \Rightarrow \exists\, y \in S : x \odot y = e$
  - Extra property
    - Commutativity: $x, y \in S \Rightarrow x \odot y = y \odot x$

Examples of groups

- $(Z_n, +)$ is a group where + is addition mod n
  - Closure $x, y \in Z_n \Rightarrow x + y \in Z_n$
  - Identity e = 0, we have that x+0 = x
  - Associativity x+y+z = (x + y) + z = x + (y+z)
  - Inverse : x + (n-x) = 0

Examples of group

- $(Z_p \setminus \{0\},\ *)$ is a group if p is prime
  - Closure $x * y \in Z_p$
  - Identity e = 1, we have that $x \times 1 = x$
  - Associativity $x * y * z = (x * y) * z = x * (y * z)$
  - Inverse :
    - For each u there exists v, y such that $uv + yn = \gcd(a, n) = 1$
    - $uv + yn \ (mod\ n) = uv = 1$
    - v is the inverse

Examples of group

- $(Z_n \setminus \{0\},\ *)$ is **not** a group if n is **not** prime
  - Closure, Identity and associativity hold
  - There exists u without an inverse :
    - Choose u such that $GCD(u, n) = c \neq 1$ (must exists since n is not prime)
    - $u'\ c\ = u$
    - $n'\ c\ = n$
    - $uv\ (mod\ n) = (cu')\ v\ (mod\ cn') = c\ (u'v\ )\ (mod\ cn')$
    - If exists x such that cx (mod cn') = 1
    - Then there exists z such that $cx\ =\ 1\ +\ cn'z$
    - This implies that $c(x - n'z)\ =\ 1$
    - Can only occur if c=1 which contradicts the fact that $c \neq 1$

Examples of group

- Examples of group
  - Numbers with addition
  - Real with addition
  - Rationals with addition
- Non-group
  - Positive numbers under addition (no inverse)
  - Odd numbers (not closed under addition)

Exponential in groups

We use $g \cdot m \; instead \; of \; g^m \; when \; the \; group \; g \; is \; additive$

- Definition
  - $g^1 = g$
  - $g^i = g \times g^{i-1}$
  - Linear time
- Fast evaluation
  - $g^1 = g$
  - $g^{2i} = (g^2)^i$
  - $g^{2i+1} = (g^2)^i \times g$
  - Logarithmic time

Properties of group

- Let G $\coloneqq$ $(S,\cdot)$ be a group, $|G| = |S| = m$ and let $g \in S$
  - Theorem: $g^m = 1$
  - Theorem: $g^x = g^{x \bmod m}$

Z_n^*

- $Z_n^* := \{x \in Z_n \mid GCD(x, n) = 1\}$
- $(Z_n^*, \times)$ is an abelian group when $\times$ denotes modular multiplication Z_n
- Going to be critical for many cryptosystems.

Subgroup generated from an element

- Let G $\coloneqq$ $(S,\cdot)$ be a group and let $g \in S$
  - We denote $G_S \coloneqq \cup_{i=1}^{\infty} \{g^i\}$
- Theorem: $(G_S,\times)$ is a group (we say that G_s is a subgroup by G)
- We say that g generates G if $G_s = G$

Subgroup generated from an element

- Let G $\coloneqq$ $(S,\times)$ be a group and let $g \in S$
  - We denote $G_S \coloneqq \cup_{i=1}^{\infty} \{g^i\}$ where $g^i = g \times \cdots \times g$ (n times)
- Theorem: $|G_S|$ divides |G|
- We say that g generates G if $|G_s| = |G|$

Discrete logarithm

- Given group G, generator g, For a given y find x such
$$g^x = y$$

- Reminder g is a generator if $\{g^i \mid i \in N\} \coloneqq G$

- Cryptographers hope that in certain groups, finding g is hard.

Field

- Field
  - Consists
    - Set S
    - Operations $(+,\times)$
    - Zero-element 0
    - One-element 1
  - Properties
    - $(S,+)$ is a *commutative* group with identity element 0 (inverse of a is $-a$)
    - $(S \setminus \{0\},\times)$ is a *commutative* group with identity element 1 (inverse of a is $a^{-1} = 1/a$)
    - Distributivity:

Example of field

- $(Z_p, +, *)$ is a field if p is prime
  - Distributivity $a * (b + c) = ab + ac$

- $(Z_p, +, *)$ is not a field if p is not prime

Message authentication code

- Worked because it was a field
- Did not work when it was not a field

Are there finite fields that do not have prime size

- Yes

- Conditions:
  - Every finite field must have size of the form p^k
  - For every such size there exists such a finite field

Finite Fields for 2^8

- S = polynomials $a_0 + a_1 x + \cdots + a_7 x^7$
  - $a_j \in \{0,1\}$
- Map each string to a polynomial
  - $(a_0, \cdots, a_7) \rightarrow a_0 + a_1 x + \cdots + a_7 x^7$
- Addition operations
  - $f(x) \coloneqq a_0 + a_1 x + \cdots + a_7 x^7$
  - $\mathrm{g}(x) \coloneqq b_0 + b_1 x + \cdots + b_7 x^7$
  - $(f + g)(x) \coloneqq (a_1 \oplus b_1) + \cdots + (a_7 \oplus b_7) x^7$

Finite Fields for 2^8 (carryless multiplication)

- $\mathrm{f}(x) = x + 1$
- $g(x) = x + 1$
- $(x + 1) \times (x + 1) = x(x + 1) \oplus (x + 1)$
  $= (x^2 + x) \oplus (x + 1) = x^2 + 1$

Difference

$(x + 1) \times (x + 1) = x + 1 \rightarrow 5$

$3 * 3 = 9$

Irreducible polynomial

- A polynomial p is irreducible if there exists no polynomials q, r such that $p = q \times r$

- For every polynomial p,q where p>q, there exists polynomial r, w such that

$$\mathrm{p} = \mathrm{qr} + \mathrm{w}$$

  - $p \; mod \; q \coloneqq w$

Field 256 (used for AES, MACS, etc)

- $h(x) \coloneqq x^8 + x^4 + x^3 + x + 1$
- S $\coloneqq$ polynomials $a_0 \ + \ a_1 x \ + \cdots + \ a_7 \ x^7$
  - $a_j \in \{0,1\}$
- Addition
  - $f(x) \coloneqq a_0 \ + \ a_1 x \ + \cdots + \ a_7 \ x^7$
  - $g(x) \coloneqq b_0 \ + \ b_1 x \ + \cdots + \ b_7 \ x^7$
  - $(f + g)(x) \coloneqq (a_1 \oplus b_1) + \cdots + (a_7 \oplus \ b_7) x^7$
- Multiplication
  - $(f \otimes g)(x) \coloneqq f(x) \times g(x) \ mod \ \ h(x)$ where times denotes carry less multiplication